Rockchip
RV1109/RV1126
Technical Reference Manual
Part1

Revision 1.0
June. 2020

Revision History

Date	Revision	Description
2020-6-5	1.0	Initial Release

Table of Content

Figure Index

Table Index

Warranty Disclaimer

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

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

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

Copyright and Patent Right

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

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

Trademarks

Rockchip and Rockchip™ 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.

Appendix

Abbreviation	Full Name	Instruction
ACA	Accessory Charger Adapter	An adaptor which allows a single USB port to be attached to both a charger and another device at the same time.
ADC	Analog-to-Digital Converter	Used to convert analog signal to digital signal.
AE	Automatic Exposure	Realizes the function of automatic exposure processing.
AF	Auto Focus	Realizes the measurement of image definition.
AGC	Automatic Gain Control	An adaptive system found in electronic devices that automatically controls the gain of a signal: the average output signal level is fed back to adjust the gain to an appropriate level for a range of input signal levels.
AI	Artificial Intelligence	A new technical science to study and develop theories, methods, techniques and application systems for simulating, extending and extending human intelligence.
AVC	Advanced Video Coding	One video agreement, also can be called H.264
AVS	Audio Video coding Standard	One video agreement
AWB	Automatic White Balance	Provides the white balance function, and configures the gains of the R, Gr, Gb, and B components.
BCSH	Brightness, Contrast, Saturation and Hue	BCSH is used for brightness, contrast, saturation and hue adjustment.
BLC	Black Level Correction	A function to subtract a DC offset from picture data.
BLE	Blending	Gather many source data to final output data
CC	Combining Check	Can correction wrong ME and MC.
CCM	Color Correction Matrix	Used to correct the color crosstalk because of three channel response of sensor when only one channel color input.
CDP	Charging Downstream Port	A Charging Downstream Port (CDP) is a downstream port on a device that complies with the USB 2.0 definition of a host or a hub, except that it shall support the Charging Downstream Port features specified herein.
CMD	Command	Command
CMOS	Complementary Metal Oxide Semiconductor	Use for CMOS sensor
CSC	Color Space Convert	RGB2YUV, YUV2RGB
CSI	Camera Serial Interface	A standard interface between a camera and a host processor for mobile applications.
DAC	Digital-to-Analog Converter	A system that converts a digital signal into an analog signal.
DAP	Debug Access Port	A TAP block that acts as an AMBA (AHB or AHB-Lite) master for access to a system bus. The DAP is the term used to encompass a set of modular blocks that support system wide debug. The DAP is a modular component, intended to be extendable to support optional access to multiple

Abbreviation	Full Name	Instruction
		systems such as memory mapped AHB and CoreSight APB through a single debug interface.
DCP	Dedicated Charging Port	A downstream port on a device that outputs power through a USB connector, but is not capable of enumerating a downstream device
DCT	Discrete Cosine Transform	One video encoder tool.
DDR	Double Data Rate	A kind of dynamic memory.
DDR3	Double Data Rate 3	A kind of dynamic memory.
DDR3L	Double Data Rate 3 Low Voltage	A kind of dynamic memory.
DDR4	Double Data Rate 4	A kind of dynamic memory.
DFI	DDR PHY Interface	An interface protocol that defines the signals, timing parameters, and programmable parameters required to transfer control information and data over the DFI, to and from the DRAM devices, and between the MC and the PHY.
DMA	Direct Memory Access	A feature of computer systems that allows certain hardware subsystems to access main system memory, independent of the central processing unit (CPU).
DMAC	Direct Memory Access Controller	A programmable controller that manages DMA transfers.
DPCC	Defect Pixel Cluster Correction	Used to correct the defect pixels of the sensor, and supports both static and dynamic mode.
DST	Destination	Destination Image
DVP	Digital Video Port	A traditional sensor output interface.The video data are output in parallel.
EEDI	Enhanced Edge based Interpolation	Could interpolation smaller than 10 degree.
EHCI	Enhanced Host Controller Interface	The Enhanced Host Controller Interface (EHCI) for a Host Controller for the USB2.0
EOI	End Of Interrupt	Indicate the completion of interrupt processing for a given interrupt.
FBC	Frame Buffer Compression	A kind of method to reduce DDR bandwidth
FBCD	Frame Buffer Compression Decoder	Decode TNR and NR compression data from ddr under thier respective fbcd mode.
FBCE	Frame Buffer Compression Encoder	Encode TNR, Sharp and FEC data to ddr under their respective fbce mode.
FEC	Fisheye Correction	Implemented by backward mapping.
FIFO	First In First Out	A method for organizing and manipulating a data buffer, where the oldest (first) entry, or head of the queue, is processed first.
FPN	Fixed Pattern Noise	A function for fixed pattern noise
FPS	Frame per Second	It is express that how many frames can be decoded by VDPU at one second.
FRC	Frame Rate Control	An algorithm of dither
FS	Full Speed	Full Speed model in USB2.0
GIC	Green Imbalance Cor rection	A method to correct the color imbalance in captured images
GPIO	General Purpose Input/Output	An uncommitted digital signal pin on an integrated circuit whose behavior (including

Abbreviation	Full Name	Instruction
		whether it acts as input or output) is controllable by the user at run time.
HDR	High-Dynamic Range	Realize the function of three/two frames synthesizing one frame.
HEVC	High Efficiency Video Coding	One video agreement, also can be called H.265
HNP	Host Negotiation Protocol	Host Negotiation Protocol in USB OTG Specification.
HS	High Speed	High Speed model in USB2.0
I2C	Inter-Integrated Circuit	Two wired (SCL and SDA), bi-directional serial bus
I2S	Inter-IC Sound, Integrated Interchip Sound	It is one of digital audio transmission standard.
IEP	Image Enhancement Processor	The main function is deinterlace.
Int8	Integer 8	8 bit integer including 1 bit sign bit
ISP	Image Signal Processor	contains standard sensor picture data processing functions
ISPP	Image Signal Post-Processing	Contains image post-processing and scaling function.
IXOY	Input X Fields Output Y frame	A kind of deinterlace mode(X and Y mean a number).
JFIF	JPEG File Interchange Format	One image file standard.
JPEG	Joint Photographic Experts Group	One video agreement, used for picture compress.
LCD	Liquid Crystal Display	A type of display.
LDCH	Lens-Distortion	Achieves distortion correction under fisheye lens.
LPDDR3	Low Power Double Data Rate 3	A kind of dynamic memory.
LPDDR4	Low Power Double Data Rate 4	A kind of dynamic memory.
LS	Low Speed	Low Speed model in USB2.0
LSC	Lens Shading Correction	A function for lens shading problem
LUT	Look Up Table	Look Up Table
LVDS	Low-Voltage Differential Signaling	A differential signal technology with low power consumption, low bit error rate, low crosstalk and low radiation.
MAD	Multiply-add Unit	This is the unit designed to process multiply-add operation
MBAFF	Macro-block Adaptive Field Frame	One feature of h.264 stream.
MBPS	Mega Bits Per Second	The express that how many bits can be decoded by VDPU at one second.
MC	Motion Compensation	Filed based
MD	Motion Detection	Frame based
ME	Motion Estimate	Frame based
MI	Memory Interface	Write or read back data interface
MIPI	Mobile Industry Processor Interface	An open standard and specification for mobile application processors initiated by the MIPI consortium
MSPS	Million sample per	Sample rate

Abbreviation	Full Name	Instruction
	second	
MTL	Media Access Controller Transaction Layer	The MAC Transaction Layer (MTL) provides the FIFO memory Interface to buffer and regulate the packets between the application system memory and the MAC. It also enables the data to be transferred between the application clock and MAC clock domains. The MTL layer has two data paths: Transmit path and Receive Path.
MV	Motion Vector	Frame based
MVC	Multiview Video Coding	Also known as MVC 3D, it is a stereoscopic video coding standard for video compression.
NG	Noise Gate	A module used to detect and gate the noise.
BIU	Bus Interface Units	BIU convert the incoming transaction information into special internal packets.
NN	Neural Network	Artificial Neural Network used in Deep Learning.
NPU	Neural Process Unit	NPU is the process unit which is dedicated to neural network. It is designed to accelerate the neural network arithmetic in field of AI (artificial intelligence) such as machine vision and natural language processing.
OHCI	Open Host Controller Interface	Open Host Controller Interface for USB.
OSD	On Screen Display	An on screen menu for making adjustments to the display
OTG	On The Go	This means USB can be host or device by configuration.
OTP	One Time Programmable	One Time Programmable Non-Volatile Memory system.
OTP NVM	One Time Programmable Non-Volatile Memory	One Time Programmable Non-Volatile Memory.
PCM	Pulse Code Modulation	A method used to digitally represent sampled analog signals. In a PCM stream, the amplitude of the analog signal is sampled regularly at uniform intervals, and each sample is quantized to the nearest value within a range of digital steps.
PD	Pull Down	A kind of film.
PDAF	Phase Detection Auto Focus	Replaces the PDAF points inside the raw input data
PDM	Pulse Density Modulation	A 1-bit over sampled audio format
PGA	Programmable Gain Amplifier	It can be controlled by software to amplify or decrease the amplitude of signal.
PWM	Pulse Width Modulation	A method for getting analog results with digital means. Digital control is used to create a square wave, a signal switched between on and off.
QoS	Quality of Service	The statistical allocation of throughput and delay, in terms of bandwidth and latency.
RAWNR	RAW Noise Reduction	Noise reduction on RAW domain
RGA	Raster Graphic Acceleration	Include image scale, alpha, overlay, rotation and so on.
RISC-V	Reduced Instruction Set Computer Five	RISC-V (pronounced "risk-five") is a new instruction set architecture (ISA) that was originally designed to support computer

Abbreviation	Full Name	Instruction
		architecture research and education, but which we now hope will also become a standard free and open architecture for industry implementations.
ROI	Region Of Interest	Tile based, Part of interest region in one picture
RSP	Role Swap Protocol	Role Swap Protocol in USB OTG Specification.
RX	Receive	
S/s	sample per second	Sample rate
SAR-ADC	Successive Approximation Register (SAR) A/D Converter	Voltage scaling module
SDP	Standard Downstream Port	A Standard Downstream Port (SDP) refers to a downstream port on a device that complies with the USB2.0 definition of a host or hub.
SDRAM	Synchronous Dynamic Random-access Memory	A kind of dynamic memory.
SLC	Single-Level Cell	Each cell in flash can only store 1bit data
SOF	Start of Frame	The first transaction in each (micro)frame. An SOF allows endpoints to identify the start of the (micro)frame and synchronize internal endpoint clocks to the host.
SRC	Source	Source Image
TDM	Time Division Multiplexing	TDM uses different time periods of the same physical connection to transmit different signals
TMO	Tone mapping	Realize wide dynamic compression
TNR	Time based Noise Reduction	Reduce noises to ensure that the image is quiet with the edges and details kept in YUV domain.
TS-ADC	Temperature-Sensor ADC	Temperature scaling module
TX	Transmit	
USB	Universal Serial Bus	Universal Serial Bus for peripheral.
UTMI	USB 2.0 Transceiver Macrocell Interface	Transceiver Macrocell Interface for USB2.0.
UVNR	UV based Noise Reduction	Filter the color noise
VDPU	Video Decoder Process Unit	A processor that can decoder video stream to picture for display.
VEPU	Video Encoder Process Unit	A processor that can encoder picture to video stream.
VOP	Video Output Processor	A video process engine and a display interface from memory frame buffer to display device.
VPU	Video Process Unit	Include VDPU and VEPU.
WDR	Wide Dynamic Range	Make the particularly bright and dark parts of the scene can see very clearly at the same time.
xHCI	eXtensible Host Controller Interface	eXtensible Host Controller Interface for Universal Serial Bus.
YNR	Y based Noise Reduction	Filter the noise of Y component

Difference between RV1109 and RV1126

Type	RV1109	RV1126
CPU	Dual A7	Quad A7
NPU	1.2Tops	2.0TOPS
ISP	5M up to 3072-pixel wide	14M <=4416*3312
Encoder	5M up to 3072-pixel wide	4K30 H.264/H.265
Decoder	5M up to 3072-pixel wide	4K30 H.264/H.265

Chapter 1 System Overview

1.1 Address Mapping

RV1109/RV1126 boot from internal BootRom, which supports remap function by software programming. Remap is controlled by PMU_SGRF_SOC_CON0[13]. When remap is set to 2'b01, the BootRom is un-accessible and PMU_SRAM is mapped to address 0xFFFF0000. When remap is set to 2'b10, the BootRom is un-accessible and SYSTEM_SRAM is mapped to address 0xFFFF0000.

Table 1-1 Address Mapping

Module	Start Address	Size	Module	Start Address	Size
GRF	0xFE000000	128KB	SARADC	0xFF5E0000	64KB
PMU_GRF	0xFE020000		CPU_TSADC	0xFF5F0000	32KB
DDR_GRF	0xFE030000	64KB	NPU_TSADC	0xFF5F8000	32KB
RESERVED	0xFE040000	64KB	DCF	0xFF600000	64KB
BUS_SGRF	0xFE0A0000	384KB	CAN	0xFF610000	64KB
PMU_SGRF	0xFE0B0000	64KB	GPIO1	0xFF620000	64KB
RESERVED	0xFE0C0000	64KB	GPIO2	0xFF630000	64KB
GIC400	0xFEFF0000	7.0625MB	I2S2_2CH	0xFF820000	64KB
DEBUG	0xFF000000	64KB	PDM	0xFF830000	64KB
CORE_PVTM	0xFF040000	256KB	AUDPWM	0xFF840000	64KB
RESERVED	0xFF060000	64KB	RKACDC_DIG	0xFF850000	64KB
RESERVED	0xFF3D0000	3.5MB	RESERVED	0xFF860000	1.3125MB
PMU	0xFF3E0000	64KB	DDR_STDBY	0xFF9B0000	32KB
I2C0	0xFF3F0000	64KB	RESERVED	0xFF9B8000	32KB
I2C2	0xFF400000	64KB	DDR_MONITOR	0xFF9C0000	32KB
UART1	0xFF410000	64KB	RESERVED	0xFF9D0000	550KB
SCRAMBLE	0xFF420000	64KB	UPCTL2	0xFFA50000	64KB
PWM0	0xFF430000	64KB	FIREWALL_APB	0xFFA60000	64KB
PWM1	0xFF440000	64KB	RESERVED	0xFFA70000	448KB
SPI0	0xFF450000	64KB	VICAP	0xFFAE0000	64KB
GPIO0	0xFF460000	64KB	RGA	0xFFAF0000	64KB
PMU_PVTM	0xFF470000	64KB	VOP_LITE	0xFFB00000	64KB
PMU_CRU	0xFF480000	64KB	CSI RX CTRL	0xFFB10000	64KB
CRU	0xFF490000	64KB	IEP	0xFFB20000	64KB
DDR_PHY	0xFF4A0000	64KB	DSI_CTRL	0xFFB30000	64KB
CSI_PHY0	0xFF4B0000	64KB	RESERVED	0xFFB40000	64KB
CSI_PHY1	0xFF4B8000	32KB	ISP2.0	0xFFB50000	64KB
OTG_APB	0xFF4C0000	32KB	ISPP	0xFFB60000	128KB
HOST_APB	0xFF4C8000	32KB	VDPU343	0xFFB80000	64KB
DSI_PHY	0xFF4D0000	32KB	VPU121	0xFFB90000	64KB
DMAC	0xFF4E0000	64KB	RESERVED	0xFFBA0000	64KB
SDMAC	0xFF4F0000	64KB	VEPU52X	0xFFBB0000	64KB
CRYPTO	0xFF500000	64KB	NPU	0xFFBC0000	256KB
I2C1	0xFF510000	64KB	NPU_PVTM	0xFFC00000	64KB
I2C3	0xFF520000	64KB	RESERVED	0xFFC10000	64KB
I2C4	0xFF530000	64KB	RESERVED	0xFFC20000	128KB
I2C5	0xFF540000	64KB	GMAC	0xFFC40000	64KB
PWM2	0xFF550000	64KB	EMMC	0xFFC50000	64KB
UART0	0xFF560000	64KB	SDMMC	0xFFC60000	64KB

Module	Start Address	Size	Module	Start Address	Size
UART2	0xFF570000	64KB	SDIO	0xFFC70000	64KB
UART3	0xFF580000	64KB	NANDC	0xFFC80000	64KB
UART4	0xFF590000	64KB	FSPI	0xFFC90000	64KB
UART5	0xFF5A0000	64KB			
SPI1	0xFF5B0000	64KB	RESERVED	0xFFCA0000	384KB
OTP_NS	0xFF5C0000	64KB	USB2.0_OTG0	0xFFD00000	1MB
OTP_S	0xFF5D0000	64KB	USB2.0_HOST0	0xFFE00000	256KB

Notes:
PMU_SRAM only has 8KB memory size.
MCU internal timer can only be accessed by access by MCU.

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

Table 1-2 Address Remapping

remap[1:0]=2'b00		remap[1:0]=2'b11		remap[1:0]=2'b10	
		not accessible	BootRom (20KB)	not accessible	BootRom (20KB)
0xFFFF0000	BootRom(20KB)	0xFFFF0000	PMU_SRAM(8KB)	0xFF000000	SYSTEM_SRAM (64KB)
0xFF710000	PMU_SRAM(8KB)	0xFF710000	PMU_SRAM(8KB)	0xFF710000	PMU_SRAM(8KB)
0xFF700000	SYSTEM_SRAM (64KB)	0xFF700000	SYSTEM_SRAM (64KB)	0xFF700000	SYSTEM_SRAM (64KB)

1.2 System Boot

RV1109/RV1126 provides system boot from off-chip devices such as SDMMC card, eMMC memory, serial Nand or Nor flash. When boot code is not ready in these devices, also provide system code download into them by USB OTG interface. All of the boot code will be stored in internal BootRom. The following is the whole boot procedure for boot code, which will be stored in BootRom in advance.
The following features are supported.

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

Following figure shows RV1109/RV1126 boot procedure flow.

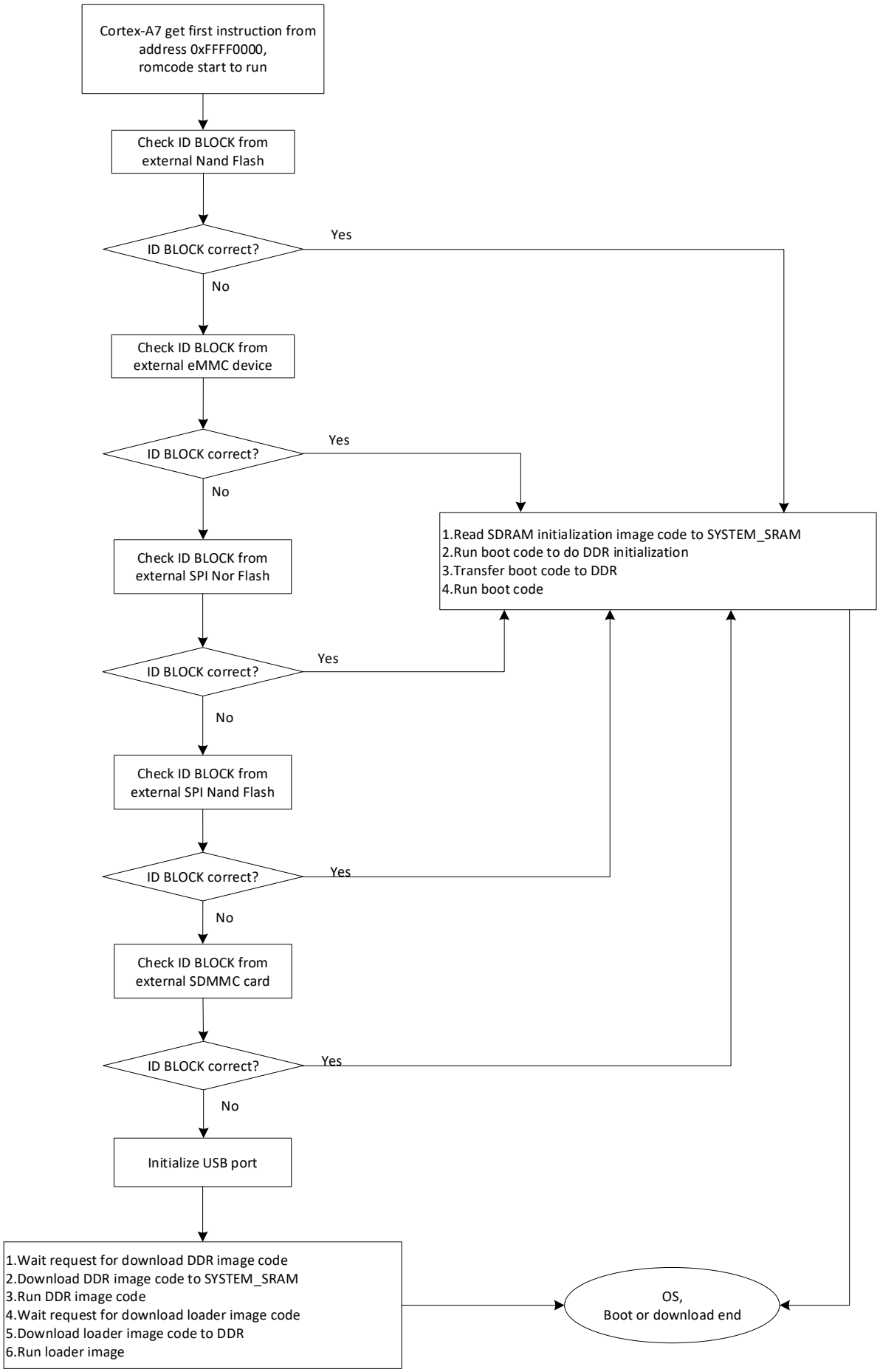


Fig. 1-1 RV1109/RV1126 Boot Procedure Flow

1.3 System Interrupt Connection

RV1109/RV1126 provides an general interrupt controller (GIC) for CPU, which has 256 SPI (shared peripheral interrupts) interrupt sources and 3 PPI(Private peripheral interrupt) interrupt source and separately generates one nIRQ and one nFIQ to CPU. The triggered type for each interrupt is high level sensitive, not programmable. The detailed interrupt sources connection is in the following table.

Table 1-3 RV1109/RV1126 Interrupt Connection List

Number	Source	Polarity
0-31PPI		High level
32	dcf_int	High level
33	dmac_int	High level
34	dmac_abort_int	High level
35	crypto_int	High level
36	i2c0_int	High level
37	i2c1_int	High level
38	i2c2_int	High level
39	i2c3_int	High level
40	i2c4_int	High level
41	i2c5_int	High level
42	spi0_int	High level
43	spi1_int	High level
44	uart0_int	High level
45	uart1_int	High level
46	uart2_int	High level
47	uart3_int	High level
48	uart4_int	High level
49	uart5_int	High level
50	pwm0_int	High level
51	pwm0_int_pwr	High level
52	pwm1_int	High level
53	pwm1_int_pwr	High level
54	pwm2_int	High level
55	pwm2_int_pwr	High level
56	timer0_int	High level
57	timer1_int	High level
58	timer2_int	High level
59	timer3_int	High level
60	timer4_int	High level
61	timer5_int	High level
62	stimer0_int	High level
63	stimer1_int	High level
64	wdt_int	High level
65	swdt_int	High level
66	gpio0_int	High level
67	gpio1_int	High level

Number	Source	Polarity
68	gpio2_int	High level
69	gpio3_int	High level
70	gpio4_int	High level
71	cpu_tsadc_int	High level
72	saradc_int	High level
73	msch_alarm_irq	High level
74	upctl_arpoison_int	High level
75	upctl_awpoison_int	High level
76	upctl_alert_err_int	High level
77	ddrmon_int	High level
78	i2s8ch_int	High level
79	i2s2ch0_int	High level
80	i2s2ch1_int	High level
81	pdm_int	High level
82	audpwm_int	High level
83	irq_isp_mmu_0	High level
84	isp_irq_isp	High level
85	ciflite_int_out_ciflite	High level
86	mi_irq_isp	High level
87	mipi_irq_isp	High level
88	intr1_csirx	High level
89	intr2_csirx	High level
90	cif_int_out_cif	High level
91	vop_intr_vopm	High level
92	vop_intr_post_lb_vopm	High level
93	mipi_dsi_host_irq_dsihost	High level
94	rga_irq	High level
95	ispp_irq	High level
96	fec_irq	High level
97	ispp_mmu0_r_irq	High level
98	ispp_mmu0_w_irq	High level
99	ispp_mmu1_irq	High level
100	venc_int	High level
101	venc_mmu0_int	High level
102	venc_mmu2_int	High level
103	vdec_dec_irq	High level
104	vdec_mmu_irq	High level
105	jpeg_dec_irq	High level
106	jpeg_enc_irq	High level
107	jpeg_mmu_irq	High level
108	sdcard_int	High level
109	sdio_int	High level
110	emmc_int	High level
111	nandc_int	High level

Number	Source	Polarity
112	sfc_int	High level
113	decom_int	High level
114	usbhost_ehci_int	High level
115	usbhost_ohci_int	High level
116	usbhost_arb_int	High level
117	usbotg_int	High level
118	usbotg_host_legacy_smi_interrupt	High level
119	usbotg_host_sys_err	High level
120	usbotg_pme_generation	High level
121	reserved	High level
122	reserved	High level
123	reserved	High level
124	reserved	High level
125	reserved	High level
126	pmu_int	High level
127	gmac_sbd_int	High level
128	gmac_pmt_int	High level
129	gmac_sbd_perch_rx_int	High level
130	gmac_sbd_perch_tx_int	High level
131	gmac_lpi_int	High level
132	can_int	High level
133	otpcs_int	High level
134	otpcns_int	High level
135	otpmask_int	High level
136	pmic_int	High level
137	sdmmc_detectn_irq_grf	High level
138	sdmmc_detectn_flt	High level
139	xaq2_intr_nanoq	High level
140	pmupvtm_int	High level
141	npupvtm_int	High level
142	cpupvtm_int	High level
143	mailbox_irq_ap	High level
144	mailbox_irq_bb	High level
145	npu_tsadc_int	High level
146	iep_irq	High level
147	usbphy_otg_bvalid_irq	High level
148	usbphy_otg_id_irq	High level
149	usbphy_otg_linestate_irq	High level
150	usbphy_host_linestate_irq	High level
151	usbphy_host_disconnect_irq	High level
152	usbphy_otg_disconnect_irq	High level
153-154	reserved	NA
155	~nPMUIRQ[0]	High level
156	~nPMUIRQ[1]	High level

Number	Source	Polarity
157	~nPMUIRQ[2]	High level
158	~nPMUIRQ[3]	High level

1.4 System DMA Hardware Request Connection

RV1109/RV1126 provides one DMA controller (DMAC) inside the system. For detailed descriptions of DMAC, please refer to Chapter 8.

Table 1-4 RV1109/RV1126 DMAC Hardware Request Connection List

Request Index	Source	Polarity
0	SPI0 RX	High level
1	SPI0 TX	High level
2	SPI1 RX	High level
3	SPI1 TX	High level
4	UART0 RX	High level
5	UART0 TX	High level
6	UART1 RX	High level
7	UART1 TX	High level
8	UART2 RX	High level
9	UART2 TX	High level
10	UART3 RX	High level
11	UART3 TX	High level
12	UART4 RX	High level
13	UART4 TX	High level
14	UART5 RX	High level
15	UART5 TX	High level
16	PWM0	High level
17	PWM1	High level
18	PWM2	High level
19	I2S0 RX	High level
20	I2S0 TX	High level
21	I2S1 RX	High level
22	I2S1 TX	High level
23	I2S2 RX	High level
24	I2S2 TX	High level
25	PDM	High level
26	AudioPWM	High level

Chapter 2 Cortex-A7

2.1 Overview

The Cortex-A7 subsystem of the device is based on the ARMv7-A architecture. The Cortex-A7 has four Cortex-A7 central processing units (CPUs). Each processor has 32KB of Level 1 (L1) instruction cache, 32KB of L1 data cache, NEON and FPU. The Cortex-A7 also includes 512KB of Level 2 (L2) cache, Snoop Control Unit(SCU), Generic Interrupt Controller (GIC), and standard CoreSight components to support SMP debug and emulation.
The key features of the Cortex-A7 include:

- ARM Cortex-A7
  - Full implements the ARMv7-A architecture profile that includes the Advanced SIMD and VFP Extensions
  - 4 Cortex-A7 processor cores
  - 32KB L1 I-Cache and 32KB L1 D-Cache per CPU
  - In-order pipeline with direct and indirect branch prediction
  - Harvard L1 memory system with a Memory Management Unit (MMU)
  - SCU ensures memory coherency
  - Interrupt controller with 128 hardware interrupt inputs
- 512KB L2 cache
  - Fixed line length of 64 bytes
  - Physically indexed and tagged cache
  - 8-way set-associative cache structure
  - Pseudo-random cache replacement policy

2.2 Block Diagram

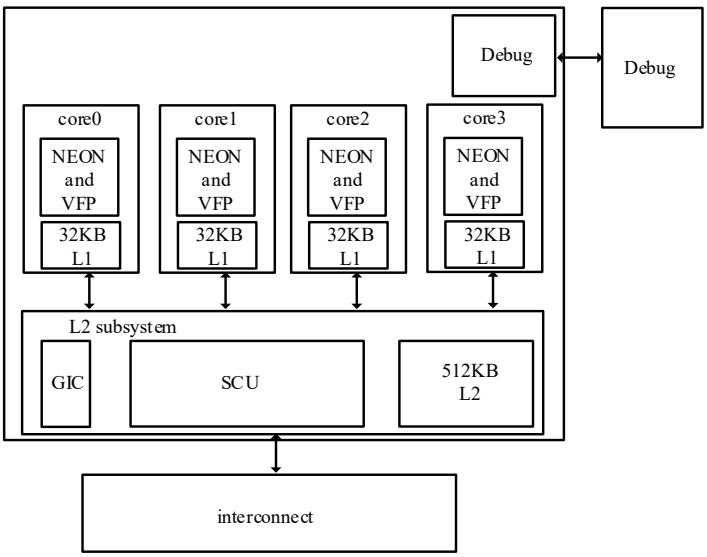


Fig. 2-1 Core Subsystem Architecture

2.3 Function Description

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

2.4 Register Description

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

Chapter 3 RISC-V

3.1 Overview

It is RISC-V compatible MCU core. The core is load-store architecture, where only load and store instructions access memory and arithmetic instructions only operate on integer registers. The core provides a 32-bit user address space that is byte-addressed and little-endian. And it only implements machine levels, one of four RISC-V privilege levels.
The key features of the RISC-V core include:

- Harvard architecture (separate instruction and data buses)
- Machine privilege level
- 32 32-bit general purpose integer registers
- Instruction set is RV32I with M and C extensions
- High-performance or area-optimized multiply/divide unit
- 3 stage pipeline implementation
- 32-bit AHB-Lite external memory interface
- Integrated Programmable Interrupt Controller(IPIC),up to 128 IRQ lines with 128to16 INTMUX
- Debug Controller with JTAG interface
- Hardware Break-point Module
- 3 embedded 64bit performance counters
- Real time clock
- Cycle counter
- Instructions-retired counter

3.2 Block Diagram

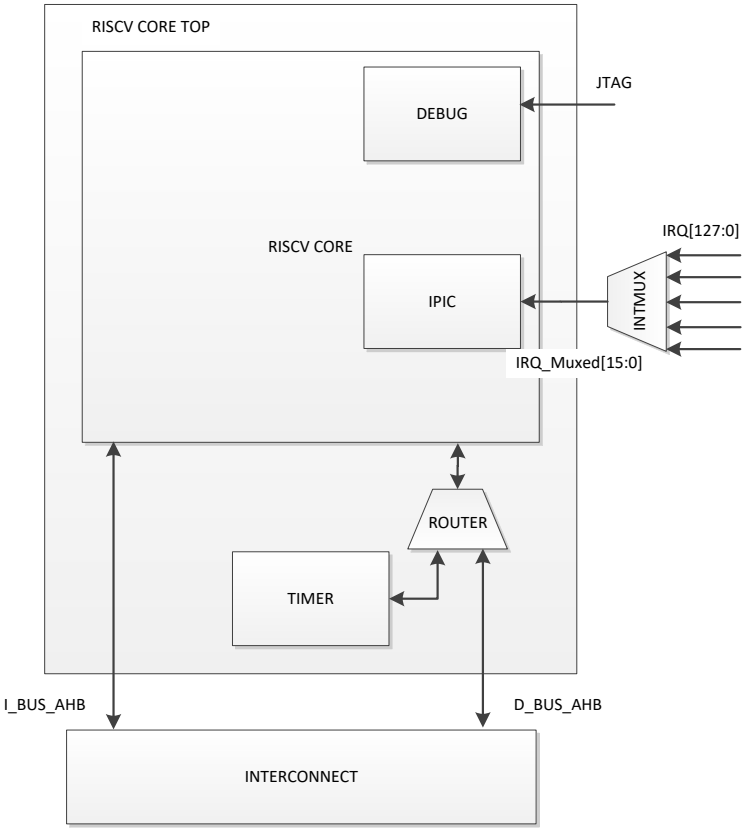


Fig. 3-1 Block Diagram of RISC-V Core

3.3 Function Description

3.3.1 Interrupt Multiplexer(INTMUX)

The Interrupt Multiplexer will select 16 interrupts from 128 interrupts using round robin algorithm. You need to configure corresponding enable bit for each interrupt. Also you could

read the flag register to obtain the interrupt in service.

3.3.2 Integrated Programmable Interrupt Controller(IPIC)

RISC-V core include Integrated Programmable Interrupt Controller (IPIC) with low latency IRQ response. IPIC can be configured using IPIC Control Status Registers.
The term Interrupt Line has the meaning of corresponding IPIC external pin where suitable source of external interrupt may be connected to.
The term Interrupt Vector has the meaning of external interrupt number which will be generated by IPIC in response to external interrupt.
IPIC supports maximum 16 Interrupt vectors [0..15] and 16 Interrupt lines [0..15], each line is statically mapped to the corresponding vector.
Interrupt Vectors are given fixed priorities. The lowest Interrupt Vector number has the highest priority. IPIC supports nested interrupts. Only one interrupt can be serviced at a time. "Void interrupt vector" is defined as a non-existent vector number 0x10. This value is used to indicate absence of a valid interrupt vector.
NOTE: Write access to the IPIC control status registers is implemented only through the use of the CSRRW(I) instructions, the CSRRS(I) and CSRRC(I) instructions are not supported.

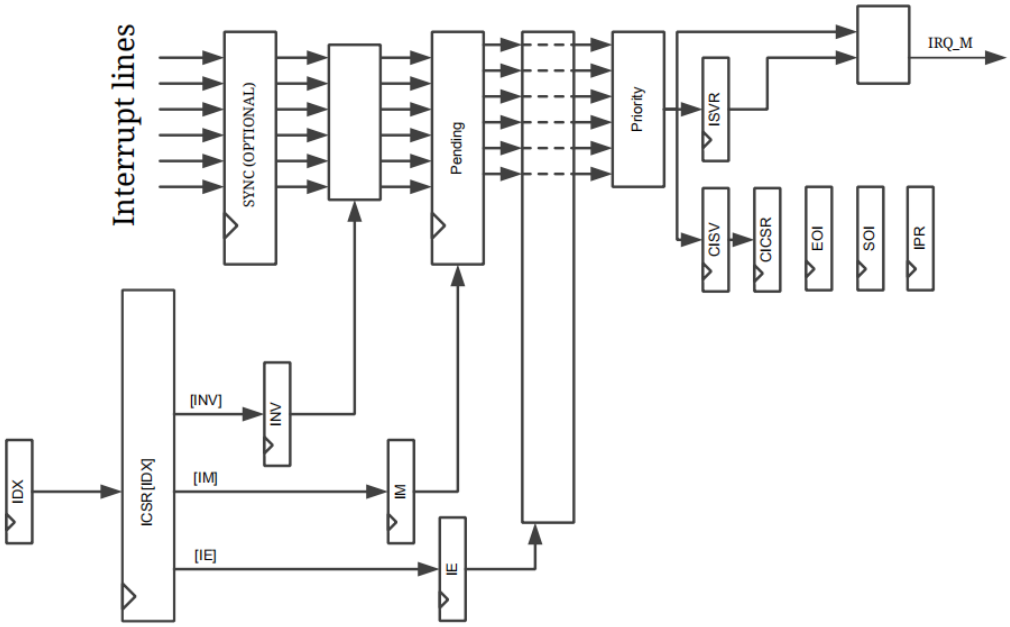


Fig. 3-2 Block Diagram of IPIC

Depending on the IM (interrupt mode), INV (line inversion) values for each vector, one of four conditions for IP (interrupt pending) bit activation is selected: high level, low level, rising edge, falling edge. Of all vectors with IP and IE (interrupt enable) bits active, the lowest numbered vector has the highest priority. Software is responsible for writing the SOI and EOI registers, thus notifying IPIC of the start and end of interrupt processing, respectively.

3.3.3 Debug

The core’s debug sub-system is implemented in compliance with the RISC-V External Debug Support specification [3]. Its block diagram is shown in Fig.1-3.

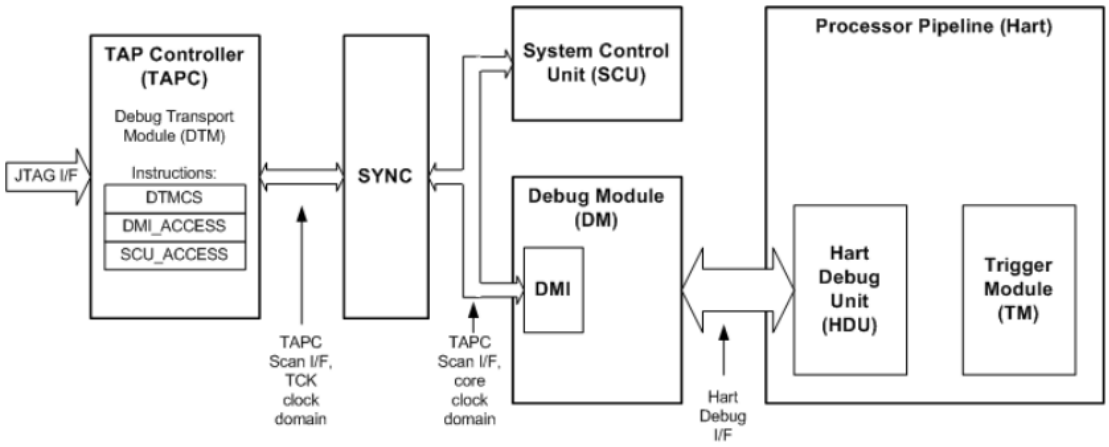


Fig. 3-3 Block Diagram of Debug Sub-system

An external debugger communicates with the core's debug sub-system via JTAG interface and TAP Controller (TAPC), playing a role of the Debug Transport Module (DTM) in terms of the RISC-V Debug Specification [3]. The TAPC implements several private TAP instructions allowing debugger to interact with internal debug units:

- DTMCS provides general control over DTM
- DMI_ACCESS provides access to the Debug Module (DM)
- SCU_ACCESS provides access to the System Control Unit (SCU)

Internal connection between TAPC and DM, DM and SCU, is a form of serial scan interface. Source and destination of the TAPC scan interface are in different clock domains: TAPC is fully running in JTAG's TCK clock domain, whereas DM and SCU are in the core clock domain. Therefore, the TAPC scan interface passes through the clock synchronization unit (SYNC).
The System Control Unit (SCU) provides control over implementation-specific reset circuitry, and allows to monitor states of main reset signals.
Using the Debug Module Interface (DMI), the Debug Module (DM) exposes a standard register interface to the core's debug features:

- run control of the core's single hart
- access to its internal registers (GPRs, CSRs)
- access to its memory space
- capability to execute arbitrary instructions from the Program Buffer

The register interface is compliant with the RISC-V Debug Specification [3].
Implementation of this debug functionality within the hart is distributed between several units, and from external prospective the most important among them are:

- Hart Debug Unit (HDU) - provides the hart's Debug Interface, connecting the hart with the DM, and contains Debug CSRs
- Trigger Module (TM) - provides a capability of hardware breakpoints, and contains Trigger CSRs

3.4 Register Description

3.4.1 General-purpose Integer Registers

General-purpose Integer Register is the user-visible registers of the core. There are 31 general-purpose registers x1-x31, which are designed to hold integer values. Register x0 is hardwired to the constant 0 and can be used as a source of constant zero or as a don't care destination register.
Don't care destination x0 is used to ignore the result of instruction execution provided that destination register is mandatory for instruction structure.
All general-purpose registers in the core are 32-bits wide.
The core implements 32-bit pc register, which is used as program counter, meaning that it holds the address of the current instruction.

3.4.2 Control and Status Register

Control/status registers (CSR) of the core are accessed atomically using instructions specifically designed for CSR access. CSR access instructions are listed in Instruction set

summary section of this specification.
According to the RISC-V specification [2], the core uses 12-bit encoding space to address up to 4096 control/status registers (CSR) in the instructions which atomically read and modify CSRs. The core implements subset of CSRs according to the mapping shown in the next paragraphs. The core follows RISC-V convention, where the upper 4 bits of the CSR address [11:8] are used to encode the read and write accessibility of the CSRs according to the privilege level. The top two bits [11:10] indicate whether the register is read/write (00, 01, or 10) or read-only (11). The next two bits [9:8] indicate the lowest privilege level that can access the CSR.
The core implements the following rules for CSR access:
1. Attempts to access a non-existent CSR raise an illegal instruction exception.
2. Attempts to write a read-only CSR also raise illegal instruction exception.
3. If a read/write register contains some bits that are read-only, then writes to the read-only bits are ignored.

3.4.3 RISC-V Registers Summary

Name	Offset	Size	Reset Value	Description
RISC-V_TIMER_CTRL	0x0000	W	0x00000001	Timer Control Register
RISC-V_TIMER_DIV	0x0004	W	0x00000000	Timer Divider Register
RISC-V_MTIME	0x0008	W	0x00000000	Machine Timer Register Low 32bit
RISC-V_MTIMEH	0x000C	W	0x00000000	Machine Timer Register High 32bit
RISC-V_MTIMECMP	0x0010	W	0x00000000	Machine Timer Compare Register Low 32bit
RISC-V_MTIMECMPH	0x0014	W	0x00000000	Machine Timer Compare Register High 32bit
RISC-V_CSR_MVENDORID	0x0F11	W	0x00000000	Machine Vendor ID Register
RISC-V_CSR_MARCHID	0x0F12	W	0x00000000	Machine Architecture ID Register
RISC-V_CSR_MIMPID	0x0F13	W	0x011432A4	Machine Implementation ID Register
RISC-V_CSR_MHARTID	0x0F14	W	0x00000000	Machine Hart ID Register
RISC-V_CSR_MSTATUS	0x0300	W	0x00001880	Machine Status Register
RISC-V_CSR_MISA	0x0301	W	0x40001104	Machine Base ISA Control Register
RISC-V_CSR_MIE	0x0304	W	0x00000000	Machine Interrupt Enable Register
RISC-V_CSR_MTVEC	0x0305	W	0x002001C0	Machine Trap-Vector Base-Address Register
RISC-V_CSR_MSCRATCH	0x0340	W	0x00000000	Machine Scratch Register
RISC-V_CSR_MEPC	0x0341	W	0x00000000	Machine Exception Program Counter Register
RISC-V_CSR_MCAUSE	0x0342	W	0x00000000	Machine Cause Register
RISC-V_CSR_MTVAL	0x0343	W	0x00000000	Machine Trap Value Register
RISC-V_CSR_MIP	0x0344	W	0x00000000	Machine Interrupt Pending Register
RISC-V_CSR_MCYCLEL	0x0B00	W	0x00000000	Machine Cycle Counter Low Register
RISC-V_CSR_MCYCLEH	0x0B80	W	0x00000000	Machine Cycle Counter High Register
RISC-V_CSR_MINSTRETL	0x0B02	W	0x00000000	Machine Instructions Retired Low Register
RISC-V_CSR_MINSTRETH	0x0B82	W	0x00000000	Machine Instructions Retired High Register
RISC-V_CSR_MCOUNTEN	0x07E0	W	0x00000005	Machine Counter Enable Register
RISC-V_CSR_DBG_SCRATCH	0x07C8	W	0x00000000	Debug Scratch Register

Name	Offset	Size	Reset Value	Description
RISC-V_CSR_IPIC_CISV	0x0BF0	W	0x00000000	Current Interrupt Vector in Service
RISC-V_CSR_IPIC_CICSR	0x0BF1	W	0x00000000	Current Interrupt Control Status Register
RISC-V_CSR_IPIC_IPR	0x0BF2	W	0x00000000	Interrupt Pending Register
RISC-V_CSR_IPIC_ISVR	0x0BF3	W	0x00000000	Interrupt Serviced Register
RISC-V_CSR_IPIC_EOI	0x0BF4	W	0x00000000	End Of Interrupt
RISC-V_CSR_IPIC_SOI	0x0BF5	W	0x00000000	Start Of Interrupt
RISC-V_CSR_IPIC_IDX	0x0BF6	W	0x00000000	Index Register
RISC-V_CSR_IPIC_ICSR	0x0BF7	W	0x00000000	Interrupt Control Status register

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

3.4.4 RISC-V Detail Registers Description

RISC-V_TIMER_CTRL

Address: Operational Base + offset (0x0000)

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

RISC-V_TIMER_DIV

Address: Operational Base + offset (0x0004)

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

RISC-V_MTIME

Address: Operational Base + offset (0x0008)

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

RISC-V_MTIMEH

Address: Operational Base + offset (0x000C)

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

RISC-V_MTIMECMP

Address: Operational Base + offset (0x0010)

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

RISC-V_MTIMECMPH

Address: Operational Base + offset (0x0014)

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

RISC-V_CSR_MVENDORID

Address: Operational Base + offset (0x0F11)

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

RISC-V_CSR_MARCHID

Address: Operational Base + offset (0x0F12)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	ARCH ID Encoding the base micro-architecture of the hart

RISC-V_CSR_MIMPID

Address: Operational Base + offset (0x0F13)

Bit	Attr	Reset Value	Description
31:0	RO	0x011432a4	YEAR-MONTH-DAY-RELEASE Bit[31]-bit[24]: BCD-coded value of the year Bit[23]-bit[16]: BCD-coded value of the month Bit[16]-bit[8]: BCD-coded value of the day Bit[7]-bit[0]: 8-bit value of intraday release number

RISC-V_CSR_MHARTID

Address: Operational Base + offset (0x0F14)

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

RISC-V_CSR_MSTATUS

Address: Operational Base + offset (0x0300)

Bit	Attr	Reset Value	Description
31:13	RO	0x00000	reserved
12:11	RO	0x3	MPP Previous privilege mode (hardwired to 11)
10:8	RO	0x0	reserved
7	RW	0x1	MPIE Previous global interrupt enable
6:4	RO	0x0	reserved
3	RW	0x0	MIE Global interrupt enable
2:0	RO	0x0	reserved

RISC-V_CSR_MISA

Address: Operational Base + offset (0x0301)

Bit	Attr	Reset Value	Description
31:30	RO	0x1	MXL Machine XLEN (hardwired to 01)
29:24	RO	0x00	reserved
23	RO	0x0	RVX Non-standard extensions

Bit	Attr	Reset Value	Description
22:13	RO	0x000	reserved
12	RO	0x1	RVM Integer Multiply/Divide extension implemented
11:9	RO	0x0	reserved
8	RO	0x1	RVI RV32I base integer instruction set
7:5	RO	0x0	reserved
4	RO	0x0	RVE RV32E base integer instruction set
3	RO	0x0	reserved
2	RO	0x1	RVC Compressed instruction extension implemented
1:0	RO	0x0	reserved

RISC-V_CSR_MIE

Address: Operational Base + offset (0x0304)

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

RISC-V_CSR_MTVEC

Address: Operational Base + offset (0x0305)

Bit	Attr	Reset Value	Description
31:6	RW	0x0008007	BASE Vector base address (upper 26 bits)
5:2	RO	0x0	reserved
1:0	RW	0x0	MODE Vector mode (0-direct mode, 1-vectored mode)

RISC-V_CSR_MSCRATCH

Address: Operational Base + offset (0x0340)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	MSCRATCH Hold a pointer to a machine-mode hart-local context space and swapped with a user register upon entry to an M-mode trap handler.

RISC-V_CSR_MEPC

Address: Operational Base + offset (0x0341)

Bit	Attr	Reset Value	Description
31:1	RW	0x00000000	MEPC Hold the virtual address of the instruction that encountered the exception.
0	RO	0x0	reserved

RISC-V_CSR_MCAUSE

Address: Operational Base + offset (0x0342)

Bit	Attr	Reset Value	Description
31	RW	0x0	INT Interrupt
30:4	RO	0x0000000	reserved
3:0	RW	0x0	EC Exception Code

RISC-V_CSR_MTVAL

Address: Operational Base + offset (0x0343)

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30:0	RW	0x00000000	MTVAL When a trap is taken into M-mode, mtval is written with exception-specific information to assist software in handling the trap.

RISC-V_CSR_MIP

Address: Operational Base + offset (0x0344)

Bit	Attr	Reset Value	Description
31:12	RO	0x00000	reserved
11	RO	0x0	MEIP Machine External Interrupt Pending
10:8	RO	0x0	reserved
7	RO	0x0	MTIP Machine Timer Interrupt Pending
6:4	RO	0x0	reserved
3	RO	0x0	MSIP Machine Software Interrupt Pending
2:0	RO	0x0	reserved

RISC-V_CSR_MCYCLEL

Address: Operational Base + offset (0x0B00)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	MCYCLEL Represent the number of clock cycles(low 32bit) since some arbitrary point of time in the past.

RISC-V_CSR_MCYCLEH

Address: Operational Base + offset (0x0B80)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	MCYCLEH Represent the number of clock cycles(high 32bit) since some arbitrary point of time in the past.

RISC-V_CSR_MINSTRETL

Address: Operational Base + offset (0x0B02)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	MINSTRETL Represent the number of instructions(low 32bit) retired by the core from some arbitrary time in the past.

RISC-V_CSR_MINSTRETH

Address: Operational Base + offset (0x0B82)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	MINSTRETH Represent the number of instructions(low 32bit) retired by the core from some arbitrary time in the past.

RISC-V_CSR_MCOUNTEN

Address: Operational Base + offset (0x07E0)

Bit	Attr	Reset Value	Description
31:3	RO	0x00000000	reserved
2	RW	0x1	IR Enable retired instructions counter
1	RO	0x0	reserved
0	RW	0x1	CY Enable cycle counter

RISC-V_CSR_DBG_SCRATCH

Address: Operational Base + offset (0x07C8)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DSCRATCH Used to exchange data between the core and the debug controller.

RISC-V_CSR_IPIC_CISV

Address: Operational Base + offset (0x0BF0)

Bit	Attr	Reset Value	Description
31:5	RO	0x0000000	reserved
4:0	RO	0x00	CISV Number of the interrupt vector currently in service

RISC-V_CSR_IPIC_CICSR

Address: Operational Base + offset (0x0BF1)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved
1	RW	0x0	IE Interrupt Enable Bit 1'b0: Interrupt disabled 1'b1: Interrupt enabled
0	W1C	0x0	IP Interrupt pending 1'b0: no interrupt 1'b1: Interrupt pending

RISC-V_CSR_IPIC_IPR

Address: Operational Base + offset (0x0BF2)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15:0	W1C	0x0000	PENDING Interrupt vector0 ~ vector15 pending status, each bit represents one vector.

RISC-V_CSR_IPIC_ISVR

Address: Operational Base + offset (0x0BF3)

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

Bit	Attr	Reset Value	Description
15:0	RO	0x0000	SERVICE When corresponding bit is set (1) - this interrupt vector is in service.

RISC-V_CSR_IPIC_EOI

Address: Operational Base + offset (0x0BF4)

Bit	Attr	Reset Value	Description
31:0	WS	0x00000000	END OF INTERRUPT Writing any value to EOI register ends the interrupt which is currently in service, read will return zero.

RISC-V_CSR_IPIC_SOI

Address: Operational Base + offset (0x0BF5)

Bit	Attr	Reset Value	Description
31:0	WS	0x00000000	START OF INTERRUPT Writing any value to SOI activates start of interrupt if one of the following conditions is true: 1. There is at least one pending interrupt with IE and ISR is zero (no interrupts in service) 2. There is at least one pending interrupt with IE and this interrupt has higher priority than the interrupts currently in service.

RISC-V_CSR_IPIC_IDX

Address: Operational Base + offset (0x0BF6)

Bit	Attr	Reset Value	Description
31:4	RO	0x0000000	reserved
3:0	RW	0x0	INDEX Interrupt vector index to access through IPIC_ICSR

RISC-V_CSR_IPIC_ICSR

Address: Operational Base + offset (0x0BF7)

Bit	Attr	Reset Value	Description
31:13	RO	0x00000	reserved
12	RO	0x0	LN External IRQ Line Number assigned to this interrupt vector. This value is always equal to IPIC_IDX, because of the static line to vector mapping.
11:9	RO	0x0	reserved
8	RO	0x0	PRV Privilege mode: hardwired to 11 (machine mode)
7:5	RO	0x0	reserved
4	RW	0x0	IS In Service
3	RW	0x0	INV Line Inversion 1'b0: no inversion 1'b1: line inversion
2	RW	0x0	IM Interrupt Mode 1'b0: Level interrupt 1'b1: Edge interrupt

Bit	Attr	Reset Value	Description
1	RW	0x0	IE Interrupt Enable Bit 1'b0: Interrupt disabled 1'b1: Interrupt enabled
0	W1C	0x0	IP Interrupt pending 1'b0: no interrupt 1'b1: Interrupt pending

3.4.5 INTMUX Registers Summary

Name	Offset	Size	Reset Value	Description
INTMUX_INT_ENABLE_GROUP0	0x0000	W	0x00000000	Interrupt Group0 Enable Register
INTMUX_INT_ENABLE_GROUP1	0x0004	W	0x00000000	Interrupt Group1 Enable Register
INTMUX_INT_ENABLE_GROUP2	0x0008	W	0x00000000	Interrupt Group2 Enable Register
INTMUX_INT_ENABLE_GROUP3	0x000C	W	0x00000000	Interrupt Group3 Enable Register
INTMUX_INT_ENABLE_GROUP4	0x0010	W	0x00000000	Interrupt Group4 Enable Register
INTMUX_INT_ENABLE_GROUP5	0x0014	W	0x00000000	Interrupt Group5 Enable Register
INTMUX_INT_ENABLE_GROUP6	0x0018	W	0x00000000	Interrupt Group6 Enable Register
INTMUX_INT_ENABLE_GROUP7	0x001C	W	0x00000000	Interrupt Group7 Enable Register
INTMUX_INT_ENABLE_GROUP8	0x0020	W	0x00000000	Interrupt Group8 Enable Register
INTMUX_INT_ENABLE_GROUP9	0x0024	W	0x00000000	Interrupt Group9 Enable Register
INTMUX_INT_ENABLE_GROUP10	0x0028	W	0x00000000	Interrupt Group10 Enable Register
INTMUX_INT_ENABLE_GROUP11	0x002C	W	0x00000000	Interrupt Group11 Enable Register
INTMUX_INT_ENABLE_GROUP12	0x0030	W	0x00000000	Interrupt Group12 Enable Register
INTMUX_INT_ENABLE_GROUP13	0x0034	W	0x00000000	Interrupt Group13 Enable Register
INTMUX_INT_ENABLE_GROUP14	0x0038	W	0x00000000	Interrupt Group14 Enable Register
INTMUX_INT_ENABLE_GROUP15	0x003C	W	0x00000000	Interrupt Group15 Enable Register
INTMUX_INT_FLAG_GROUP0	0x0080	W	0x00000000	Interrupt Group0 Flag Register
INTMUX_INT_FLAG_GROUP1	0x0084	W	0x00000000	Interrupt Group1 Flag Register
INTMUX_INT_FLAG_GROUP2	0x0088	W	0x00000000	Interrupt Group2 Flag Register
INTMUX_INT_FLAG_GROUP3	0x008C	W	0x00000000	Interrupt Group3 Flag Register
INTMUX_INT_FLAG_GROUP4	0x0090	W	0x00000000	Interrupt Group4 Flag Register

Name	Offset	Size	Reset Value	Description
INTMUX_INT_FLAG_GROUP5	0x0094	W	0x00000000	Interrupt Group5 Flag Register
INTMUX_INT_FLAG_GROUP6	0x0098	W	0x00000000	Interrupt Group6 Flag Register
INTMUX_INT_FLAG_GROUP7	0x009C	W	0x00000000	Interrupt Group7 Flag Register
INTMUX_INT_FLAG_GROUP8	0x00A0	W	0x00000000	Interrupt Group8 Flag Register
INTMUX_INT_FLAG_GROUP9	0x00A4	W	0x00000000	Interrupt Group9 Flag Register
INTMUX_INT_FLAG_GROUP10	0x00A8	W	0x00000000	Interrupt Group10 Flag Register
INTMUX_INT_FLAG_GROUP11	0x00AC	W	0x00000000	Interrupt Group11 Flag Register
INTMUX_INT_FLAG_GROUP12	0x00B0	W	0x00000000	Interrupt Group12 Flag Register
INTMUX_INT_FLAG_GROUP13	0x00B4	W	0x00000000	Interrupt Group13 Flag Register
INTMUX_INT_FLAG_GROUP14	0x00B8	W	0x00000000	Interrupt Group14 Flag Register
INTMUX_INT_FLAG_GROUP15	0x00BC	W	0x00000000	Interrupt Group15 Flag Register

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

3.4.6 INTMUX Detail Registers Description

INTMUX_INT_ENABLE_GROUP0

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x00	ENABLE Each bit represents one interrupt enable bit. 1'b0: Interrupt disabled 1'b1: Interrupt enabled

INTMUX_INT_ENABLE_GROUP1

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x00	ENABLE Each bit represents one interrupt enable bit. 1'b0: Interrupt disabled 1'b1: Interrupt enabled

INTMUX_INT_ENABLE_GROUP2

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x00	ENABLE Each bit represents one interrupt enable bit. 1'b0: Interrupt disabled 1'b1: Interrupt enabled

INTMUX_INT_ENABLE_GROUP3

Address: Operational Base + offset (0x000C)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x00	ENABLE Each bit represents one interrupt enable bit. 1'b0: Interrupt disabled 1'b1: Interrupt enabled

INTMUX_INT_ENABLE_GROUP4

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x00	ENABLE Each bit represents one interrupt enable bit. 1'b0: Interrupt disabled 1'b1: Interrupt enabled

INTMUX_INT_ENABLE_GROUP5

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x00	ENABLE Each bit represents one interrupt enable bit. 1'b0: Interrupt disabled 1'b1: Interrupt enabled

INTMUX_INT_ENABLE_GROUP6

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x00	ENABLE Each bit represents one interrupt enable bit. 1'b0: Interrupt disabled 1'b1: Interrupt enabled

INTMUX_INT_ENABLE_GROUP7

Address: Operational Base + offset (0x001C)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x00	ENABLE Each bit represents one interrupt enable bit. 1'b0: Interrupt disabled 1'b1: Interrupt enabled

INTMUX_INT_ENABLE_GROUP8

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x00	ENABLE Each bit represents one interrupt enable bit. 1'b0: Interrupt disabled 1'b1: Interrupt enabled

INTMUX_INT_ENABLE_GROUP9

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x00	ENABLE Each bit represents one interrupt enable bit. 1'b0: Interrupt disabled 1'b1: Interrupt enabled

INTMUX_INT_ENABLE_GROUP10

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x00	ENABLE Each bit represents one interrupt enable bit. 1'b0: Interrupt disabled 1'b1: Interrupt enabled

INTMUX_INT_ENABLE_GROUP11

Address: Operational Base + offset (0x002C)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x00	ENABLE Each bit represents one interrupt enable bit. 1'b0: Interrupt disabled 1'b1: Interrupt enabled

INTMUX_INT_ENABLE_GROUP12

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x00	ENABLE Each bit represents one interrupt enable bit. 1'b0: Interrupt disabled 1'b1: Interrupt enabled

INTMUX_INT_ENABLE_GROUP13

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x00	ENABLE Each bit represents one interrupt enable bit. 1'b0: Interrupt disabled 1'b1: Interrupt enabled

INTMUX_INT_ENABLE_GROUP14

Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x00	ENABLE Each bit represents one interrupt enable bit. 1'b0: Interrupt disabled 1'b1: Interrupt enabled

INTMUX_INT_ENABLE_GROUP15

Address: Operational Base + offset (0x003C)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x00	ENABLE Each bit represents one interrupt enable bit. 1'b0: Interrupt disabled 1'b1: Interrupt enabled

INTMUX_INT_FLAG_GROUP0

Address: Operational Base + offset (0x0080)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RO	0x00	FLAG Each bit represents one interrupt is in service. 1'b0: Interrupt in service 1'b1: Interrupt out of service

INTMUX_INT_FLAG_GROUP1

Address: Operational Base + offset (0x0084)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RO	0x00	FLAG Each bit represents one interrupt is in service. 1'b0: Interrupt in service 1'b1: Interrupt out of service

INTMUX_INT_FLAG_GROUP2

Address: Operational Base + offset (0x0088)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RO	0x00	FLAG Each bit represents one interrupt is in service. 1'b0: Interrupt in service 1'b1: Interrupt out of service

INTMUX_INT_FLAG_GROUP3

Address: Operational Base + offset (0x008C)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RO	0x00	FLAG Each bit represents one interrupt is in service. 1'b0: Interrupt in service 1'b1: Interrupt out of service

INTMUX_INT_FLAG_GROUP4

Address: Operational Base + offset (0x0090)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RO	0x00	FLAG Each bit represents one interrupt is in service. 1'b0: Interrupt in service 1'b1: Interrupt out of service

INTMUX_INT_FLAG_GROUP5

Address: Operational Base + offset (0x0094)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RO	0x00	FLAG Each bit represents one interrupt is in service. 1'b0: Interrupt in service 1'b1: Interrupt out of service

INTMUX_INT_FLAG_GROUP6

Address: Operational Base + offset (0x0098)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RO	0x00	FLAG Each bit represents one interrupt is in service. 1'b0: Interrupt in service 1'b1: Interrupt out of service

INTMUX_INT_FLAG_GROUP7

Address: Operational Base + offset (0x009C)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RO	0x00	FLAG Each bit represents one interrupt is in service. 1'b0: Interrupt in service 1'b1: Interrupt out of service

INTMUX_INT_FLAG_GROUP8

Address: Operational Base + offset (0x00A0)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RO	0x00	FLAG Each bit represents one interrupt is in service. 1'b0: Interrupt in service 1'b1: Interrupt out of service

INTMUX_INT_FLAG_GROUP9

Address: Operational Base + offset (0x00A4)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RO	0x00	FLAG Each bit represents one interrupt is in service. 1'b0: Interrupt in service 1'b1: Interrupt out of service

INTMUX_INT_FLAG_GROUP10

Address: Operational Base + offset (0x00A8)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RO	0x00	FLAG Each bit represents one interrupt is in service. 1'b0: Interrupt in service 1'b1: Interrupt out of service

INTMUX_INT_FLAG_GROUP11

Address: Operational Base + offset (0x00AC)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RO	0x00	FLAG Each bit represents one interrupt is in service. 1'b0: Interrupt in service 1'b1: Interrupt out of service

INTMUX_INT_FLAG_GROUP12

Address: Operational Base + offset (0x00B0)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RO	0x00	FLAG Each bit represents one interrupt is in service. 1'b0: Interrupt in service 1'b1: Interrupt out of service

INTMUX_INT_FLAG_GROUP13

Address: Operational Base + offset (0x00B4)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RO	0x00	FLAG Each bit represents one interrupt is in service. 1'b0: Interrupt in service 1'b1: Interrupt out of service

INTMUX_INT_FLAG_GROUP14

Address: Operational Base + offset (0x00B8)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RO	0x00	FLAG Each bit represents one interrupt is in service. 1'b0: Interrupt in service 1'b1: Interrupt out of service

INTMUX_INT_FLAG_GROUP15

Address: Operational Base + offset (0x00BC)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RO	0x00	FLAG Each bit represents one interrupt is in service. 1'b0: Interrupt in service 1'b1: Interrupt out of service

3.5 Interface Description

Table 3-1 Serial Wire Debug Interface Description

Module Pin	Direction	Pin Name	IOMUX Setting
TCK	I	SDMMC0_D2/UART3_RX_M1/A7_JTAG_TCK_M0/RISC-V_JTAG_TCK/GPIO1_A6_u	GRF_GPIO1A_IOMUX_SEL_H[11:8]=4'b0100
TMS	I/O	SDMMC0_D3/UART3_TX_M1/A7_JTAG_TMS_M0/RISC-V_JTAG_TMS/GPIO1_A7_u	GRF_GPIO1A_IOMUX_SEL_H[15:12]=4'b0100

Module Pin	Direction	Pin Name	IOMUX Setting
TRSTN	I	SDMMC0_D1/TEST_CLK0_OUT/UART2_TX_M0/RISC-V_JTAG_TRSTn/GPIO1_A5_u	GRF_GPIO1A_IOMUX_SEL_H[7:4]=4'b0100
TDO	O	SDMMC0_CLK/UART3_RTSN_M1/RISC-V_JTAG_TDO/GPIO1_B0_u	GRF_GPIO1B_IOMUX_SEL_L[3:0]=4'b0100
TDI	I	SDMMC0_CMD/UART3_CTSN_M1/RISC-V_JTAG_TDI/GPIO1_B1_u	GRF_GPIO1B_IOMUX_SEL_L[7:4]=4'b0100

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

3.6 Application Notes

- How to boot?
  Currently, the RISC-V core boot address is configurable, its default value is 32'h0000_0000, and the default value of trap vector base address is 32'h0000_01c0, you can configure SCR1_BOOT_ADDR of SGRF before de-assert its reset. if you change the boot address, don't forget to configure the RISC-V_CSR_MTVEC register by system instructions.
- Are 16 IRQ lines enough?
  Actually, there are 123 IRQ lines in full chip, and all of these IRQ lines are connected to RISC-V core, there is a 128to16 MUX for these IRQ lines before entering to IPIC of RISC-V core. Each 8 IRQ lines is regard as one group and connect to one IRQ line of RISC-V core. Not only We should configure IPIC, but also configure INTMUX before activates IRQ.

3.7 Referenced documents

- [[[1]]] The RISC-V Instruction Set Manual Volume I: User-Level ISA
- [[[2]]] The RISC-V Instruction Set Manual Volume II: Privileged Architecture
- [[[3]]] RISC-V External Debug Support, Version 0.132

Chapter 4 Embedded SRAM

4.1 Overview

There are two embedded SRAMs, SYSTEM_SRAM and PMU_SRAM.

4.1.1 Features supported

- SYSTEM_SRAM
  - Provide 64KB access space
  - Support security and non-security access
  - Secure or non-secure space is software programmable

- PMU_SRAM
  - Provide 8KB access space
  - Support secure access only

4.2 Block Diagram

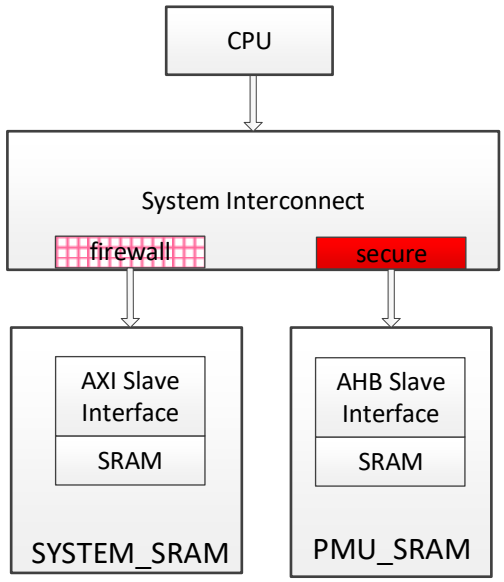


Fig. 4-1 Embedded SRAM block diagram

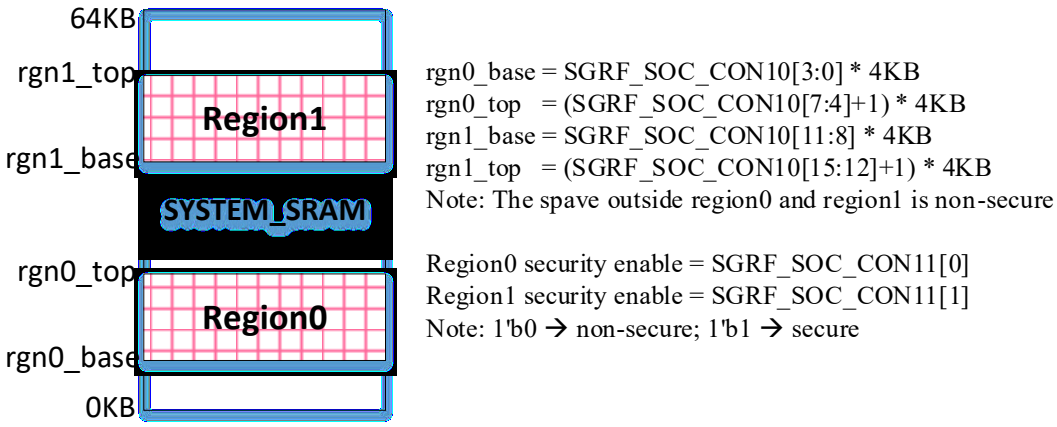


Fig. 4-2 SYSTEM_SRAM Security Configuration

4.3 Function Description

4.3.1 AXI slave interface of SYSTEM_SRAM

The AXI slave interface is bridge which translates AXI bus access to SRAM interface of

SYSTEM_SRAM.

4.3.2 AHB slave interface of PMU_SRAM

The AHB slave interface is bridge which translates AHB bus access to SRAM interface of PMU_SRAM.

4.3.3 Embedded SRAM access path

The SYSTEM_SRAM can only be accessed by Cortex-A7, MCU, CRYPTO, DCF, DECOM, DMAC and USB2_OTG. The PMU_SRAM can only be accessed by Cortex-A7, MCU, CRYPTO and DMAC.

Chapter 5 Clock & Reset Unit (CRU)

5.1 Overview

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

- CRU_PMU, used for always on system, with base address 0xFF480000
- CRU, used for general system except always on system, with base address 0xFF490000

The CRU supports the following features:

- Compliance with AMBA APB interface
- Embedded with 5 PLLs
- Flexible selection of clock source
- Support dividing clock separately
- Support gating clock separately
- Support software reset each module separately

5.2 Block Diagram

The CRU comprises with:

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

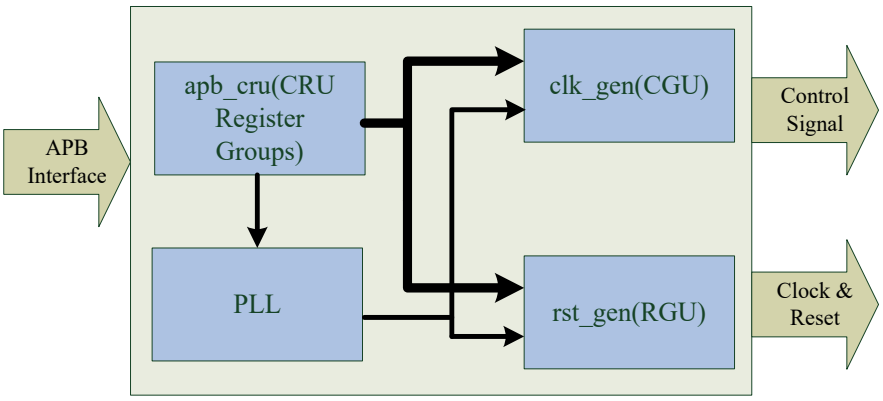


Fig. 5-1 CRU Block Diagram

5.3 Function Description

5.3.1 System Clock Solution

There are 5 PLLs in RV1109/RV1126: APLL, HPLL, DPLL, CPLL and GPLL. Each PLL can only receive 24MHz oscillator as input reference clock and can be set to three work modes: normal mode, slow mode and deep slow mode. When power on or changing PLL setting, we must program PLL into slow mode or deep slow mode.
To maximize the flexibility, some of clocks can select divider source from multiple PLLs. To provide some specific frequency, another solution is integrated: fractional divider. Divfree50 divider and divfreeNP5 divider are also provided for some modules. All clocks can be gated by software.
The basic units for clock generation are:

- Gating
- MUX(multiplexer)
- Divfree(Glitch free divider)
  - clk_out_freq=clk_in_freq/divisor
  - When divisor is even, the clock duty cycle of clk_out is 50%

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

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

5.3.2 System Reset Solution

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

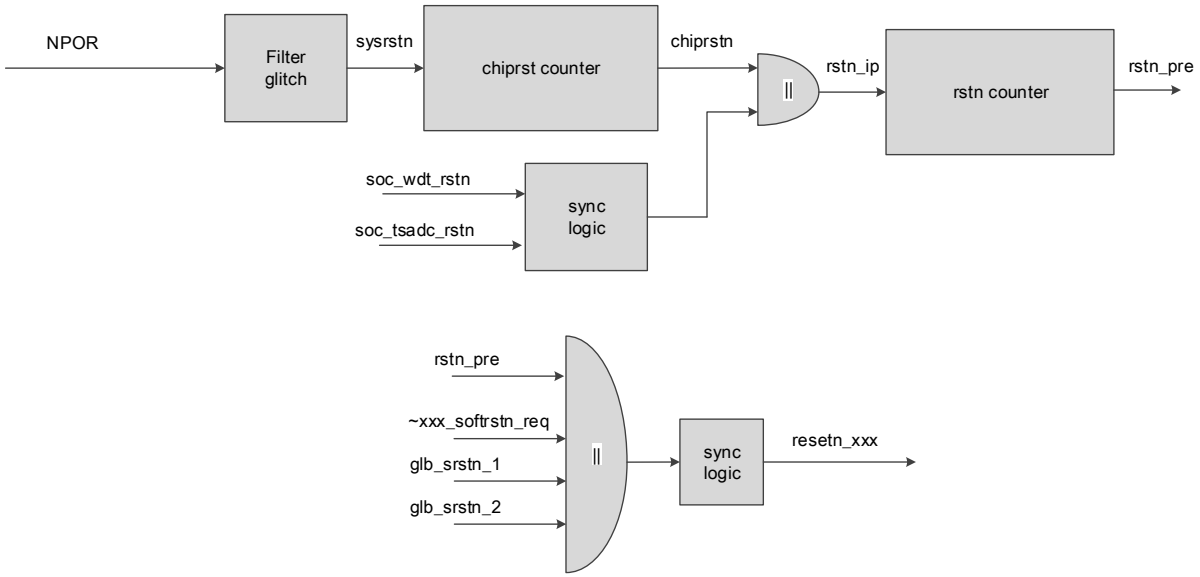


Fig. 5-2 Reset Architecture Diagram

Reset source of each reset signal includes:

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

5.3.3 PLL Introduction

The PLLs inside RV1109/RV1126 output clock's frequency up to 6.4GHz. The PLL is a general purpose, high-performance PLL-based clock generator. The PLL is a multi-function, general purpose frequency synthesizer. Ultra-wide input and output ranges along with best-in-class jitter performance allow the PLL to be used for almost any clocking application. With excellent supply noise immunity, the PLL is ideal for use in noisy mixed signal SoC environments.

The PLL supports the following features:

- Input frequency range: 5MHz to 1200MHz for integer mode and 10MHz to 1200MHz for fractional mode
- PFD minimum reference frequency range: 5MHz for integer mode and 10MHz for fractional mode
- Output frequency range: 16MHz to 6.4GHz
- VCO output frequency range: 1.6GHz to 6.4GHz
- 24 bit fractional accuracy, and fractional mode jitter performance to nearly match integer mode performance.
- 4:1 VCO frequency range allows PLL to be optimized for minimum jitter or minimum

power
- Isolated analog supply (1.8V) allows for excellent supply rejection in noisy SoC applications
- Lock detect signal indicates when frequency lock has been achieved
- Offset calibration to reduce jitter

PLL block diagram is shown below.

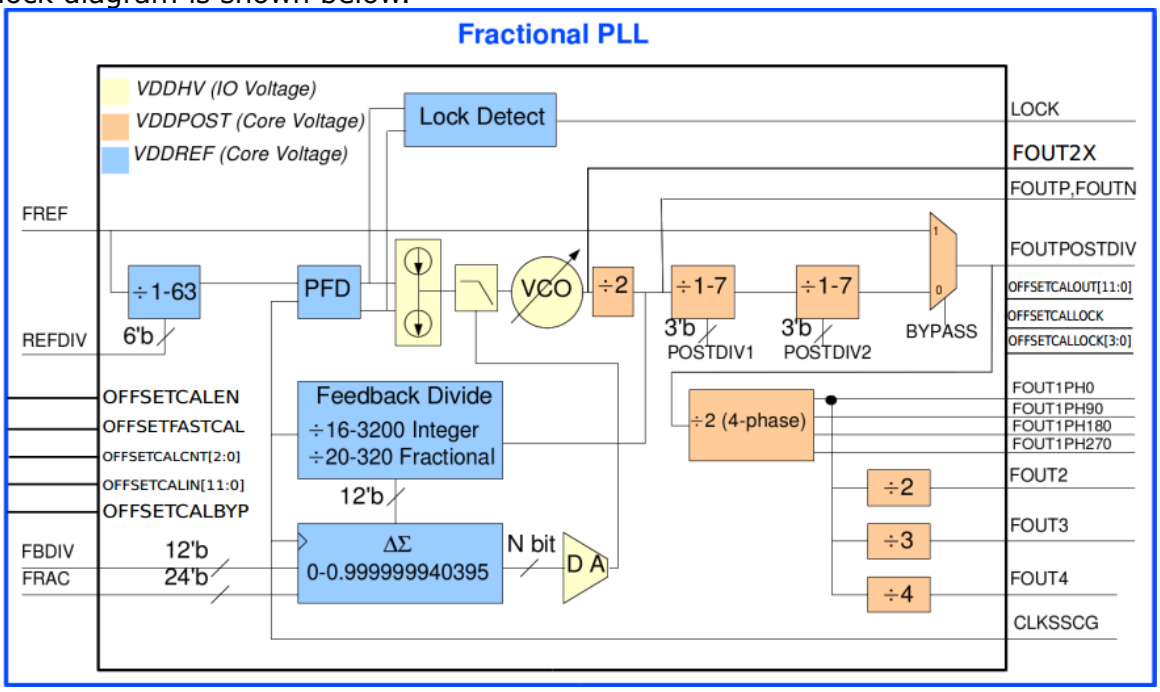


Fig. 5-3 PLL Block Diagram

5.3.3.1 PLL Offset Calibration

Offset calibration is included in PLL in order to reduce jitter resulting from systematic mismatch in the PLL's forward path.

The total systematic mismatch of the PLL consists of many contributors including device mismatch and leakage. This total mismatch is represented as φ_{err} at the input of the PFD. This is the intrinsic offset of the PLL. φ_c is the offset correction implemented with the offset calibration block. When settled, φ_c will be approximately equal to φ_{err}, resulting in minimum jitter.

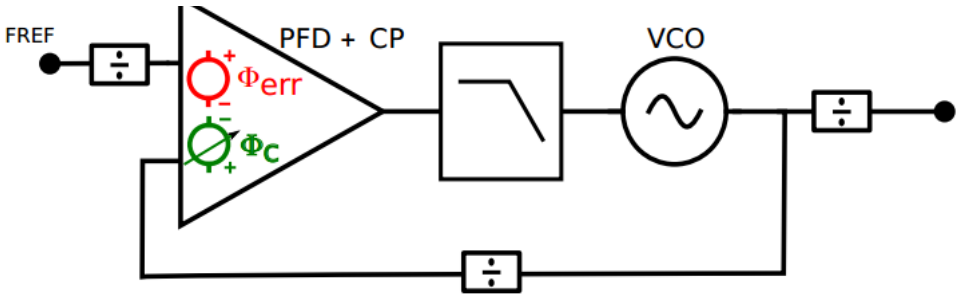


Fig. 5-4 PLL Offset Calibration

CALCODE is the internal offset calibration setting and can be monitored at CALOUT. The optimal CALCODE is available at CALOUT when CALLOCK is asserted. The offset calibration is only enabled if CALEN==1. The starting condition of the calibration process is set by CALIN. CALIN provides a mechanism for initializing the calibration process. If the CALIN is preset to previously calibrated CALCODE prior to enabling the offset calibration, the system is able to start at φ_c. The calibration process steps through CALCODE, which is the internal offset calibration setting, to find the optimal φc that corrects for φ_{err}.

The CALCNT input sets the update rate of the offset calibration. Smaller values of CALCNT result in faster CALCODE updates.

When FASTCAL=1, the offset calibration process steps through the CALCODE quickly and reaches the noise floor within 32*2^(5+CALCNT) PFD cycles. The calibration process continues to converge towards optimal φ_c, and asserts CALLOCK when this state is reached. The state of CALCODE can be monitored at CALOUT and the optimal CALCODE is available

when CALLOCK=1.
If the CALIN is preset to previously calibrated CALCODE prior to enabling the offset calibration, the system is able to start within the noise floor region and continue to converge towards optimal φ_c.
If CALBYPASS=1, the CALCODE is set by CALIN. If calibration has already been run, the optimal CALCODE can be stored and used as an input when the PLL is enabled in the future. Future calibration would begin with the optimal CALCODE, allowing the IP to start with no systematic contributions to the period jitter.

5.4 CRU_PMU Register Description

5.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
CRU_PMU_MODE_CON00	0x0000	W	0x00000000	Mode control register 0
CRU_PMU_GPLL_CON0	0x0010	W	0x00002064	GPLL control register 0
CRU_PMU_GPLL_CON1	0x0014	W	0x00001081	GPLL control register 1
CRU_PMU_GPLL_CON2	0x0018	W	0x00000000	GPLL control register 2
CRU_PMU_GPLL_CON3	0x001C	W	0x00000007	GPLL control register 3
CRU_PMU_GPLL_CON4	0x0020	W	0x00007F00	GPLL control register 4
CRU_PMU_GPLL_CON5	0x0024	W	0x00000000	GPLL control register 5
CRU_PMU_GPLL_CON6	0x0028	W	0x00000000	GPLL control register 6
CRU_PMU_GPLL_OFFSETCAL_STATUS	0x0030	W	0x00000000	GPLL offset calibration status register
CRU_PMU_CLKSEL_CON00	0x0100	W	0x00000000	Internal clock select and division register 0
CRU_PMU_CLKSEL_CON01	0x0104	W	0x0000000B	Internal clock select and division register 1
CRU_PMU_CLKSEL_CON02	0x0108	W	0x0000000B	Internal clock select and division register 2
CRU_PMU_CLKSEL_CON03	0x010C	W	0x0000000B	Internal clock select and division register 3
CRU_PMU_CLKSEL_CON04	0x0110	W	0x00000800	Internal clock select and division register 4
CRU_PMU_CLKSEL_CON05	0x0114	W	0x00000000	Internal clock select and division register 5
CRU_PMU_CLKSEL_CON06	0x0118	W	0x00000000	Internal clock select and division register 6
CRU_PMU_CLKSEL_CON07	0x011C	W	0x000063F1	Internal clock select and division register 7
CRU_PMU_CLKSEL_CON08	0x0120	W	0x00000000	Internal clock select and division register 8
CRU_PMU_CLKSEL_CON09	0x0124	W	0x0000000B	Internal clock select and division register 9
CRU_PMU_CLKSEL_CON12	0x0130	W	0x00000005	Internal clock select and division register 12
CRU_PMU_CLKSEL_CON13	0x0134	W	0x00000000	Internal clock select and division register 13
CRU_PMU_GATE_CON00	0x0180	W	0x00000000	Internal clock gate and division register 0
CRU_PMU_GATE_CON01	0x0184	W	0x00000000	Internal clock gate and division register 1
CRU_PMU_GATE_CON02	0x0188	W	0x00000000	Internal clock gate and division register 2
CRU_PMU_SOFTRST_CON00	0x0200	W	0x00000000	Internal clock reset register 0

Name	Offset	Size	Reset Value	Description
CRU_PMU_SOFTRST_CON01	0x0204	W	0x00000000	Internal clock reset register 1

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

5.4.2 Detail Register Description

CRU_PMU_MODE_CON00

Address: Operational Base + offset (0x0000)

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

CRU_PMU_GPLL_CON0

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	bypass PLL bypass. FREF bypasses PLL to FOUTPOSTDIV. 1'b0: No bypass 1'b1: Bypass
14:12	RW	0x2	postdiv1 First post divide value. Valid settings are [1-7].
11:0	RW	0x064	fbdiv Feedback divide value. Valid divider settings are: [16, 2500] in integer mode [20, 500] in fractional mode Tips: No plus one operation

CRU_PMU_GPLL_CON1

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	pllpdsel PLL global power down source selection. If pllpdsel=1, PLL can be power down only by pllpd1, otherwise PLL is power down when any one of refdiv/fbdiv/fracdiv is changed or pllpd0 is asserted.
14	RW	0x0	pllpd1 PLL global power down request 1. 1'b0: No power down 1'b1: Power down

Bit	Attr	Reset Value	Description
13	RW	0x0	pllpd0 PLL global power down request 0. 1'b0: No power down 1'b1: Power down
12	RW	0x1	dsmpd PLL delta sigma modulator enable. 1'b0: Modulator is enable 1'b1: Modulator is disabled
11	RO	0x0	reserved
10	RW	0x0	pll_lock PLL lock status. 1'b0: Unlock 1'b1: Lock
9	RO	0x0	reserved
8:6	RW	0x2	postdiv2 Second post divide value. Valid settings are [1-7].
5:0	RW	0x01	refdiv Reference clock divide value, Valid settings are [1-63].

CRU_PMU_GPLL_CON2

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27	RW	0x0	fout4phasepd Power down 4-phase clocks. 1'b0: No power down 1'b1: Power down
26	RW	0x0	foutvcopd Power down buffered VCO clock. 1'b0: No power down 1'b1: Power down
25	RW	0x0	foutpostdivpd Power down all outputs except for buffered VCO clock. 1'b0: No power down 1'b1: Power down
24	RW	0x0	dacpd Power down quantization noise cancellation DAC. 1'b0: No power down 1'b1: Power down
23:0	RW	0x000000	fracdiv Fractional part of feedback divide, fraction = fracdiv/2^24.

CRU_PMU_GPLL_CON3

Address: Operational Base + offset (0x001C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:13	RO	0x0	reserved
12:8	RW	0x00	ssmod_spread Spread amplitude. % = 0.1 * SPREAD[4: 0]
7:4	RW	0x0	ssmod_divval Divider required to set the modulation frequency

Bit	Attr	Reset Value	Description
3	RW	0x0	ssmod_downspread Select center spread or down spread. 1'b0: Down spread 1'b1: Center spread
2	RW	0x1	ssmod_reset Reset modulator state. 1'b0: No reset 1'b1: Reset
1	RW	0x1	ssmod_disable_sscg Bypass SSMOD by module. 1'b0: No bypass 1'b1: Bypass
0	RW	0x1	ssmod_bp Bypass SSMOD by integration. 1'b0: No bypass 1'b1: Bypass

CRU_PMU_GPLL_CON4

Address: Operational Base + offset (0x0020)

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

CRU_PMU_GPLL_CON5

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:4	RW	0x000	offsetcal_in offsetcal_byp=1'b0: Initial condition for offset calibration logic. offsetcal_byp=1'b1: Override value for offset calibration. It is a signed integer with positive values delaying the reset of the faster path, and negative values delaying the reset of the slower path. 5'b0 is the minimum value, with each count increasing the reset time by one buffer delay. If offsetcal_en=1, this can be used to force a offset correction value based on a previous readout of offsetcal_out[11:0]. If offsetcal_byp=1 this value is forced directly into the calibration logic. If offsetcal_byp=0 this is the initial condition for the calibration sequence.

Bit	Attr	Reset Value	Description
3	RW	0x0	offsetcal_fast Offset fast calibration enable. Set this to 1 for initial calibration if an initial value is not already known. Should be set to 0 for normal operation.
2	RW	0x0	offsetcal_en Offset calibration enable to actively adjust for input offset. 1'b0: Offset calibration is disabled. Static phase offset is determined by analog matching only. 1'b1: Offset calibration is enabled. Static phase offset is adjusted by sensing phase at the input.
1	RW	0x0	offsetcal_byp Offset calibration bypass. 1'b0: Use the offset calibration output(when offsetcal_en=1) to set the phase correction 1'b1: Use the offsetcal_in[11:0] value (when offsetcal_en=1) to set the phase correction
0	RW	0x0	fout2xpd Power down 2X clocks. 1'b0: No power down 1'b1: Power down

CRU_PMU_GPLL_CON6

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:3	RO	0x0000	reserved
2:0	RW	0x0	offsetcal_cnt Programmable counter for offset calibration loop. Selects the number of PFD edges to wait after each offset calibration step. Count is defined as 2^(offsetcal_cnt+4).

CRU_PMU_GPLL_OFFSETCAL_STATUS

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:17	RO	0x0000	reserved
16	RW	0x0	offsetcal_lock Asserted when phase lock is achieved when offsetcal_en is set to 1'b1
15:12	RO	0x0	reserved
11:0	RW	0x000	offsetcal_out This is the output of either the offset calibration block (if offsetcal_byp=0) or a buffered version of offsetcal_in[11:0] (if offsetcal_byp=1). It can be used to read out the phase calibration state to use as an override value so that offset calibration can be bypassed for faster locking. The value changes on the rising edge of FREF, so it can be clocked out on the falling edge of FREF.

CRU_PMU_CLKSEL_CON00

Address: Operational Base + offset (0x0100)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:9	RO	0x00	reserved
8:7	RW	0x0	clk_deepslow_sel clk_deepslow clock mux. 2'b00: clk_pmupvtm_divout 2'b01: xin_osc1_32k 2'b10: clk_osc0_div32k Others: Reserved
6:5	RO	0x0	reserved
4:0	RW	0x00	xin_osc0_div_div Divide xin_osc0_div by (div_con + 1)

CRU_PMU_CLKSEL_CON01

Address: Operational Base + offset (0x0104)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:5	RO	0x000	reserved
4:0	RW	0x0b	pclk_pdpmu_pre_div Divide pclk_pdpmu_pre by (div_con + 1)

CRU_PMU_CLKSEL_CON02

Address: Operational Base + offset (0x0108)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:7	RO	0x000	reserved
6:0	RW	0x0b	clk_i2c0_div Divide clk_i2c0 by (div_con + 1)

CRU_PMU_CLKSEL_CON03

Address: Operational Base + offset (0x010C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:7	RO	0x000	reserved
6:0	RW	0x0b	clk_i2c2_div Divide clk_i2c2 by (div_con + 1)

CRU_PMU_CLKSEL_CON04

Address: Operational Base + offset (0x0110)

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

Bit	Attr	Reset Value	Description
15:12	RO	0x0	reserved
11:10	RW	0x2	sclk_uart1_sel sclk_uart1 clock mux. 2'b00: sclk_uart1_div 2'b01: sclk_uart1_fracdiv 2'b10: xin_osc0_func Others: Reserved
9:8	RW	0x0	sclk_uart1_div_sel sclk_uart1_div clock mux. 2'b00: clk_gpll_mux 2'b01: usbphy_clk480m_mux 2'b10: clk_cpll_mux 2'b11: xin_osc0_func
7	RO	0x0	reserved
6:0	RW	0x00	sclk_uart1_div_div Divide sclk_uart1_div by (div_con + 1)

CRU_PMU_CLKSEL_CON05

Address: Operational Base + offset (0x0114)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sclk_uart1_fracdiv_div sclk_uart1_fracdiv fraction division register. High 16-bit for numerator Low 16-bit for denominator

CRU_PMU_CLKSEL_CON06

Address: Operational Base + offset (0x0118)

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

CRU_PMU_CLKSEL_CON07

Address: Operational Base + offset (0x011C)

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

Bit	Attr	Reset Value	Description
15	RW	0x0	clk_mipidsiphy_ref_sel clk_mipidsiphy_ref clock mux. 1'b0: clk_ref24m 1'b1: xin_osc0_mipidsiphyref_g
14:8	RW	0x63	clk_ref12m_div Divide clk_ref12m by (div_con + 1)
7	RW	0x1	clk_usbphy_host_ref_sel clk_usbphy_host_ref clock mux. 1'b0: clk_ref12m 1'b1: xin_osc0_div2_usbphyref_host
6	RW	0x1	clk_usbphy_otg_ref_sel clk_usbphy_otg_ref clock mux. 1'b0: clk_ref12m 1'b1: xin_osc0_div2_usbphyref_otg
5:0	RW	0x31	clk_ref24m_div Divide clk_ref24m by (div_con + 1)

CRU_PMU_CLKSEL_CON08

Address: Operational Base + offset (0x0120)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	dbclk_gpio0_sel dbclk_gpio0 clock mux. 1'b0: xin_osc0_func 1'b1: clk_deepslow
14:0	RO	0x0000	reserved

CRU_PMU_CLKSEL_CON09

Address: Operational Base + offset (0x0124)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x0	clk_spi0_sel clk_spi0 clock mux. 1'b0: clk_gpll_mux 1'b1: xin_osc0_func
6:0	RW	0x0b	clk_spi0_div Divide clk_spi0 by (div_con + 1)

CRU_PMU_CLKSEL_CON12

Address: Operational Base + offset (0x0130)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:9	RO	0x00	reserved

Bit	Attr	Reset Value	Description
8	RW	0x0	clk_wifi_sel clk_wifi clock mux. 1'b0: clk_wifi_osc0 1'b1: clk_wifi_div
7:6	RO	0x0	reserved
5:0	RW	0x05	clk_wifi_div_div Divide clk_wifi_div by (div_con + 1)

CRU_PMU_CLKSEL_CON13

Address: Operational Base + offset (0x0134)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	clk_osc0_div32k_div clk_osc0_div32k fraction division register. High 16-bit for numerator Low 16-bit for denominator

CRU_PMU_GATE_CON00

Address: Operational Base + offset (0x0180)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	clk_pmu_en clk_pmu clock gating control. When high, disable clock.
14	RW	0x0	sclk_uart1_en sclk_uart1 clock gating control. When high, disable clock.
13	RW	0x0	sclk_uart1_fracdiv_en sclk_uart1_fracdiv clock gating control. When high, disable clock.
12	RW	0x0	sclk_uart1_div_en sclk_uart1_div 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	clk_i2c2_en clk_i2c2 clock gating control. When high, disable clock.
9	RW	0x0	pclk_i2c2_en pclk_i2c2 clock gating control. When high, disable clock.
8	RO	0x0	reserved
7	RW	0x0	pclk_scrkeygen_en pclk_scrkeygen clock gating control. When high, disable clock.
6	RW	0x0	clk_i2c0_en clk_i2c0 clock gating control. When high, disable clock.
5	RW	0x0	pclk_i2c0_en pclk_i2c0 clock gating control. When high, disable clock.

Bit	Attr	Reset Value	Description
4	RW	0x0	pclk_pmusgrf_en pclk_pmusgrf clock gating control. When high, disable clock.
3	RO	0x0	reserved
2	RW	0x0	pclk_pdpmu_biu_en pclk_pdpmu_biu clock gating control. When high, disable clock.
1	RW	0x0	pclk_pmu_en pclk_pmu clock gating control. When high, disable clock.
0	RW	0x0	pclk_pdpmu_en pclk_pdpmu_pre clock gating control. When high, disable clock.

CRU_PMU_GATE_CON01

Address: Operational Base + offset (0x0184)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	clk_ref12m_en clk_ref12m clock gating control. When high, disable clock.
14	RW	0x0	clk_ref24m_en clk_ref24m clock gating control. When high, disable clock.
13	RW	0x0	pclk_pmugrf_en pclk_pmugrf clock gating control. When high, disable clock.
12	RW	0x0	clk_spi0_en clk_spi0 clock gating control. When high, disable clock.
11	RW	0x0	pclk_spi0_en pclk_spi0 clock gating control. When high, disable clock.
10	RW	0x0	dbclk_gpio0_en dbclk_gpio0 clock gating control. When high, disable clock.
9	RW	0x0	pclk_gpio0_en pclk_gpio0 clock gating control. When high, disable clock.
8	RW	0x0	xin_osc0_dsiphy_ref_en xin_osc0_mipidsiphy_ref clock gating control. When high, disable clock.
7	RW	0x0	xin_osc0_hostphy_ref_en xin_osc0_div2_hostphy_ref clock gating control. When high, disable clock.
6	RW	0x0	xin_osc0_otgphy_ref_en xin_osc0_div2_otgphy_ref clock gating control. When high, disable clock.
5	RW	0x0	clk_capture_pwm1_en clk_capture_pwm1_ndft clock gating control. When high, disable clock.

Bit	Attr	Reset Value	Description
4	RW	0x0	clk_pwm1_en clk_pwm1 clock gating control. When high, disable clock.
3	RW	0x0	pclk_pwm1_en pclk_pwm1 clock gating control. When high, disable clock.
2	RW	0x0	clk_capture_pwm0_en clk_capture_pwm0_ndft clock gating control. When high, disable clock.
1	RW	0x0	clk_pwm0_en clk_pwm0 clock gating control. When high, disable clock.
0	RW	0x0	pclk_pwm0_en pclk_pwm0 clock gating control. When high, disable clock.

CRU_PMU_GATE_CON02

Address: Operational Base + offset (0x0188)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RO	0x0	reserved
11	RW	0x0	clk_wifi_osc0_en clk_wifi_osc0 clock gating control. When high, disable clock.
10	RW	0x0	clk_wifi_div_en clk_wifi_div clock gating control. When high, disable clock.
9	RW	0x0	clk_osc0_div32k_en clk_osc0_div32k clock gating control. When high, disable clock.
8	RW	0x0	xin_osc0_div_en xin_osc0_div clock gating control. When high, disable clock.
7	RW	0x0	clk_core_pmupvtm_en clk_core_pmupvtm clock gating control. When high, disable clock.
6	RW	0x0	pclk_pmupvtm_en pclk_pmupvtm clock gating control. When high, disable clock.
5	RW	0x0	clk_pmupvtm_en clk_pmupvtm clock gating control. When high, disable clock.
4	RW	0x0	pclk_pmucru_en pclk_pmucru clock gating control. When high, disable clock.
3:1	RO	0x0	reserved
0	RW	0x0	pclk_chipverotp_en pclk_chipverotp clock gating control. When high, disable clock.

CRU_PMU_SOFTRST_CON00

Address: Operational Base + offset (0x0200)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	resetn_ddr_fail_safe When high, reset relative logic.
14	RW	0x0	resetn_pwm1 When high, reset relative logic.
13	RW	0x0	presetn_pwm1 When high, reset relative logic.
12	RW	0x0	resetn_pwm0 When high, reset relative logic.
11	RW	0x0	presetn_pwm0 When high, reset relative logic.
10	RW	0x0	sresetn_uart1 When high, reset relative logic.
9	RW	0x0	presetn_uart1 When high, reset relative logic.
8	RW	0x0	resetn_i2c2 When high, reset relative logic.
7	RW	0x0	presetn_i2c2 When high, reset relative logic.
6:5	RO	0x0	reserved
4	RW	0x0	resetn_i2c0 When high, reset relative logic.
3	RW	0x0	presetn_i2c0 When high, reset relative logic.
2	RW	0x0	presetn_pmusgrf_remap When high, reset relative logic.
1	RW	0x0	presetn_pmusgrf When high, reset relative logic.
0	RW	0x0	presetn_pdpmu_biu When high, reset relative logic.

CRU_PMU_SOFTRST_CON01

Address: Operational Base + offset (0x0204)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14	RW	0x0	presetn_pmucru When high, reset relative logic.
13:10	RO	0x0	reserved
9	RW	0x0	presetn_pmupvtm When high, reset relative logic.
8	RW	0x0	resetn_pmupvtm When high, reset relative logic.
7	RO	0x0	reserved
6	RW	0x0	presetn_chipverotp When high, reset relative logic.
5	RW	0x0	presetn_pmugrf When high, reset relative logic.

Bit	Attr	Reset Value	Description
4	RW	0x0	resetn_spi0 When high, reset relative logic.
3	RW	0x0	presetn_spi0 When high, reset relative logic.
2	RW	0x0	dbresetn_gpio0 When high, reset relative logic.
1	RW	0x0	presetn_gpio0 When high, reset relative logic.
0	RO	0x0	reserved

5.5 CRU Register Description

5.5.1 Registers Summary

Name	Offset	Size	Reset Value	Description
CRU_APLL_CON0	0x0000	W	0x000020FA	APLL control register 0
CRU_APLL_CON1	0x0004	W	0x00001083	APLL control register 1
CRU_APLL_CON2	0x0008	W	0x00000000	APLL control register 2
CRU_APLL_CON3	0x000C	W	0x00000007	APLL control register 3
CRU_APLL_CON4	0x0010	W	0x00007F00	APLL control register 4
CRU_APLL_CON5	0x0014	W	0x00000000	APLL control register 5
CRU_APLL_CON6	0x0018	W	0x00000000	APLL control register 6
CRU_DPLL_CON0	0x0020	W	0x00003215	DPLL control register 0
CRU_DPLL_CON1	0x0024	W	0x00001084	DPLL control register 1
CRU_DPLL_CON2	0x0028	W	0x00000000	DPLL control register 2
CRU_DPLL_CON3	0x002C	W	0x00000007	DPLL control register 3
CRU_DPLL_CON4	0x0030	W	0x00007F00	DPLL control register 4
CRU_DPLL_CON5	0x0034	W	0x00000000	DPLL control register 5
CRU_DPLL_CON6	0x0038	W	0x00000000	DPLL control register 6
CRU_CPLL_CON0	0x0040	W	0x000020FA	CPLL control register 0
CRU_CPLL_CON1	0x0044	W	0x00001083	CPLL control register 1
CRU_CPLL_CON2	0x0048	W	0x00000000	CPLL control register 2
CRU_CPLL_CON3	0x004C	W	0x00000007	CPLL control register 3
CRU_CPLL_CON4	0x0050	W	0x00007F00	CPLL control register 4
CRU_CPLL_CON5	0x0054	W	0x00000000	CPLL control register 5
CRU_CPLL_CON6	0x0058	W	0x00000000	CPLL control register 6
CRU_HPLL_CON0	0x0060	W	0x00002145	HPLL control register 0
CRU_HPLL_CON1	0x0064	W	0x00001083	HPLL control register 1
CRU_HPLL_CON2	0x0068	W	0x00000000	HPLL control register 2
CRU_HPLL_CON3	0x006C	W	0x00000007	HPLL control register 3
CRU_HPLL_CON4	0x0070	W	0x00007F00	HPLL control register 4
CRU_HPLL_CON5	0x0074	W	0x00000000	HPLL control register 5
CRU_HPLL_CON6	0x0078	W	0x00000000	HPLL control register 6
CRU_APLL_OFFSETCAL_STATUS	0x0080	W	0x00000000	APLL offset calibration status register
CRU_DPLL_OFFSETCAL_STATUS	0x0084	W	0x00000000	APLL offset calibration status register
CRU_CPLL_OFFSETCAL_STATUS	0x0088	W	0x00000000	APLL offset calibration status register
CRU_HPLL_OFFSETCAL_STATUS	0x008C	W	0x00000000	APLL offset calibration status register
CRU_MODE_CON01	0x0090	W	0x00000000	Mode control register 1

Name	Offset	Size	Reset Value	Description
CRU_CLKSEL_CON00	0x0100	W	0x00000580	Internal clock select and division register 0
CRU_CLKSEL_CON01	0x0104	W	0x00000013	Internal clock select and division register 1
CRU_CLKSEL_CON02	0x0108	W	0x00000583	Internal clock select and division register 2
CRU_CLKSEL_CON03	0x010C	W	0x0000050B	Internal clock select and division register 3
CRU_CLKSEL_CON04	0x0110	W	0x00000103	Internal clock select and division register 4
CRU_CLKSEL_CON05	0x0114	W	0x00000B0B	Internal clock select and division register 5
CRU_CLKSEL_CON06	0x0118	W	0x00000B0B	Internal clock select and division register 6
CRU_CLKSEL_CON07	0x011C	W	0x00000305	Internal clock select and division register 7
CRU_CLKSEL_CON08	0x0120	W	0x0000000B	Internal clock select and division register 8
CRU_CLKSEL_CON09	0x0124	W	0x00000000	Internal clock select and division register 9
CRU_CLKSEL_CON10	0x0128	W	0x00000800	Internal clock select and division register 10
CRU_CLKSEL_CON11	0x012C	W	0x00000000	Internal clock select and division register 11
CRU_CLKSEL_CON12	0x0130	W	0x00000800	Internal clock select and division register 12
CRU_CLKSEL_CON13	0x0134	W	0x00000000	Internal clock select and division register 13
CRU_CLKSEL_CON14	0x0138	W	0x00000800	Internal clock select and division register 14
CRU_CLKSEL_CON15	0x013C	W	0x00000000	Internal clock select and division register 15
CRU_CLKSEL_CON16	0x0140	W	0x00000800	Internal clock select and division register 16
CRU_CLKSEL_CON17	0x0144	W	0x00000000	Internal clock select and division register 17
CRU_CLKSEL_CON18	0x0148	W	0x00000800	Internal clock select and division register 18
CRU_CLKSEL_CON19	0x014C	W	0x00000000	Internal clock select and division register 19
CRU_CLKSEL_CON20	0x0150	W	0x00000001	Internal clock select and division register 20
CRU_CLKSEL_CON21	0x0154	W	0x00000000	Internal clock select and division register 21
CRU_CLKSEL_CON22	0x0158	W	0x00000000	Internal clock select and division register 22
CRU_CLKSEL_CON23	0x015C	W	0x00000000	Internal clock select and division register 23
CRU_CLKSEL_CON24	0x0160	W	0x00000000	Internal clock select and division register 24
CRU_CLKSEL_CON25	0x0164	W	0x00000305	Internal clock select and division register 25

Name	Offset	Size	Reset Value	Description
CRU_CLKSEL_CON26	0x0168	W	0x00000007	Internal clock select and division register 26
CRU_CLKSEL_CON27	0x016C	W	0x00000B0B	Internal clock select and division register 27
CRU_CLKSEL_CON28	0x0170	W	0x00000000	Internal clock select and division register 28
CRU_CLKSEL_CON29	0x0174	W	0x00000000	Internal clock select and division register 29
CRU_CLKSEL_CON30	0x0178	W	0x0000014F	Internal clock select and division register 30
CRU_CLKSEL_CON31	0x017C	W	0x00001305	Internal clock select and division register 31
CRU_CLKSEL_CON32	0x0180	W	0x00000000	Internal clock select and division register 32
CRU_CLKSEL_CON33	0x0184	W	0x0000070B	Internal clock select and division register 33
CRU_CLKSEL_CON34	0x0188	W	0x00000000	Internal clock select and division register 34
CRU_CLKSEL_CON35	0x018C	W	0x00000003	Internal clock select and division register 35
CRU_CLKSEL_CON36	0x0190	W	0x0000000B	Internal clock select and division register 36
CRU_CLKSEL_CON37	0x0194	W	0x00000000	Internal clock select and division register 37
CRU_CLKSEL_CON38	0x0198	W	0x00000000	Internal clock select and division register 38
CRU_CLKSEL_CON40	0x01A0	W	0x00000182	Internal clock select and division register 40
CRU_CLKSEL_CON41	0x01A4	W	0x00000101	Internal clock select and division register 41
CRU_CLKSEL_CON42	0x01A8	W	0x00008382	Internal clock select and division register 42
CRU_CLKSEL_CON43	0x01AC	W	0x00008183	Internal clock select and division register 43
CRU_CLKSEL_CON44	0x01B0	W	0x00000183	Internal clock select and division register 44
CRU_CLKSEL_CON45	0x01B4	W	0x00000103	Internal clock select and division register 45
CRU_CLKSEL_CON46	0x01B8	W	0x00000102	Internal clock select and division register 46
CRU_CLKSEL_CON47	0x01BC	W	0x00000007	Internal clock select and division register 47
CRU_CLKSEL_CON48	0x01C0	W	0x00000000	Internal clock select and division register 48
CRU_CLKSEL_CON49	0x01C4	W	0x00000101	Internal clock select and division register 49
CRU_CLKSEL_CON50	0x01C8	W	0x00000101	Internal clock select and division register 50
CRU_CLKSEL_CON51	0x01CC	W	0x00000742	Internal clock select and division register 51
CRU_CLKSEL_CON52	0x01D0	W	0x00000000	Internal clock select and division register 52

Name	Offset	Size	Reset Value	Description
CRU_CLKSEL_CON53	0x01D4	W	0x00000503	Internal clock select and division register 53
CRU_CLKSEL_CON54	0x01D8	W	0x00004203	Internal clock select and division register 54
CRU_CLKSEL_CON55	0x01DC	W	0x00000002	Internal clock select and division register 55
CRU_CLKSEL_CON56	0x01E0	W	0x00000002	Internal clock select and division register 56
CRU_CLKSEL_CON57	0x01E4	W	0x00000002	Internal clock select and division register 57
CRU_CLKSEL_CON58	0x01E8	W	0x00000005	Internal clock select and division register 58
CRU_CLKSEL_CON59	0x01EC	W	0x00000005	Internal clock select and division register 59
CRU_CLKSEL_CON61	0x01F4	W	0x00000709	Internal clock select and division register 61
CRU_CLKSEL_CON63	0x01FC	W	0x00000107	Internal clock select and division register 63
CRU_CLKSEL_CON64	0x0200	W	0x0000010B	Internal clock select and division register 64
CRU_CLKSEL_CON65	0x0204	W	0x00000011	Internal clock select and division register 65
CRU_CLKSEL_CON66	0x0208	W	0x00000701	Internal clock select and division register 66
CRU_CLKSEL_CON67	0x020C	W	0x00000011	Internal clock select and division register 67
CRU_CLKSEL_CON68	0x0210	W	0x00000501	Internal clock select and division register 68
CRU_CLKSEL_CON69	0x0214	W	0x00000101	Internal clock select and division register 69
CRU_CLKSEL_CON70	0x0218	W	0x00000000	Internal clock select and division register 70
CRU_CLKSEL_CON71	0x021C	W	0x00000000	Internal clock select and division register 71
CRU_CLKSEL_CON72	0x0220	W	0x0000000B	Internal clock select and division register 72
CRU_CLKSEL_CON73	0x0224	W	0x00000007	Internal clock select and division register 73
CRU_CLKSEL_CON74	0x0228	W	0x00000000	Internal clock select and division register 74
CRU_CLKSEL_CON75	0x022C	W	0x00000000	Internal clock select and division register 75
CRU_CLKSEL_CON76	0x0230	W	0x00000101	Internal clock select and division register 76
CRU_CLKSEL_CON77	0x0234	W	0x00000101	Internal clock select and division register 77
CRU_GATE_CON00	0x0280	W	0x00000000	Internal clock gate and division register 0
CRU_GATE_CON02	0x0288	W	0x00000000	Internal clock gate and division register 2
CRU_GATE_CON03	0x028C	W	0x00000000	Internal clock gate and division register 3

Name	Offset	Size	Reset Value	Description
CRU_GATE_CON04	0x0290	W	0x00000000	Internal clock gate and division register 4
CRU_GATE_CON05	0x0294	W	0x00000000	Internal clock gate and division register 5
CRU_GATE_CON06	0x0298	W	0x00000000	Internal clock gate and division register 6
CRU_GATE_CON07	0x029C	W	0x00000000	Internal clock gate and division register 7
CRU_GATE_CON08	0x02A0	W	0x00000000	Internal clock gate and division register 8
CRU_GATE_CON09	0x02A4	W	0x00000000	Internal clock gate and division register 9
CRU_GATE_CON10	0x02A8	W	0x00000000	Internal clock gate and division register 10
CRU_GATE_CON11	0x02AC	W	0x00000000	Internal clock gate and division register 11
CRU_GATE_CON12	0x02B0	W	0x00000000	Internal clock gate and division register 12
CRU_GATE_CON13	0x02B4	W	0x00000000	Internal clock gate and division register 13
CRU_GATE_CON14	0x02B8	W	0x00000000	Internal clock gate and division register 14
CRU_GATE_CON15	0x02BC	W	0x00000000	Internal clock gate and division register 15
CRU_GATE_CON16	0x02C0	W	0x00000000	Internal clock gate and division register 16
CRU_GATE_CON17	0x02C4	W	0x00000000	Internal clock gate and division register 17
CRU_GATE_CON18	0x02C8	W	0x00000000	Internal clock gate and division register 18
CRU_GATE_CON19	0x02CC	W	0x00000000	Internal clock gate and division register 19
CRU_GATE_CON20	0x02D0	W	0x00000000	Internal clock gate and division register 20
CRU_GATE_CON21	0x02D4	W	0x00000000	Internal clock gate and division register 21
CRU_GATE_CON22	0x02D8	W	0x00000000	Internal clock gate and division register 22
CRU_GATE_CON23	0x02DC	W	0x00000000	Internal clock gate and division register 23
CRU_GATE_CON24	0x02E0	W	0x00000000	Internal clock gate and division register 24
CRU_SOFTRST_CON00	0x0300	W	0x00000000	Internal clock reset register 0
CRU_SOFTRST_CON01	0x0304	W	0x00000000	Internal clock reset register 1
CRU_SOFTRST_CON02	0x0308	W	0x00000400	Internal clock reset register 2
CRU_SOFTRST_CON03	0x030C	W	0x00000000	Internal clock reset register 3
CRU_SOFTRST_CON04	0x0310	W	0x00000000	Internal clock reset register 4
CRU_SOFTRST_CON05	0x0314	W	0x00000000	Internal clock reset register 5
CRU_SOFTRST_CON06	0x0318	W	0x00000000	Internal clock reset register 6
CRU_SOFTRST_CON07	0x031C	W	0x00000000	Internal clock reset register 7
CRU_SOFTRST_CON08	0x0320	W	0x00000000	Internal clock reset register 8
CRU_SOFTRST_CON09	0x0324	W	0x00000000	Internal clock reset register 9
CRU_SOFTRST_CON10	0x0328	W	0x00000000	Internal clock reset register 10

Name	Offset	Size	Reset Value	Description
CRU_SOFTRST_CON11	0x032C	W	0x00000000	Internal clock reset register 11
CRU_SOFTRST_CON12	0x0330	W	0x00001000	Internal clock reset register 12
CRU_SOFTRST_CON13	0x0334	W	0x00000000	Internal clock reset register 13
CRU_SOFTRST_CON14	0x0338	W	0x00000000	Internal clock reset register 14
CRU_SSCGTBL_CON0	0x0380	W	0x00000000	External wave table configuration register 0
CRU_SSCGTBL_CON1	0x0384	W	0x00000000	External wave table configuration register 1
CRU_SSCGTBL_CON2	0x0388	W	0x00000000	External wave table configuration register 2
CRU_SSCGTBL_CON3	0x038C	W	0x00000000	External wave table configuration register 3
CRU_SSCGTBL_CON4	0x0390	W	0x00000000	External wave table configuration register 4
CRU_SSCGTBL_CON5	0x0394	W	0x00000000	External wave table configuration register 5
CRU_SSCGTBL_CON6	0x0398	W	0x00000000	External wave table configuration register 6
CRU_SSCGTBL_CON7	0x039C	W	0x00000000	External wave table configuration register 7
CRU_SSCGTBL_CON8	0x03A0	W	0x00000000	External wave table configuration register 8
CRU_SSCGTBL_CON9	0x03A4	W	0x00000000	External wave table configuration register 9
CRU_SSCGTBL_CON10	0x03A8	W	0x00000000	External wave table configuration register 10
CRU_SSCGTBL_CON11	0x03AC	W	0x00000000	External wave table configuration register 11
CRU_SSCGTBL_CON12	0x03B0	W	0x00000000	External wave table configuration register 12
CRU_SSCGTBL_CON13	0x03B4	W	0x00000000	External wave table configuration register 13
CRU_SSCGTBL_CON14	0x03B8	W	0x00000000	External wave table configuration register 14
CRU_SSCGTBL_CON15	0x03BC	W	0x00000000	External wave table configuration register 15
CRU_SSCGTBL_CON16	0x03C0	W	0x00000000	External wave table configuration register 16
CRU_SSCGTBL_CON17	0x03C4	W	0x00000000	External wave table configuration register 17
CRU_SSCGTBL_CON18	0x03C8	W	0x00000000	External wave table configuration register 18
CRU_SSCGTBL_CON19	0x03CC	W	0x00000000	External wave table configuration register 19
CRU_SSCGTBL_CON20	0x03D0	W	0x00000000	External wave table configuration register 20
CRU_SSCGTBL_CON21	0x03D4	W	0x00000000	External wave table configuration register 21
CRU_SSCGTBL_CON22	0x03D8	W	0x00000000	External wave table configuration register 22
CRU_SSCGTBL_CON23	0x03DC	W	0x00000000	External wave table configuration register 23

Name	Offset	Size	Reset Value	Description
CRU_SSCGTBL_CON24	0x03E0	W	0x00000000	External wave table configuration register 24
CRU_SSCGTBL_CON25	0x03E4	W	0x00000000	External wave table configuration register 25
CRU_SSCGTBL_CON26	0x03E8	W	0x00000000	External wave table configuration register 26
CRU_SSCGTBL_CON27	0x03EC	W	0x00000000	External wave table configuration register 27
CRU_SSCGTBL_CON28	0x03F0	W	0x00000000	External wave table configuration register 28
CRU_SSCGTBL_CON29	0x03F4	W	0x00000000	External wave table configuration register 29
CRU_SSCGTBL_CON30	0x03F8	W	0x00000000	External wave table configuration register 30
CRU_SSCGTBL_CON31	0x03FC	W	0x00000000	External wave table configuration register 31
CRU_GLB_CNT_TH	0x0400	W	0x00640064	Global counter threshold register
CRU_GLB_RST_ST	0x0404	W	0x00000000	Global reset status register
CRU_GLB_SRST_FST_VALUE	0x0408	W	0x00000000	Global software reset first value register
CRU_GLB_SRST_SND_VALUE	0x040C	W	0x00000000	Global software reset second value register
CRU_GLB_RST_CON	0x0410	W	0x00000000	Global reset control register
CRU_SDMMC_CON0	0x0440	W	0x00000004	SDMMC clock control register 0
CRU_SDMMC_CON1	0x0444	W	0x00000004	SDMMC clock control register 1
CRU_SDIO_CON0	0x0448	W	0x00000004	SDIO clock control register 0
CRU_SDIO_CON1	0x044C	W	0x00000004	SDIO clock control register 1
CRU_EMMC_CON0	0x0450	W	0x00000004	EMMC clock control register 0
CRU_EMMC_CON1	0x0454	W	0x00000004	EMMC clock control register 1
CRU_GMAC_CON	0x0460	W	0x00000000	GMAC clock control register
CRU_MISC_CON0	0x0464	W	0x00000000	Miscellaneous control register 0
CRU_MISC_CON1	0x0468	W	0x00000000	Miscellaneous control register 1

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

5.5.2 Detail Register Description

CRU_APLL_CON0

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	bypass PLL Bypass. FREF bypasses PLL to FOUTPOSTDIV. 1'b0: No bypass 1'b1: Bypass
14:12	RW	0x2	postdiv1 First post divide value. Valid settings are [1-7].
11:0	RW	0x0fa	fbdiv Feedback Divide Value, valid divider settings are: [16, 2500] in integer mode [20, 500] in fractional mode Tips: No plus one operation

CRU_APLL_CON1

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	pllpdsel PLL global power down source selection. If pllpdsel=1, PLL can be power down only by pllpd1, otherwise PLL is power down when any one of refdiv/fbdiv/fracdiv is changed or pllpd0 is asserted.
14	RW	0x0	pllpd1 PLL global power down request 1. 1'b0: No power down 1'b1: Power down
13	RW	0x0	pllpd0 PLL global power down request 0. 1'b0: No power down 1'b1: Power down
12	RW	0x1	dsmpd PLL delta sigma modulator enable. 1'b0: Modulator is enable 1'b1: Modulator is disabled
11	RO	0x0	reserved
10	RW	0x0	pll_lock PLL lock status. 1'b0: Unlock 1'b1: Lock
9	RO	0x0	reserved
8:6	RW	0x2	postdiv2 Second post divide value. Valid settings are [1-7].
5:0	RW	0x03	refdiv Reference clock divide value, Valid settings are [1-63].

CRU_APLL_CON2

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27	RW	0x0	fout4phasepd Power down 4-phase clocks. 1'b0: No power down 1'b1: Power down
26	RW	0x0	foutvcopd Power down buffered VCO clock. 1'b0: No power down 1'b1: Power down
25	RW	0x0	foutpostdivpd Power down all outputs except for buffered VCO clock. 1'b0: No power down 1'b1: Power down
24	RW	0x0	dacpd Power down quantization noise cancellation DAC. 1'b0: No power down 1'b1: Power down

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

CRU_APLL_CON3

Address: Operational Base + offset (0x000C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:13	RO	0x0	reserved
12:8	RW	0x00	ssmod_spread Spread amplitude. % = 0.1 * SPREAD[4:0]
7:4	RW	0x0	ssmod_divval Divider required to set the modulation frequency
3	RW	0x0	ssmod_downspread Select center spread or down spread. 1'b0: Down spread 1'b1: Center spread
2	RW	0x1	ssmod_reset Reset modulator state. 1'b0: No reset 1'b1: Reset
1	RW	0x1	ssmod_disable_sscg Bypass SSMOD by module. 1'b0: No bypass 1'b1: Bypass
0	RW	0x1	ssmod_bp Bypass SSMOD by integration. 1'b0: No bypass 1'b1: Bypass

CRU_APLL_CON4

Address: Operational Base + offset (0x0010)

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

CRU_APLL_CON5

Address: Operational Base + offset (0x0014)

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

Bit	Attr	Reset Value	Description
15:4	RW	0x000	offsetcal_in offsetcal_byp=1'b0: Initial condition for offset calibration logic. offsetcal_byp=1'b1: Override value for offset calibration. It is a signed integer with positive values delaying the reset of the faster path, and negative values delaying the reset of the slower path. 5'b0 is the minimum value, with each count increasing the reset time by one buffer delay. If offsetcal_en=1, this can be used to force a offset correction value based on a previous readout of offsetcal_out[11:0]. If offsetcal_byp=1 this value is forced directly into the calibration logic. If offsetcal_byp=0 this is the initial condition for the calibration sequence.
3	RW	0x0	offsetcal_fast Offset fast calibration enable. Set this to 1 for initial calibration if an initial value is not already known. Should be set to 0 for normal operation.
2	RW	0x0	offsetcal_en Offset calibration enable to actively adjust for input offset. 1'b0: Offset calibration is disabled. Static phase offset is determined by analog matching only. 1'b1: Offset calibration is enabled. Static phase offset is adjusted by sensing phase at the input.
1	RW	0x0	offsetcal_byp Offset calibration bypass. 1'b0: Use the offset calibration output(when offsetcal_en=1) to set the phase correction 1'b1: Use the offsetcal_in[11:0] value (when offsetcal_en=1) to set the phase correction
0	RW	0x0	fout2xpd Power down 2X clocks. 1'b0: No power down 1'b1: Power down

CRU_APLL_CON6

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:3	RO	0x0000	reserved
2:0	RW	0x0	offsetcal_cnt Programmable counter for offset calibration loop. Selects the number of PFD edges to wait after each offset calibration step. Count is defined as 2^(offsetcal_cnt+4).

CRU_DPLL_CON0

Address: Operational Base + offset (0x0020)

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

Bit	Attr	Reset Value	Description
15	RW	0x0	bypass PLL Bypass. FREF bypasses PLL to FOUTPOSTDIV. 1'b0: No bypass 1'b1: Bypass
14:12	RW	0x3	postdiv1 First post divide value. Valid settings are [1-7].
11:0	RW	0x215	fbdiv Feedback Divide Value, valid divider settings are: [16, 2500] in integer mode [20, 500] in fractional mode Tips: No plus one operation

CRU_DPLL_CON1

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	pllpdsel PLL global power down source selection. If pllpdsel=1, PLL can be power down only by pllpd1, otherwise PLL is power down when any one of refdiv/fbdiv/fracdiv is changed or pllpd0 is asserted.
14	RW	0x0	pllpd1 PLL global power down request 1. 1'b0: No power down 1'b1: Power down
13	RW	0x0	pllpd0 PLL global power down request 0. 1'b0: No power down 1'b1: Power down
12	RW	0x1	dsmpd PLL delta sigma modulator enable. 1'b0: Modulator is enable 1'b1: Modulator is disabled
11	RO	0x0	reserved
10	RW	0x0	pll_lock PLL lock status. 1'b0: Unlock 1'b1: Lock
9	RO	0x0	reserved
8:6	RW	0x2	postdiv2 Second post divide value. Valid settings are [1-7].
5:0	RW	0x04	refdiv Reference clock divide value, Valid settings are [1-63].

CRU_DPLL_CON2

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27	RW	0x0	fout4phasepd Power down 4-phase clocks. 1'b0: No power down 1'b1: Power down

Bit	Attr	Reset Value	Description
26	RW	0x0	foutvcopd Power down buffered VCO clock. 1'b0: No power down 1'b1: Power down
25	RW	0x0	foutpostdivpd Power down all outputs except for buffered VCO clock. 1'b0: No power down 1'b1: Power down
24	RW	0x0	dacpd Power down quantization noise cancellation DAC. 1'b0: No power down 1'b1: Power down
23:0	RW	0x000000	fracdiv Fractional part of feedback divide, fraction = fracdiv/2^24.

CRU_DPLL_CON3

Address: Operational Base + offset (0x002C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:13	RO	0x0	reserved
12:8	RW	0x00	ssmod_spread Spread amplitude. % = 0.1 * SPREAD[4:0]
7:4	RW	0x0	ssmod_divval Divider required to set the modulation frequency
3	RW	0x0	ssmod_downspread Select center spread or down spread. 1'b0: Down spread 1'b1: Center spread
2	RW	0x1	ssmod_reset Reset modulator state. 1'b0: No reset 1'b1: Reset
1	RW	0x1	ssmod_disable_sscg Bypass SSMOD by module. 1'b0: No bypass 1'b1: Bypass
0	RW	0x1	ssmod_bp Bypass SSMOD by integration. 1'b0: No bypass 1'b1: Bypass

CRU_DPLL_CON4

Address: Operational Base + offset (0x0030)

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

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

CRU_DPLL_CON5

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:4	RW	0x000	offsetcal_in offsetcal_byp=1'b0: Initial condition for offset calibration logic. offsetcal_byp=1'b1: Override value for offset calibration. It is a signed integer with positive values delaying the reset of the faster path, and negative values delaying the reset of the slower path. 5'b0 is the minimum value, with each count increasing the reset time by one buffer delay. If offsetcal_en=1, this can be used to force a offset correction value based on a previous readout of offsetcal_out[11:0]. If offsetcal_byp=1 this value is forced directly into the calibration logic. If offsetcal_byp=0 this is the initial condition for the calibration sequence.
3	RW	0x0	offsetcal_fast Offset fast calibration enable. Set this to 1 for initial calibration if an initial value is not already known. Should be set to 0 for normal operation.
2	RW	0x0	offsetcal_en Offset calibration enable to actively adjust for input offset. 1'b0: Offset calibration is disabled. Static phase offset is determined by analog matching only. 1'b1: Offset calibration is enabled. Static phase offset is adjusted by sensing phase at the input.
1	RW	0x0	offsetcal_byp Offset calibration bypass. 1'b0: Use the offset calibration output(when offsetcal_en=1) to set the phase correction 1'b1: Use the offsetcal_in[11:0] value (when offsetcal_en=1) to set the phase correction
0	RW	0x0	fout2xpd Power down 2X clocks. 1'b0: No power down 1'b1: Power down

CRU_DPLL_CON6

Address: Operational Base + offset (0x0038)

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

Bit	Attr	Reset Value	Description
2:0	RW	0x0	offsetcal_cnt Programmable counter for offset calibration loop. Selects the number of PFD edges to wait after each offset calibration step. Count is defined as 2^(offsetcal_cnt+4).

CRU_CPLL_CON0

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	bypass PLL Bypass. FREF bypasses PLL to FOUTPOSTDIV. 1'b0: No bypass 1'b1: Bypass
14:12	RW	0x2	postdiv1 First post divide value. Valid settings are [1-7].
11:0	RW	0x0fa	fbdiv Feedback Divide Value, valid divider settings are: [16, 2500] in integer mode [20, 500] in fractional mode Tips: No plus one operation

CRU_CPLL_CON1

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	pllpdsel PLL global power down source selection. If pllpdsel=1, PLL can be power down only by pllpd1, otherwise PLL is power down when any one of refdiv/fbdiv/fracdiv is changed or pllpd0 is asserted.
14	RW	0x0	pllpd1 PLL global power down request 1. 1'b0: No power down 1'b1: Power down
13	RW	0x0	pllpd0 PLL global power down request 0. 1'b0: No power down 1'b1: Power down
12	RW	0x1	dsmpd PLL delta sigma modulator enable. 1'b0: Modulator is enable 1'b1: Modulator is disabled
11	RO	0x0	reserved
10	RW	0x0	pll_lock PLL lock status. 1'b0: Unlock 1'b1: Lock
9	RO	0x0	reserved

Bit	Attr	Reset Value	Description
8:6	RW	0x2	postdiv2 Second post divide value. Valid settings are [1-7].
5:0	RW	0x03	refdiv Reference clock divide value, Valid settings are [1-63].

CRU_CPLL_CON2

Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27	RW	0x0	fout4phasepd Power down 4-phase clocks. 1'b0: No power down 1'b1: Power down
26	RW	0x0	foutvcopd Power down buffered VCO clock. 1'b0: No power down 1'b1: Power down
25	RW	0x0	foutpostdivpd Power down all outputs except for buffered VCO clock. 1'b0: No power down 1'b1: Power down
24	RW	0x0	dacpd Power down quantization noise cancellation DAC. 1'b0: No power down 1'b1: Power down
23:0	RW	0x000000	fracdiv Fractional part of feedback divide, fraction = fracdiv/2^24.

CRU_CPLL_CON3

Address: Operational Base + offset (0x004C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:13	RO	0x0	reserved
12:8	RW	0x00	ssmod_spread Spread amplitude. % = 0.1 * SPREAD[4:0]
7:4	RW	0x0	ssmod_divval Divider required to set the modulation frequency
3	RW	0x0	ssmod_downspread Select center spread or down spread. 1'b0: Down spread 1'b1: Center spread
2	RW	0x1	ssmod_reset Reset modulator state. 1'b0: No reset 1'b1: Reset
1	RW	0x1	ssmod_disable_sscg Bypass SSMOD by module. 1'b0: No bypass 1'b1: Bypass

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

CRU_CPLL_CON4

Address: Operational Base + offset (0x0050)

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

CRU_CPLL_CON5

Address: Operational Base + offset (0x0054)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:4	RW	0x000	offsetcal_in offsetcal_byp=1'b0: Initial condition for offset calibration logic. offsetcal_byp=1'b1: Override value for offset calibration. It is a signed integer with positive values delaying the reset of the faster path, and negative values delaying the reset of the slower path. 5'b0 is the minimum value, with each count increasing the reset time by one buffer delay. If offsetcal_en=1, this can be used to force a offset correction value based on a previous readout of offsetcal_out[11:0]. If offsetcal_byp=1 this value is forced directly into the calibration logic. If offsetcal_byp=0 this is the initial condition for the calibration sequence.
3	RW	0x0	offsetcal_fast Offset fast calibration enable. Set this to 1 for initial calibration if an initial value is not already known. Should be set to 0 for normal operation.
2	RW	0x0	offsetcal_en Offset calibration enable to actively adjust for input offset. 1'b0: Offset calibration is disabled. Static phase offset is determined by analog matching only. 1'b1: Offset calibration is enabled. Static phase offset is adjusted by sensing phase at the input.

Bit	Attr	Reset Value	Description
1	RW	0x0	offsetcal_byp Offset calibration bypass. 1'b0: Use the offset calibration output(when offsetcal_en=1) to set the phase correction 1'b1: Use the offsetcal_in[11:0] value (when offsetcal_en=1) to set the phase correction
0	RW	0x0	fout2xpd Power down 2X clocks. 1'b0: No power down 1'b1: Power down

CRU_CPLL_CON6

Address: Operational Base + offset (0x0058)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:3	RO	0x0000	reserved
2:0	RW	0x0	offsetcal_cnt Programmable counter for offset calibration loop. Selects the number of PFD edges to wait after each offset calibration step. Count is defined as 2^(offsetcal_cnt+4).

CRU_HPLL_CON0

Address: Operational Base + offset (0x0060)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	bypass PLL Bypass. FREF bypasses PLL to FOUTPOSTDIV. 1'b0: No bypass 1'b1: Bypass
14:12	RW	0x2	postdiv1 First post divide value. Valid settings are [1-7].
11:0	RW	0x145	fbdiv Feedback Divide Value, valid divider settings are: [16, 2500] in integer mode [20, 500] in fractional mode Tips: No plus one operation

CRU_HPLL_CON1

Address: Operational Base + offset (0x0064)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	pllpdsel PLL global power down source selection. If pllpdsel=1, PLL can be power down only by pllpd1, otherwise PLL is power down when any one of refdiv/fbdiv/fracdiv is changed or pllpd0 is asserted.

Bit	Attr	Reset Value	Description
14	RW	0x0	pllpd1 PLL global power down request 1. 1'b0: No power down 1'b1: Power down
13	RW	0x0	pllpd0 PLL global power down request 0. 1'b0: No power down 1'b1: Power down
12	RW	0x1	dsmpd PLL delta sigma modulator enable. 1'b0: Modulator is enable 1'b1: Modulator is disabled
11	RO	0x0	reserved
10	RW	0x0	pll_lock PLL lock status. 1'b0: Unlock 1'b1: Lock
9	RO	0x0	reserved
8:6	RW	0x2	postdiv2 Second post divide value. Valid settings are [1-7].
5:0	RW	0x03	refdiv Reference clock divide value, Valid settings are [1-63].

CRU_HPLL_CON2

Address: Operational Base + offset (0x0068)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27	RW	0x0	fout4phasepd Power down 4-phase clocks. 1'b0: No power down 1'b1: Power down
26	RW	0x0	foutvcopd Power down buffered VCO clock. 1'b0: No power down 1'b1: Power down
25	RW	0x0	foutpostdivpd Power down all outputs except for buffered VCO clock. 1'b0: No power down 1'b1: Power down
24	RW	0x0	dacpd Power down quantization noise cancellation DAC. 1'b0: No power down 1'b1: Power down
23:0	RW	0x000000	fracdiv Fractional part of feedback divide, fraction = fracdiv/2^24.

CRU_HPLL_CON3

Address: Operational Base + offset (0x006C)

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

Bit	Attr	Reset Value	Description
12:8	RW	0x00	ssmod_spread Spread amplitude. % = 0.1 * SPREAD[4:0]
7:4	RW	0x0	ssmod_divval Divider required to set the modulation frequency
3	RW	0x0	ssmod_downspread Select center spread or down spread. 1'b0: Down spread 1'b1: Center spread
2	RW	0x1	ssmod_reset Reset modulator state. 1'b0: No reset 1'b1: Reset
1	RW	0x1	ssmod_disable_sscg Bypass SSMOD by module. 1'b0: No bypass 1'b1: Bypass
0	RW	0x1	ssmod_bp Bypass SSMOD by integration. 1'b0: No bypass 1'b1: Bypass

CRU_HPLL_CON4

Address: Operational Base + offset (0x0070)

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

CRU_HPLL_CON5

Address: Operational Base + offset (0x0074)

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

Bit	Attr	Reset Value	Description
15:4	RW	0x000	offsetcal_in offsetcal_byp=1'b0: Initial condition for offset calibration logic. offsetcal_byp=1'b1: Override value for offset calibration. It is a signed integer with positive values delaying the reset of the faster path, and negative values delaying the reset of the slower path. 5'b0 is the minimum value, with each count increasing the reset time by one buffer delay. If offsetcal_en=1, this can be used to force a offset correction value based on a previous readout of offsetcal_out[11:0]. If offsetcal_byp=1 this value is forced directly into the calibration logic. If offsetcal_byp=0 this is the initial condition for the calibration sequence.
3	RW	0x0	offsetcal_fast Offset fast calibration enable. Set this to 1 for initial calibration if an initial value is not already known. Should be set to 0 for normal operation.
2	RW	0x0	offsetcal_en Offset calibration enable to actively adjust for input offset. 1'b0: Offset calibration is disabled. Static phase offset is determined by analog matching only. 1'b1: Offset calibration is enabled. Static phase offset is adjusted by sensing phase at the input.
1	RW	0x0	offsetcal_byp Offset calibration bypass. 1'b0: Use the offset calibration output(when offsetcal_en=1) to set the phase correction 1'b1: Use the offsetcal_in[11:0] value (when offsetcal_en=1) to set the phase correction
0	RW	0x0	fout2xpd Power down 2X clocks. 1'b0: No power down 1'b1: Power down

CRU_HPLL_CON6

Address: Operational Base + offset (0x0078)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:3	RO	0x0000	reserved
2:0	RW	0x0	offsetcal_cnt Programmable counter for offset calibration loop. Selects the number of PFD edges to wait after each offset calibration step. Count is defined as 2^(offsetcal_cnt+4).

CRU_APLL_OFFSETCAL_STATUS

Address: Operational Base + offset (0x0080)

Bit	Attr	Reset Value	Description
31:17	RO	0x0000	reserved
16	RW	0x0	offsetcal_lock Asserted when phase lock is achieved when offsetcal_en is set to 1'b1
15:12	RO	0x0	reserved

Bit	Attr	Reset Value	Description
11:0	RW	0x000	offsetcal_out This is the output of either the offset calibration block (if offsetcal_byp=0) or a buffered version of offsetcal_in[11:0] (if offsetcal_byp=1). It can be used to read out the phase calibration state to use as an override value so that offset calibration can be bypassed for faster locking. The value changes on the rising edge of FREF, so it can be clocked out on the falling edge of FREF

CRU_DPLL_OFFSETCAL_STATUS

Address: Operational Base + offset (0x0084)

Bit	Attr	Reset Value	Description
31:17	RO	0x0000	reserved
16	RW	0x0	offsetcal_lock Asserted when phase lock is achieved when offsetcal_en is set to 1'b1
15:12	RO	0x0	reserved
11:0	RW	0x000	offsetcal_out This is the output of either the offset calibration block (if offsetcal_byp=0) or a buffered version of offsetcal_in[11:0] (if offsetcal_byp=1). It can be used to read out the phase calibration state to use as an override value so that offset calibration can be bypassed for faster locking. The value changes on the rising edge of FREF, so it can be clocked out on the falling edge of FREF

CRU_CPLL_OFFSETCAL_STATUS

Address: Operational Base + offset (0x0088)

Bit	Attr	Reset Value	Description
31:17	RO	0x0000	reserved
16	RW	0x0	offsetcal_lock Asserted when phase lock is achieved when offsetcal_en is set to 1'b1
15:12	RO	0x0	reserved
11:0	RW	0x000	offsetcal_out This is the output of either the offset calibration block (if offsetcal_byp=0) or a buffered version of offsetcal_in[11:0] (if offsetcal_byp=1). It can be used to read out the phase calibration state to use as an override value so that offset calibration can be bypassed for faster locking. The value changes on the rising edge of FREF, so it can be clocked out on the falling edge of FREF

CRU_HPLL_OFFSETCAL_STATUS

Address: Operational Base + offset (0x008C)

Bit	Attr	Reset Value	Description
31:17	RO	0x0000	reserved
16	RW	0x0	offsetcal_lock Asserted when phase lock is achieved when offsetcal_en is set to 1'b1
15:12	RO	0x0	reserved

Bit	Attr	Reset Value	Description
11:0	RW	0x000	offsetcal_out This is the output of either the offset calibration block (if offsetcal_byp=0) or a buffered version of offsetcal_in[11:0] (if offsetcal_byp=1). It can be used to read out the phase calibration state to use as an override value so that offset calibration can be bypassed for faster locking. The value changes on the rising edge of FREF, so it can be clocked out on the falling edge of FREF

CRU_MODE_CON01

Address: Operational Base + offset (0x0090)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RO	0x0	reserved
11:10	RW	0x0	usbphy_clk480m_muxpll_mode usbphy_clk480m_mux clock mux. 2'b00: xin_osc0_div 2'b01: usbphy_clk480m 2'b10: clk_deepslow Others: Reserved
9:8	RO	0x0	reserved
7:6	RW	0x0	clk_hpll_mode clk_hpll_mux clock mux. 2'b00: xin_osc0_func 2'b01: clk_hpll 2'b10: clk_deepslow Others: Reserved
5:4	RW	0x0	clk_cpll_mode clk_cpll_mux clock mux. 2'b00: xin_osc0_func 2'b01: clk_cpll 2'b10: clk_deepslow Others: Reserved
3:2	RW	0x0	clk_dpll_mode clk_dpll_mux clock mux. 2'b00: xin_osc0_func 2'b01: clk_dpll 2'b10: clk_deepslow Others: Reserved
1:0	RW	0x0	clk_apll_mode clk_apll_mux clock mux. 2'b00: xin_osc0_func 2'b01: clk_apll 2'b10: clk_deepslow Others: Reserved

CRU_CLKSEL_CON00

Address: Operational Base + offset (0x0100)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:13	RO	0x0	reserved
12:8	RW	0x05	hclk_pdcore_biu_div Divide hclk_pdcore_biu by (div_con + 1)
7:6	RW	0x2	clk_core_pre_sel clk_core_pre clock mux. 2'b00: clk_gpll_mux 2'b01: clk_cpll_mux 2'b10: clk_apll_mux Others: Reserved
5	RO	0x0	reserved
4:0	RW	0x00	clk_core_pre_div Divide clk_core_pre by (div_con + 1)

CRU_CLKSEL_CON01

Address: Operational Base + offset (0x0104)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7:4	RW	0x1	aclk_core_pre_div Divide aclk_core_pre by (div_con + 1)
3	RO	0x0	reserved
2:0	RW	0x3	pclk_dbg_pre_div Divide pclk_dbg_pre by (div_con + 1)

CRU_CLKSEL_CON02

Address: Operational Base + offset (0x0108)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	hclk_pdbus_pre_sel hclk_pdbus_pre clock mux. 1'b0: clk_gpll_mux 1'b1: clk_cpll_mux
14:13	RO	0x0	reserved
12:8	RW	0x05	hclk_pdbus_pre_div Divide hclk_pdbus_pre by (div_con + 1)
7:6	RW	0x2	aclk_pdbus_pre_sel aclk_pdbus_pre clock mux. 2'b00: clk_gpll_mux 2'b01: clk_cpll_mux 2'b10: clk_dpll_mux Others: Reserved
5	RO	0x0	reserved
4:0	RW	0x03	aclk_pdbus_pre_div Divide aclk_pdbus_pre by (div_con + 1)

CRU_CLKSEL_CON03

Address: Operational Base + offset (0x010C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	clk_mcu_sel clk_mcu clock mux. 1'b0: clk_gpll_mux 1'b1: clk_cpll_mux
14:13	RO	0x0	reserved
12:8	RW	0x05	clk_mcu_div Divide clk_mcu by (div_con + 1)
7	RW	0x0	pclk_pdbus_pre_sel pclk_pdbus_pre clock mux. 1'b0: clk_gpll_mux 1'b1: clk_cpll_mux
6:5	RO	0x0	reserved
4:0	RW	0x0b	pclk_pdbus_pre_div Divide pclk_pdbus_pre by (div_con + 1)

CRU_CLKSEL_CON04

Address: Operational Base + offset (0x0110)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:13	RO	0x0	reserved
12:8	RW	0x01	hclk_pdcrypt_div Divide hclk_pdcrypt by (div_con + 1)
7	RW	0x0	aclk_pdcrypt_sel aclk_pdcrypt clock mux. 1'b0: clk_gpll_mux 1'b1: clk_cpll_mux
6:5	RO	0x0	reserved
4:0	RW	0x03	aclk_pdcrypt_div Divide aclk_pdcrypt by (div_con + 1)

CRU_CLKSEL_CON05

Address: Operational Base + offset (0x0114)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14:8	RW	0x0b	clk_i2c3_div Divide clk_i2c3 by (div_con + 1)
7	RO	0x0	reserved
6:0	RW	0x0b	clk_i2c1_div Divide clk_i2c1 by (div_con + 1)

CRU_CLKSEL_CON06

Address: Operational Base + offset (0x0118)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14:8	RW	0x0b	clk_i2c5_div Divide clk_i2c5 by (div_con + 1)
7	RO	0x0	reserved
6:0	RW	0x0b	clk_i2c4_div Divide clk_i2c4 by (div_con + 1)

CRU_CLKSEL_CON07

Address: Operational Base + offset (0x011C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	clk_crypto_pka_sel clk_crypto_pka clock mux. 1'b0: clk_gpll_mux 1'b1: clk_cpll_mux
14:13	RO	0x0	reserved
12:8	RW	0x03	clk_crypto_pka_div Divide clk_crypto_pka by (div_con + 1)
7	RW	0x0	clk_crypto_core_sel clk_crypto_core clock mux. 1'b0: clk_gpll_mux 1'b1: clk_cpll_mux
6:5	RO	0x0	reserved
4:0	RW	0x05	clk_crypto_core_div Divide clk_crypto_core by (div_con + 1)

CRU_CLKSEL_CON08

Address: Operational Base + offset (0x0120)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x0	clk_spi1_sel clk_spi1 clock mux. 1'b0: clk_gpll_mux 1'b1: xin_osc0_func
6:0	RW	0x0b	clk_spi1_div Divide clk_spi1 by (div_con + 1)

CRU_CLKSEL_CON09

Address: Operational Base + offset (0x0124)

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

Bit	Attr	Reset Value	Description
15	RW	0x0	clk_pwm2_sel clk_pwm2 clock mux. 1'b0: xin_osc0_func 1'b1: clk_gpll_mux
14:8	RW	0x00	clk_pwm2_div Divide clk_pwm2 by (div_con + 1)
7:0	RO	0x00	reserved

CRU_CLKSEL_CON10

Address: Operational Base + offset (0x0128)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RO	0x0	reserved
11:10	RW	0x2	sclk_uart0_sel sclk_uart0 clock mux. 2'b00: sclk_uart0_div 2'b01: sclk_uart0_frac 2'b10: xin_osc0_func Others: Reserved
9:8	RW	0x0	sclk_uart0_div_sel sclk_uart0_div clock mux. 2'b00: clk_gpll_mux 2'b01: clk_cpll_mux 2'b10: usbphy_clk480m_mux 2'b11: xin_osc0_func
7	RO	0x0	reserved
6:0	RW	0x00	sclk_uart0_div_div Divide sclk_uart0_div by (div_con + 1)

CRU_CLKSEL_CON11

Address: Operational Base + offset (0x012C)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sclk_uart0_frac_div sclk_uart0_frac fraction division register. High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL_CON12

Address: Operational Base + offset (0x0130)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RO	0x0	reserved
11:10	RW	0x2	sclk_uart2_sel sclk_uart2 clock mux. 2'b00: sclk_uart2_div 2'b01: sclk_uart2_frac 2'b10: xin_osc0_func Others: Reserved

Bit	Attr	Reset Value	Description
9:8	RW	0x0	sclk_uart2_div_sel sclk_uart2_div clock mux. 2'b00: clk_gpll_mux 2'b01: clk_cpll_mux 2'b10: usbphy_clk480m_mux 2'b11: xin_osc0_func
7	RO	0x0	reserved
6:0	RW	0x00	sclk_uart2_div_div Divide sclk_uart2_div by (div_con + 1)

CRU_CLKSEL_CON13

Address: Operational Base + offset (0x0134)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sclk_uart2_frac_div sclk_uart2_frac fraction division register. High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL_CON14

Address: Operational Base + offset (0x0138)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RO	0x0	reserved
11:10	RW	0x2	sclk_uart3_sel sclk_uart3 clock mux. 2'b00: sclk_uart3_div 2'b01: sclk_uart3_frac 2'b10: xin_osc0_func Others: Reserved
9:8	RW	0x0	sclk_uart3_div_sel sclk_uart3_div clock mux. 2'b00: clk_gpll_mux 2'b01: clk_cpll_mux 2'b10: usbphy_clk480m_mux 2'b11: xin_osc0_func
7	RO	0x0	reserved
6:0	RW	0x00	sclk_uart3_div_div Divide sclk_uart3_div by (div_con + 1)

CRU_CLKSEL_CON15

Address: Operational Base + offset (0x013C)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sclk_uart3_frac_div sclk_uart3_frac fraction division register. High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL_CON16

Address: Operational Base + offset (0x0140)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RO	0x0	reserved
11:10	RW	0x2	sclk_uart4_sel sclk_uart4 clock mux. 2'b00: sclk_uart4_div 2'b01: sclk_uart4_frac 2'b10: xin_osc0_func Others: Reserved
9:8	RW	0x0	sclk_uart4_div_sel sclk_uart4_div clock mux. 2'b00: clk_gpll_mux 2'b01: clk_cpll_mux 2'b10: usbphy_clk480m_mux 2'b11: xin_osc0_func
7	RO	0x0	reserved
6:0	RW	0x00	sclk_uart4_div_div Divide sclk_uart4_div by (div_con + 1)

CRU_CLKSEL_CON17

Address: Operational Base + offset (0x0144)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sclk_uart4_frac_div sclk_uart4_frac fraction division register. High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL_CON18

Address: Operational Base + offset (0x0148)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RO	0x0	reserved
11:10	RW	0x2	sclk_uart5_sel sclk_uart5 clock mux. 2'b00: sclk_uart5_div 2'b01: sclk_uart5_frac 2'b10: xin_osc0_func Others: Reserved
9:8	RW	0x0	sclk_uart5_div_sel sclk_uart5_div clock mux. 2'b00: clk_gpll_mux 2'b01: clk_cpll_mux 2'b10: usbphy_clk480m_mux 2'b11: xin_osc0_func
7	RO	0x0	reserved
6:0	RW	0x00	sclk_uart5_div_div Divide sclk_uart5_div by (div_con + 1)

CRU_CLKSEL_CON19

Address: Operational Base + offset (0x014C)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sclk_uart5_frac_div sclk_uart5_frac fraction division register. 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	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:11	RO	0x00	reserved
10:0	RW	0x001	clk_saradc_ndft_div Divide clk_saradc_ndft by (div_con + 1)

CRU_CLKSEL_CON21

Address: Operational Base + offset (0x0154)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	dbclk_gpio1_sel dbclk_gpio1 clock mux. 1'b0: xin_osc0_func 1'b1: clk_deepslow
14:0	RO	0x0000	reserved

CRU_CLKSEL_CON22

Address: Operational Base + offset (0x0158)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	dbclk_gpio2_sel dbclk_gpio2 clock mux. 1'b0: xin_osc0_func 1'b1: clk_deepslow
14:0	RO	0x0000	reserved

CRU_CLKSEL_CON23

Address: Operational Base + offset (0x015C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	dbclk_gpio3_sel dbclk_gpio3 clock mux. 1'b0: xin_osc0_func 1'b1: clk_deepslow
14:0	RO	0x0000	reserved

CRU_CLKSEL_CON24

Address: Operational Base + offset (0x0160)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	dbclk_gpio4_sel dbclk_gpio4 clock mux. 1'b0: xin_osc0_func 1'b1: clk_deepslow
14:0	RO	0x0000	reserved

CRU_CLKSEL_CON25

Address: Operational Base + offset (0x0164)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	dclk_decom_sel dclk_decom clock mux. 1'b0: clk_gpll_mux 1'b1: clk_cpll_mux
14:8	RW	0x03	dclk_decom_div Divide dclk_decom by (div_con + 1)
7	RW	0x0	clk_can_sel clk_can clock mux. 1'b0: clk_gpll_mux 1'b1: xin_osc0_func
6:0	RW	0x05	clk_can_div Divide clk_can by (div_con + 1)

CRU_CLKSEL_CON26

Address: Operational Base + offset (0x0168)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:5	RO	0x000	reserved
4:0	RW	0x07	hclk_pdaudio_div Divide hclk_pdaudio by (div_con + 1)

CRU_CLKSEL_CON27

Address: Operational Base + offset (0x016C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	mclk_i2s0_rx_div_sel mclk_i2s0_rx_div clock mux. 1'b0: clk_cpll_mux 1'b1: clk_gpll_mux

Bit	Attr	Reset Value	Description
14:8	RW	0x0b	mclk_i2s0_rx_div_div Divide mclk_i2s0_rx_div by (div_con + 1)
7	RW	0x0	mclk_i2s0_tx_div_sel mclk_i2s0_tx_div clock mux. 1'b0: clk_cpll_mux 1'b1: clk_gpll_mux
6:0	RW	0x0b	mclk_i2s0_tx_div_div Divide mclk_i2s0_tx_div by (div_con + 1)

CRU_CLKSEL_CON28

Address: Operational Base + offset (0x0170)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	mclk_i2s0_tx_fracdiv_div mclk_i2s0_tx_fracdiv fraction division register. High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL_CON29

Address: Operational Base + offset (0x0174)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	mclk_i2s0_rx_fracdiv_div mclk_i2s0_rx_fracdiv fraction division register. 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	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:9	RO	0x00	reserved
8	RW	0x1	mclk_i2s0_rx_out2io_sel mclk_i2s0_rx_out2io clock mux. 1'b0: mclk_i2s0_rx 1'b1: xin_osc0_half
7	RO	0x0	reserved
6	RW	0x1	mclk_i2s0_tx_out2io_sel mclk_i2s0_tx_out2io clock mux. 1'b0: mclk_i2s0_tx 1'b1: xin_osc0_half
5:4	RO	0x0	reserved
3:2	RW	0x3	mclk_i2s0_rx_sel mclk_i2s0_rx clock mux. 2'b00: mclk_i2s0_rx_div 2'b01: mclk_i2s0_rx_fracdiv 2'b10: i2s0_mclkin 2'b11: xin_osc0_half
1:0	RW	0x3	mclk_i2s0_tx_sel mclk_i2s0_tx clock mux. 2'b00: mclk_i2s0_tx_div 2'b01: mclk_i2s0_tx_fracdiv 2'b10: i2s0_mclkin 2'b11: xin_osc0_half

CRU_CLKSEL_CON31

Address: Operational Base + offset (0x017C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:13	RO	0x0	reserved
12	RW	0x1	mclk_i2s1_out2io_sel mclk_i2s1_out2io clock mux. 1'b0: mclk_i2s1 1'b1: xin_osc0_half
11	RO	0x0	reserved
10:8	RW	0x3	mclk_i2s1_sel mclk_i2s1 clock mux. 3'b000: mclk_i2s1_div 3'b001: mclk_i2s1_fracdiv 3'b010: i2s1_mclkin 3'b011: xin_osc0_half 3'b100: mclk_i2s1_fracdiv_v2 Others: Reserved
7	RW	0x0	mclk_i2s1_div_sel mclk_i2s1_div clock mux. 1'b0: clk_cpll_mux 1'b1: clk_gpll_mux
6:0	RW	0x05	mclk_i2s1_div_div Divide mclk_i2s1_div by (div_con + 1)

CRU_CLKSEL_CON32

Address: Operational Base + offset (0x0180)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	mclk_i2s1_fracdiv_div mclk_i2s1_fracdiv fraction division register. High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL_CON33

Address: Operational Base + offset (0x0184)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:11	RO	0x00	reserved
10	RW	0x1	mclk_i2s2_out2io_sel mclk_i2s2_out2io clock mux. 1'b0: mclk_i2s2 1'b1: xin_osc0_half
9:8	RW	0x3	mclk_i2s2_sel mclk_i2s2 clock mux. 2'b00: mclk_i2s2_div 2'b01: mclk_i2s2_fracdiv 2'b10: i2s2_mclkin 2'b11: xin_osc0_half

Bit	Attr	Reset Value	Description
7	RW	0x0	mclk_i2s2_div_sel mclk_i2s2_div clock mux. 1'b0: clk_cpll_mux 1'b1: clk_gpll_mux
6:0	RW	0x0b	mclk_i2s2_div_div Divide mclk_i2s2_div by (div_con + 1)

CRU_CLKSEL_CON34

Address: Operational Base + offset (0x0188)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	mclk_i2s2ch1_fracdiv_div mclk_i2s2_fracdiv fraction division register. High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL_CON35

Address: Operational Base + offset (0x018C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:10	RO	0x00	reserved
9:8	RW	0x0	mclk_pdm_sel mclk_pdm clock mux. 2'b00: clk_gpll_mux 2'b01: clk_cpll_mux 2'b10: xin_osc0_func Others: Reserved
7	RO	0x0	reserved
6:0	RW	0x03	mclk_pdm_div Divide mclk_pdm by (div_con + 1)

CRU_CLKSEL_CON36

Address: Operational Base + offset (0x0190)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:10	RO	0x00	reserved
9:8	RW	0x0	sclk_audpwm_sel sclk_audpwm clock mux. 2'b00: sclk_audpwm_div 2'b01: sclk_audpwm_fracdiv 2'b10: xin_osc0_func Others: Reserved
7	RW	0x0	sclk_audpwm_div_sel sclk_audpwm_div clock mux. 1'b0: clk_gpll_mux 1'b1: clk_cpll_mux
6:0	RW	0x0b	sclk_audpwm_div_div Divide sclk_audpwm_div by (div_con + 1)

CRU_CLKSEL_CON37

Address: Operational Base + offset (0x0194)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sclk_audpwm_fracdiv_div sclk_audpwm_fracdiv fraction division register. High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL_CON38

Address: Operational Base + offset (0x0198)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	mclk_i2s1_fracdiv_v2_div mclk_i2s1_fracdiv_v2 fraction division register. High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL_CON40

Address: Operational Base + offset (0x01A0)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RW	0x0	clk_venc_core_sel clk_venc_core clock mux. 2'b00: clk_cpll_mux 2'b01: clk_gpll_mux 2'b10: clk_hpll_mux Others: Reserved
13	RO	0x0	reserved
12:8	RW	0x01	clk_venc_core_div Divide clk_venc_core by (div_con + 1)
7:6	RW	0x2	aclk_pdvepu_sel aclk_pdvepu clock mux. 2'b00: clk_cpll_mux 2'b01: clk_hpll_mux 2'b10: clk_gpll_mux Others: Reserved
5	RO	0x0	reserved
4:0	RW	0x02	aclk_pdvepu_div Divide aclk_pdvepu by (div_con + 1)

CRU_CLKSEL_CON41

Address: Operational Base + offset (0x01A4)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:13	RO	0x0	reserved
12:8	RW	0x01	hclk_pdvdec_div Divide hclk_pdvdec by (div_con + 1)
7:5	RO	0x0	reserved
4:0	RW	0x01	hclk_pdvepu_div Divide hclk_pdvepu by (div_con + 1)

CRU_CLKSEL_CON42

Address: Operational Base + offset (0x01A8)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RW	0x2	clk_vdec_core_sel clk_vdec_core clock mux. 2'b00: clk_cpll_mux 2'b01: clk_hpll_mux 2'b10: clk_gpll_mux Others: Reserved
13	RO	0x0	reserved
12:8	RW	0x03	clk_vdec_core_div Divide clk_vdec_core by (div_con + 1)
7:6	RW	0x2	aclk_pdvdec_sel aclk_pdvdec clock mux. 2'b00: clk_cpll_mux 2'b01: clk_hpll_mux 2'b10: clk_gpll_mux Others: Reserved
5	RO	0x0	reserved
4:0	RW	0x02	aclk_pdvdec_div Divide aclk_pdvdec by (div_con + 1)

CRU_CLKSEL_CON43

Address: Operational Base + offset (0x01AC)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RW	0x2	clk_vdec_hevc_ca_sel clk_vdec_hevc_ca clock mux. 2'b00: clk_cpll_mux 2'b01: clk_hpll_mux 2'b10: clk_gpll_mux Others: Reserved
13	RO	0x0	reserved
12:8	RW	0x01	clk_vdec_hevc_ca_div Divide clk_vdec_hevc_ca by (div_con + 1)
7:6	RW	0x2	clk_vdec_ca_sel clk_vdec_ca clock mux. 2'b00: clk_cpll_mux 2'b01: clk_hpll_mux 2'b10: clk_gpll_mux Others: Reserved
5	RO	0x0	reserved
4:0	RW	0x03	clk_vdec_ca_div Divide clk_vdec_ca by (div_con + 1)

CRU_CLKSEL_CON44

Address: Operational Base + offset (0x01B0)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:13	RO	0x0	reserved
12:8	RW	0x01	hclk_pdjpeg_div Divide hclk_pdjpeg by (div_con + 1)
7:6	RW	0x2	aclk_pdjpeg_sel aclk_pdjpeg clock mux. 2'b00: clk_cpll_mux 2'b01: clk_hpll_mux 2'b10: clk_gpll_mux Others: Reserved
5	RO	0x0	reserved
4:0	RW	0x03	aclk_pdjpeg_div Divide aclk_pdjpeg by (div_con + 1)

CRU_CLKSEL_CON45

Address: Operational Base + offset (0x01B4)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:13	RO	0x0	reserved
12:8	RW	0x01	hclk_pdvo_div Divide hclk_pdvo by (div_con + 1)
7	RW	0x0	aclk_pdvo_sel aclk_pdvo clock mux. 1'b0: clk_gpll_mux 1'b1: clk_cpll_mux
6:5	RO	0x0	reserved
4:0	RW	0x03	aclk_pdvo_div Divide aclk_pdvo by (div_con + 1)

CRU_CLKSEL_CON46

Address: Operational Base + offset (0x01B8)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:13	RO	0x0	reserved
12:8	RW	0x01	pclk_pdvo_div Divide pclk_pdvo by (div_con + 1)
7	RW	0x0	clk_rga_core_sel clk_rga_core clock mux. 1'b0: clk_gpll_mux 1'b1: clk_cpll_mux
6:5	RO	0x0	reserved
4:0	RW	0x02	clk_rga_core_div Divide clk_rga_core by (div_con + 1)

CRU_CLKSEL_CON47

Address: Operational Base + offset (0x01BC)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RO	0x0	reserved
11:10	RW	0x0	dclk_vop_sel dclk_vop clock mux. 2'b00: dclk_vop_div 2'b01: dclk_vop_fracdiv 2'b10: xin_osc0_func Others: Reserved
9	RO	0x0	reserved
8	RW	0x0	dclk_vop_div_sel dclk_vop_div clock mux. 1'b0: clk_gpll_mux 1'b1: clk_cpll_mux
7:0	RW	0x07	dclk_vop_div_div Divide dclk_vop_div by (div_con + 1)

CRU_CLKSEL_CON48

Address: Operational Base + offset (0x01C0)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dclk_vop_fracdiv_div dclk_vop_fracdiv fraction division register. High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL_CON49

Address: Operational Base + offset (0x01C4)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:13	RO	0x0	reserved
12:8	RW	0x01	hclk_pdvi_div Divide hclk_pdvi by (div_con + 1)
7:6	RW	0x0	aclk_pdvi_div_sel aclk_pdvi_div clock mux. 2'b00: clk_cpll_mux 2'b01: clk_gpll_mux 2'b10: clk_hpll_mux Others: Reserved
5	RO	0x0	reserved
4:0	RW	0x01	aclk_pdvi_div_div Divide aclk_pdvi_div by (div_con + 1)

CRU_CLKSEL_CON50

Address: Operational Base + offset (0x01C8)

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

Bit	Attr	Reset Value	Description
15:14	RW	0x0	clk_cif_out2io_sel clk_cif_out2io clock mux. 2'b00: xin_osc0_func 2'b01: clk_cif_out2io_div 2'b10: clk_cif_out2io_fracdiv Others: Reserved
13	RO	0x0	reserved
12:8	RW	0x01	pclk_pdvi_div Divide pclk_pdvi by (div_con + 1)
7:6	RW	0x0	clk_isp_div_sel clk_isp_div clock mux. 2'b00: clk_gpll_mux 2'b01: clk_cpll_mux 2'b10: clk_hpll_mux Others: Reserved
5	RO	0x0	reserved
4:0	RW	0x01	clk_isp_div_div Divide clk_isp_div by (div_con + 1)

CRU_CLKSEL_CON51

Address: Operational Base + offset (0x01CC)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	clk_cif_out2io_div_sel clk_cif_out2io_div clock mux. 1'b0: clk_gpll_mux 1'b1: usbphy_clk480m_mux
14	RO	0x0	reserved
13:8	RW	0x07	clk_cif_out2io_div_div Divide clk_cif_out2io_div by (div_con + 1)
7:6	RW	0x1	dclk_cif_sel dclk_cif clock mux. 2'b00: clk_gpll_mux 2'b01: clk_cpll_mux 2'b10: clk_hpll_mux Others: Reserved
5	RO	0x0	reserved
4:0	RW	0x02	dclk_cif_div Divide dclk_cif by (div_con + 1)

CRU_CLKSEL_CON52

Address: Operational Base + offset (0x01D0)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	clk_cif_out2io_fracdiv_div clk_cif_out2io_fracdiv fraction division register. High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL_CON53

Address: Operational Base + offset (0x01D4)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:13	RO	0x0	reserved
12:8	RW	0x05	hclk_pdphp_div Divide hclk_pdphp by (div_con + 1)
7	RW	0x0	aclk_pdphp_sel aclk_pdphp clock mux. 1'b0: clk_gpll_mux 1'b1: clk_cpll_mux
6:5	RO	0x0	reserved
4:0	RW	0x03	aclk_pdphp_div Divide aclk_pdphp by (div_con + 1)

CRU_CLKSEL_CON54

Address: Operational Base + offset (0x01D8)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RW	0x1	dclk_ciflite_sel dclk_ciflite clock mux. 2'b00: clk_gpll_mux 2'b01: clk_cpll_mux 2'b10: clk_hpll_mux Others: Reserved
13	RO	0x0	reserved
12:8	RW	0x02	dclk_ciflite_div Divide dclk_ciflite by (div_con + 1)
7	RW	0x0	clk_iep_core_sel clk_iep_core clock mux. 1'b0: clk_gpll_mux 1'b1: clk_cpll_mux
6:5	RO	0x0	reserved
4:0	RW	0x03	clk_iep_core_div Divide clk_iep_core by (div_con + 1)

CRU_CLKSEL_CON55

Address: Operational Base + offset (0x01DC)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RW	0x0	cclkin_sdmmc_sel cclkin_sdmmc clock mux. 2'b00: clk_gpll_mux 2'b01: clk_cpll_mux 2'b10: xin_osc0_func Others: Reserved
13:8	RO	0x00	reserved
7:0	RW	0x02	cclkin_sdmmc_div Divide cclkin_sdmmc by (div_con + 1)

CRU_CLKSEL_CON56

Address: Operational Base + offset (0x01E0)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RW	0x0	cclkin_sdio_sel cclkin_sdio clock mux. 2'b00: clk_gpll_mux 2'b01: clk_cpll_mux 2'b10: xin_osc0_func Others: Reserved
13:8	RO	0x00	reserved
7:0	RW	0x02	cclkin_sdio_div Divide cclkin_sdio by (div_con + 1)

CRU_CLKSEL_CON57

Address: Operational Base + offset (0x01E4)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RW	0x0	cclkin_emmc_sel cclkin_emmc clock mux. 2'b00: clk_gpll_mux 2'b01: clk_cpll_mux 2'b10: xin_osc0_func Others: Reserved
13:8	RO	0x00	reserved
7:0	RW	0x02	cclkin_emmc_div Divide cclkin_emmc by (div_con + 1)

CRU_CLKSEL_CON58

Address: Operational Base + offset (0x01E8)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	sclk_fspi_sel sclk_fspi clock mux. 1'b0: clk_cpll_mux 1'b1: clk_gpll_mux
14:8	RO	0x00	reserved
7:0	RW	0x05	sclk_fspi_div Divide sclk_fspi by (div_con + 1)

CRU_CLKSEL_CON59

Address: Operational Base + offset (0x01EC)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	nclk_nandc_sel nclk_nandc clock mux. 1'b0: clk_gpll_mux 1'b1: clk_cpll_mux
14:8	RO	0x00	reserved
7:0	RW	0x05	nclk_nandc_div Divide nclk_nandc by (div_con + 1)

CRU_CLKSEL_CON61

Address: Operational Base + offset (0x01F4)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	clk_gmac_ethernet_out2io_sel clk_gmac_ethernet_out2io clock mux. 1'b0: clk_cpll_mux 1'b1: clk_gpll_mux
14:13	RO	0x0	reserved
12:8	RW	0x07	clk_gmac_ethernet_out2io_div Divide clk_gmac_ethernet_out2io by (div_con + 1)
7	RW	0x0	clk_usbhost_utmi_ohci_sel clk_usbhost_utmi_ohci clock mux. 1'b0: usbphy_clk480m_mux 1'b1: clk_gpll_mux
6:5	RO	0x0	reserved
4:0	RW	0x09	clk_usbhost_utmi_ohci_div Divide clk_usbhost_utmi_ohci by (div_con + 1)

CRU_CLKSEL_CON63

Address: Operational Base + offset (0x01FC)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:13	RO	0x0	reserved
12:8	RW	0x01	pclk_pdgmac_div Divide pclk_pdgmac by (div_con + 1)
7	RW	0x0	clk_gmac_divout_sel clk_gmac_divout clock mux. 1'b0: clk_cpll_mux 1'b1: clk_gpll_mux
6:5	RO	0x0	reserved
4:0	RW	0x07	clk_gmac_divout_div Divide clk_gmac_divout by (div_con + 1)

CRU_CLKSEL_CON64

Address: Operational Base + offset (0x0200)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	clk_ddrphy_com_sel clk_ddrphy_com clock mux. 1'b0: clk_dpll_mux 1'b1: clk_gpll_mux
14:13	RO	0x0	reserved
12:8	RW	0x01	clk_ddrphy_com_div Divide clk_ddrphy_com by (div_con + 1)
7:5	RO	0x0	reserved
4:0	RW	0x0b	pclk_pdddr_pre_div Divide pclk_pdddr_pre by (div_con + 1)

CRU_CLKSEL_CON65

Address: Operational Base + offset (0x0204)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:13	RO	0x0	reserved
12	RW	0x0	aclk_pdnpu_sel aclk_pdnpu clock mux. 1'b0: aclk_pdnpu_div 1'b1: aclk_pdnpu_np5
11:10	RO	0x0	reserved
9:8	RW	0x0	aclk_pdnpu_div_sel aclk_pdnpu_div clock mux. 2'b00: clk_gpll_mux 2'b01: clk_cpll_mux 2'b10: clk_apll_mux 2'b11: clk_hpll_mux
7:4	RW	0x1	aclk_pdnpu_np5_div Divide aclk_pdnpu_np5 by (div_con + 1)
3:0	RW	0x1	aclk_pdnpu_div_div Divide aclk_pdnpu_div by (div_con + 1)

CRU_CLKSEL_CON66

Address: Operational Base + offset (0x0208)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RO	0x0	reserved
11:8	RW	0x7	hclk_pdnpu_div Divide hclk_pdnpu by (div_con + 1)
7:5	RO	0x0	reserved
4:0	RW	0x01	pclk_pdnpu_div Divide pclk_pdnpu by (div_con + 1)

CRU_CLKSEL_CON67

Address: Operational Base + offset (0x020C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:13	RO	0x0	reserved
12	RW	0x0	clk_npu_sel clk_npu clock mux. 1'b0: clk_npu_div 1'b1: clk_npu_np5
11:10	RO	0x0	reserved
9:8	RW	0x0	clk_npu_div_sel clk_npu_div clock mux. 2'b00: clk_gpll_mux 2'b01: clk_cpll_mux 2'b10: clk_apll_mux 2'b11: clk_hpll_mux
7:4	RW	0x1	clk_npu_np5_div Divide clk_npu_np5 by (div_con + 1)
3:0	RW	0x1	clk_npu_div_div Divide clk_npu_div by (div_con + 1)

CRU_CLKSEL_CON68

Address: Operational Base + offset (0x0210)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x05	reserved
7:6	RW	0x0	aclk_pdispp_div_sel aclk_pdispp_div clock mux. 2'b00: clk_cpll_mux 2'b01: clk_gpll_mux 2'b10: clk_hpll_mux Others: Reserved
5	RO	0x0	reserved
4:0	RW	0x01	aclk_pdispp_div_div Divide aclk_pdispp_div by (div_con + 1)

CRU_CLKSEL_CON69

Address: Operational Base + offset (0x0214)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:13	RO	0x0	reserved
12:8	RW	0x01	hclk_pdispp_div Divide hclk_pdispp by (div_con + 1)
7:6	RW	0x0	clk_ispp_div_sel clk_ispp_div clock mux. 2'b00: clk_cpll_mux 2'b01: clk_gpll_mux 2'b10: clk_hpll_mux Others: Reserved

Bit	Attr	Reset Value	Description
5	RO	0x0	reserved
4:0	RW	0x01	clk_ispp_div_div Divide clk_ispp_div by (div_con + 1)

CRU_CLKSEL_CON70

Address: Operational Base + offset (0x0218)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:11	RO	0x00	reserved
10:0	RW	0x000	clk_cpu_tsadc_div Divide clk_cpu_tsadc by (div_con + 1)

CRU_CLKSEL_CON71

Address: Operational Base + offset (0x021C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:11	RO	0x00	reserved
10:0	RW	0x000	clk_npu_tsadc_div Divide clk_npu_tsadc by (div_con + 1)

CRU_CLKSEL_CON72

Address: Operational Base + offset (0x0220)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:9	RO	0x00	reserved
8	RW	0x0	clk_acdcdig_i2c_sel clk_acdcdig_i2c clock mux. 1'b0: clk_gpll_mux 1'b1: xin_osc0_func
7	RO	0x0	reserved
6:0	RW	0x0b	clk_acdcdig_i2c_div Divide clk_acdcdig_i2c by (div_con + 1)

CRU_CLKSEL_CON73

Address: Operational Base + offset (0x0224)

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

Bit	Attr	Reset Value	Description
11:10	RW	0x0	mipicsi_clk_out2io_sel mipicsi_clk_out2io clock mux. 2'b00: xin_osc0_func 2'b01: mipicsi_clk_out2io_div 2'b10: mipicsi_clk_out2io_fracdiv Others: Reserved
9	RO	0x0	reserved
8	RW	0x0	mipicsi_clk_out2io_div_sel mipicsi_clk_out2io_div clock mux. 1'b0: clk_gpll_mux 1'b1: usbphy_clk480m_mux
7:5	RO	0x0	reserved
4:0	RW	0x07	mipicsi_clk_out2io_div_div Divide mipicsi_clk_out2io_div by (div_con + 1)

CRU_CLKSEL_CON74

Address: Operational Base + offset (0x0228)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	mipicsi_clk_out2io_fracdiv_div mipicsi_clk_out2io_fracdiv fraction division register. High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL_CON75

Address: Operational Base + offset (0x022C)

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

Bit	Attr	Reset Value	Description
12:8	RW	0x00	clk_testout_sel clk_testout clock mux. 5'd00: xin_osc0_func 5'd01: clk_32k 5'd02: clk_core_pre 5'd03: clk_ddrphy 5'd04: aclk_pdbus 5'd05: hclk_pdbus 5'd06: clk_npu 5'd07: aclk_pdnpu 5'd08: aclk_pdvdec 5'd09: clk_vdec_hevc_ca 5'd10: aclk_pdvepu 5'd11: clk_venc_core 5'd12: aclk_pdispp 5'd13: aclk_pdjpeg 5'd14: aclk_pdvi 5'd15: aclk_pdvo 5'd16: dclk_vop 5'd17: clk_spi1 5'd18: clk_mcu 5'd19: sclk_uart3 5'd20: aclk_pdphp 5'd21: sclk_fspi 5'd22: mclk_i2s1 5'd23: mclk_i2s2 5'd24: mclk_i2s0_rx 5'd25: mclk_i2s0_tx 5'd26: clk_gmac_src 5'd27: sclk_audpwm 5'd28: nclk_nandc 5'd29: cclkin_sdmmc 5'd30: cclkin_sdio 5'd31: cclkin_emmc
7:5	RO	0x0	reserved
4:0	RW	0x00	clk_testout_div Divide clk_testout by (div_con + 1)

CRU_CLKSEL_CON76

Address: Operational Base + offset (0x0230)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13	RW	0x0	clk_isp_sel clk_isp clock mux. 1'b0: clk_isp_div 1'b1: clk_isp_np5
12:8	RW	0x01	clk_isp_np5_div Divide clk_isp_np5 by (div_con + 1)
7:6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5	RW	0x0	aclk_pdvi_sel aclk_pdvi clock mux. 1'b0: aclk_pdvi_div 1'b1: aclk_pdvi_np5
4:0	RW	0x01	aclk_pdvi_np5_div Divide aclk_pdvi_np5 by (div_con + 1)

CRU_CLKSEL_CON77

Address: Operational Base + offset (0x0234)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13	RW	0x0	clk_ispp_sel clk_ispp clock mux. 1'b0: clk_ispp_div 1'b1: clk_ispp_np5
12:8	RW	0x01	clk_ispp_np5_div Divide clk_ispp_np5 by (div_con + 1)
7:6	RO	0x0	reserved
5	RW	0x0	aclk_pdispp_sel aclk_pdispp clock mux. 1'b0: aclk_pdispp_div 1'b1: aclk_pdispp_np5
4:0	RW	0x01	aclk_pdispp_np5_div Divide aclk_pdispp_np5 by (div_con + 1)

CRU_GATE_CON00

Address: Operational Base + offset (0x0280)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	pclk_dbg_pre_scan_en pclk_dbg_pre_scan clock gating control. When high, disable clock.
14	RW	0x0	aclk_core_pre_scan_en aclk_core_pre_scan clock gating control. When high, disable clock.
13	RW	0x0	clk_core_pre_div2_en clk_core_pre_div2 clock gating control. When high, disable clock.
12	RW	0x0	clk_core_cpupvtm_en clk_core_cpupvtm clock gating control. When high, disable clock.
11	RW	0x0	clk_cpupvtm_en clk_cpupvtm clock gating control. When high, disable clock.
10	RW	0x0	pclk_cpupvtm_en pclk_cpupvtm clock gating control. When high, disable clock.

Bit	Attr	Reset Value	Description
9	RW	0x0	clk_cpu_jtag_en clk_cpu_jtag clock gating control. When high, disable clock.
8	RW	0x0	hclk_pdcore_en hclk_pdcore_biu clock gating control. When high, disable clock.
7	RW	0x0	clk_core_pre_g_en clk_core_pre_g clock gating control. When high, disable clock.
6	RW	0x0	pclk_core_pre_en pclk_core_pre clock gating control. When high, disable clock.
5	RW	0x0	pclk_dbg_daplite_en pclk_dbg_datlite clock gating control. When high, disable clock.
4	RW	0x0	pclk_dbg_biu_en pclk_dbg_biu clock gating control. When high, disable clock.
3	RW	0x0	aclk_core_biu_en aclk_core_biu clock gating control. When high, disable clock.
2	RW	0x0	clk_core_pre_en clk_core_pre clock gating control. When high, disable clock.
1	RO	0x0	reserved
0	RW	0x0	aclk_dbg_pre_en aclk_dbg_pre clock gating control. When high, disable clock.

CRU_GATE_CON02

Address: Operational Base + offset (0x0288)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14	RW	0x0	clk_mcu_biu_en clk_mcu_biu clock gating control. When high, disable clock.
13	RW	0x0	pclk_pdbus_en pclk_pdbus clock gating control. When high, disable clock.
12	RW	0x0	hclk_pdbus_en hclk_pdbus clock gating control. When high, disable clock.
11	RW	0x0	aclk_pdbus_en aclk_pdbus clock gating control. When high, disable clock.
10	RW	0x0	aclk_pdbus_hold_biu1_en aclk_pdbus_hold_biu1 clock gating control. When high, disable clock.
9	RW	0x0	hclk_pdbus_biu3_en hclk_pdbus_biu3 clock gating control. When high, disable clock.

Bit	Attr	Reset Value	Description
8	RW	0x0	aclk_pdbus_biu3_en aclk_pdbus_biu3 clock gating control. When high, disable clock.
7	RW	0x0	hclk_pdbus_biu2_en hclk_pdbus_biu2 clock gating control. When high, disable clock.
6	RW	0x0	aclk_pdbus_biu2_en aclk_pdbus_biu2 clock gating control. When high, disable clock.
5	RW	0x0	pclk_pdbus_biu1_en pclk_pdbus_biu1 clock gating control. When high, disable clock.
4	RW	0x0	hclk_pdbus_biu1_en hclk_pdbus_biu1 clock gating control. When high, disable clock.
3	RW	0x0	aclk_pdbus_biu1_en aclk_pdbus_biu1 clock gating control. When high, disable clock.
2	RW	0x0	pclk_pdbus_pre_en pclk_pdbus_pre clock gating control. When high, disable clock.
1	RW	0x0	hclk_pdbus_pre_en hclk_pdbus_pre clock gating control. When high, disable clock.
0	RW	0x0	aclk_pdbus_pre_en aclk_pdbus_pre clock gating control. When high, disable clock.

CRU_GATE_CON03

Address: Operational Base + offset (0x028C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	clk_i2c4_en clk_i2c4 clock gating control. When high, disable clock.
14	RW	0x0	pclk_i2c4_en pclk_i2c4 clock gating control. When high, disable clock.
13	RW	0x0	clk_i2c3_en clk_i2c3 clock gating control. When high, disable clock.
12	RW	0x0	pclk_i2c3_en pclk_i2c3 clock gating control. When high, disable clock.
11	RW	0x0	clk_i2c1_en clk_i2c1 clock gating control. When high, disable clock.
10	RW	0x0	pclk_i2c1_en pclk_i2c1 clock gating control. When high, disable clock.

Bit	Attr	Reset Value	Description
9	RW	0x0	aclk_system_sram_en aclk_system_sram clock gating control. When high, disable clock.
8	RO	0x0	reserved
7	RW	0x0	pclk_dcf_en pclk_dcf clock gating control. When high, disable clock.
6	RW	0x0	aclk_dcf_en aclk_dcf clock gating control. When high, disable clock.
5	RW	0x0	clk_crypto_pka_en clk_crypto_pka clock gating control. When high, disable clock.
4	RW	0x0	clk_crypto_core_en clk_crypto_core clock gating control. When high, disable clock.
3	RW	0x0	hclk_crypto_en hclk_crypto clock gating control. When high, disable clock.
2	RW	0x0	aclk_crypto_en aclk_crypto clock gating control. When high, disable clock.
1:0	RO	0x0	reserved

CRU_GATE_CON04

Address: Operational Base + offset (0x0290)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14	RW	0x0	hclk_pdcrypt_biu_en hclk_pdcrypt_biu clock gating control. When high, disable clock.
13	RW	0x0	aclk_pdcrypt_biu_en aclk_pdcrypt_biu clock gating control. When high, disable clock.
12	RW	0x0	hclk_pdcrypt_en hclk_pdcrypt clock gating control. When high, disable clock.
11	RW	0x0	aclk_pdcrypt_en aclk_pdcrypt clock gating control. When high, disable clock.
10	RW	0x0	clk_mcu_jtag_en clk_mcu_jtag clock gating control. When high, disable clock.
9	RW	0x0	clk_mcu_rtc_en clk_mcu_rtc_ndft clock gating control. When high, disable clock.
8	RW	0x0	clk_mcu_core_en clk_mcu_core clock gating control. When high, disable clock.

Bit	Attr	Reset Value	Description
7	RW	0x0	clk_mcu_en clk_mcu clock gating control. When high, disable clock.
6	RW	0x0	clk_capture_pwm2_en clk_capture_pwm2_ndft clock gating control. When high, disable clock.
5	RW	0x0	clk_pwm2_en clk_pwm2 clock gating control. When high, disable clock.
4	RW	0x0	pclk_pwm2_en pclk_pwm2 clock gating control. When high, disable clock.
3	RW	0x0	clk_spi1_en clk_spi1 clock gating control. When high, disable clock.
2	RW	0x0	pclk_spi1_en pclk_spi1 clock gating control. When high, disable clock.
1	RW	0x0	clk_i2c5_en clk_i2c5 clock gating control. When high, disable clock.
0	RW	0x0	pclk_i2c5_en pclk_i2c5 clock gating control. When high, disable clock.

CRU_GATE_CON05

Address: Operational Base + offset (0x0294)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	sclk_uart4_en sclk_uart4 clock gating control. When high, disable clock.
14	RW	0x0	sclk_uart4_frac_en sclk_uart4_frac clock gating control. When high, disable clock.
13	RW	0x0	sclk_uart4_div_en sclk_uart4_div clock gating control. When high, disable clock.
12	RW	0x0	pclk_uart4_en pclk_uart4 clock gating control. When high, disable clock.
11	RW	0x0	sclk_uart3_en sclk_uart3 clock gating control. When high, disable clock.
10	RW	0x0	sclk_uart3_frac_en sclk_uart3_frac clock gating control. When high, disable clock.
9	RW	0x0	sclk_uart3_div_en sclk_uart3_div clock gating control. When high, disable clock.

Bit	Attr	Reset Value	Description
8	RW	0x0	pclk_uart3_en pclk_uart3 clock gating control. When high, disable clock.
7	RW	0x0	sclk_uart2_en sclk_uart2 clock gating control. When high, disable clock.
6	RW	0x0	sclk_uart2_frac_en sclk_uart2_frac clock gating control. When high, disable clock.
5	RW	0x0	sclk_uart2_div_en sclk_uart2_div clock gating control. When high, disable clock.
4	RW	0x0	pclk_uart2_en pclk_uart2 clock gating control. When high, disable clock.
3	RW	0x0	sclk_uart0_en sclk_uart0 clock gating control. When high, disable clock.
2	RW	0x0	sclk_uart0_frac_en sclk_uart0_frac clock gating control. When high, disable clock.
1	RW	0x0	sclk_uart0_div_en sclk_uart0_div clock gating control. When high, disable clock.
0	RW	0x0	pclk_uart0_en pclk_uart0 clock gating control. When high, disable clock.

CRU_GATE_CON06

Address: Operational Base + offset (0x0298)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	pclk_grf_en pclk_grf clock gating control. When high, disable clock.
14	RW	0x0	pclk_wdt_en pclk_wdt clock gating control. When high, disable clock.
13	RW	0x0	clk_timer5_en clk_timer5 clock gating control. When high, disable clock.
12	RW	0x0	clk_timer4_en clk_timer4 clock gating control. When high, disable clock.
11	RW	0x0	clk_timer3_en clk_timer3 clock gating control. When high, disable clock.
10	RW	0x0	clk_timer2_en clk_timer2 clock gating control. When high, disable clock.

Bit	Attr	Reset Value	Description
9	RW	0x0	clk_timer1_en clk_timer1 clock gating control. When high, disable clock.
8	RW	0x0	clk_timer0_en clk_timer0 clock gating control. When high, disable clock.
7	RW	0x0	pclk_timer_en pclk_timer clock gating control. When high, disable clock.
6	RW	0x0	aclk_spinlock_en aclk_spinlock clock gating control. When high, disable clock.
5	RW	0x0	clk_saradc_en clk_saradc_ndft clock gating control. When high, disable clock.
4	RW	0x0	pclk_saradc_en pclk_saradc clock gating control. When high, disable clock.
3	RW	0x0	sclk_uart5_en sclk_uart5 clock gating control. When high, disable clock.
2	RW	0x0	sclk_uart5_frac_en sclk_uart5_frac clock gating control. When high, disable clock.
1	RW	0x0	sclk_uart5_div_en sclk_uart5_div clock gating control. When high, disable clock.
0	RW	0x0	pclk_uart5_en pclk_uart5 clock gating control. When high, disable clock.

CRU_GATE_CON07

Address: Operational Base + offset (0x029C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14	RW	0x0	pclk_intmux_en pclk_intmux clock gating control. When high, disable clock.
13	RW	0x0	dclk_decom_en dclk_decom clock gating control. When high, disable clock.
12	RW	0x0	pclk_decom_en pclk_decom clock gating control. When high, disable clock.
11	RW	0x0	aclk_decom_en aclk_decom clock gating control. When high, disable clock.
10	RW	0x0	pclk_mailbox_en pclk_mailbox clock gating control. When high, disable clock.

Bit	Attr	Reset Value	Description
9	RW	0x0	clk_can_en clk_can clock gating control. When high, disable clock.
8	RW	0x0	pclk_can_en pclk_can clock gating control. When high, disable clock.
7	RW	0x0	dbclk_gpio4_en dbclk_gpio4 clock gating control. When high, disable clock.
6	RW	0x0	pclk_gpio4_en pclk_gpio4 clock gating control. When high, disable clock.
5	RW	0x0	dbclk_gpio3_en dbclk_gpio3 clock gating control. When high, disable clock.
4	RW	0x0	pclk_gpio3_en pclk_gpio3 clock gating control. When high, disable clock.
3	RW	0x0	dbclk_gpio2_en dbclk_gpio2 clock gating control. When high, disable clock.
2	RW	0x0	pclk_gpio2_en pclk_gpio2 clock gating control. When high, disable clock.
1	RW	0x0	dbclk_gpio1_en dbclk_gpio1 clock gating control. When high, disable clock.
0	RW	0x0	pclk_gpio1_en pclk_gpio1 clock gating control. When high, disable clock.

CRU_GATE_CON08

Address: Operational Base + offset (0x02A0)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:5	RO	0x000	reserved
4	RW	0x0	pclk_sgrf_en pclk_sgrf clock gating control. When high, disable clock.
3:0	RO	0x0	reserved

CRU_GATE_CON09

Address: Operational Base + offset (0x02A4)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	mclk_i2s2_fracdiv_v2_en mclk_i2s2_fracdiv_v2 clock gating control. When high, disable clock.

Bit	Attr	Reset Value	Description
14	RW	0x0	mclk_i2s0_rx_out2io_en mclk_i2s0_rx_out2io clock gating control. When high, disable clock.
13	RW	0x0	mclk_i2s0_tx_out2io_en mclk_i2s0_tx_out2io clock gating control. When high, disable clock.
12:11	RO	0x0	reserved
10	RW	0x0	mclk_i2s0_rx_src_en mclk_i2s0_rx clock gating control. When high, disable clock.
9	RW	0x0	mclk_i2s0_tx_src_en mclk_i2s0_tx clock gating control. When high, disable clock.
8	RW	0x0	mclk_i2s0_rx_fracdiv_en mclk_i2s0_rx_fracdiv clock gating control. When high, disable clock.
7	RW	0x0	mclk_i2s0_rx_div_en mclk_i2s0_rx_div clock gating control. When high, disable clock.
6	RW	0x0	mclk_i2s0_tx_fracdiv_en mclk_i2s0_tx_fracdiv clock gating control. When high, disable clock.
5	RW	0x0	mclk_i2s0_tx_div_en mclk_i2s0_tx_div clock gating control. When high, disable clock.
4	RW	0x0	hclk_i2s0_en hclk_i2s0 clock gating control. When high, disable clock.
3	RW	0x0	pclk_pdaudio_biu_en pclk_pdaudio_biu clock gating control. When high, disable clock.
2	RW	0x0	hclk_pdaudio_biu_en hclk_pdaudio_biu clock gating control. When high, disable clock.
1	RO	0x0	reserved
0	RW	0x0	hclk_pdaudio_en hclk_pdaudio clock gating control. When high, disable clock.

CRU_GATE_CON10

Address: Operational Base + offset (0x02A8)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	sclk_audpwm_en sclk_audpwm clock gating control. When high, disable clock.
14	RW	0x0	sclk_audpwm_fracdiv_en sclk_audpwm_fracdiv clock gating control. When high, disable clock.
13	RW	0x0	sclk_audpwm_div_en sclk_audpwm_div clock gating control. When high, disable clock.

Bit	Attr	Reset Value	Description
12	RW	0x0	hclk_audpwm_en hclk_audpwm clock gating control. When high, disable clock.
11	RW	0x0	mclk_pdm_en mclk_pdm clock gating control. When high, disable clock.
10	RW	0x0	hclk_pdm_en hclk_pdm clock gating control. When high, disable clock.
9	RW	0x0	mclk_i2s2_out2io_en mclk_i2s2_out2io clock gating control. When high, disable clock.
8	RW	0x0	mclk_i2s2_en mclk_i2s2 clock gating control. When high, disable clock.
7	RW	0x0	mclk_i2s2_fracdiv_en mclk_i2s2_fracdiv clock gating control. When high, disable clock.
6	RW	0x0	mclk_i2s2_div_en mclk_i2s2_div clock gating control. When high, disable clock.
5	RW	0x0	hclk_i2s2_en hclk_i2s2 clock gating control. When high, disable clock.
4	RW	0x0	mclk_i2s1_out2io_en mclk_i2s1_out2io clock gating control. When high, disable clock.
3	RW	0x0	mclk_i2s1_en mclk_i2s1 clock gating control. When high, disable clock.
2	RW	0x0	mclk_i2s1_fracdiv_en mclk_i2s1_fracdiv clock gating control. When high, disable clock.
1	RW	0x0	mclk_i2s1_div_en mclk_i2s1_div clock gating control. When high, disable clock.
0	RW	0x0	hclk_i2s1_en hclk_i2s1 clock gating control. When high, disable clock.

CRU_GATE_CON11

Address: Operational Base + offset (0x02AC)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:4	RO	0x000	reserved
3	RW	0x0	clk_acdcdig_dac_en clk_acdcdig_dac clock gating control. When high, disable clock.
2	RW	0x0	clk_acdcdig_adc_en clk_acdcdig_adc clock gating control. When high, disable clock.

Bit	Attr	Reset Value	Description
1	RW	0x0	clk_acdcdig_i2c_en clk_acdcdig_i2c clock gating control. When high, disable clock.
0	RW	0x0	pclk_acdcdig_en pclk_acdcdig clock gating control. When high, disable clock.

CRU_GATE_CON12

Address: Operational Base + offset (0x02B0)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:10	RO	0x00	reserved
9	RW	0x0	clk_iep_core_en clk_iep_core clock gating control. When high, disable clock.
8	RW	0x0	hclk_iep_en hclk_iep clock gating control. When high, disable clock.
7	RW	0x0	aclk_iep_en aclk_iep clock gating control. When high, disable clock.
6	RW	0x0	hclk_venc_en hclk_venc clock gating control. When high, disable clock.
5	RW	0x0	aclk_venc_en aclk_venc clock gating control. When high, disable clock.
4	RW	0x0	hclk_pdvepu_biu_en hclk_pdvepu_biu clock gating control. When high, disable clock.
3	RW	0x0	aclk_pdvepu_biu_en aclk_pdvepu_biu clock gating control. When high, disable clock.
2	RW	0x0	hclk_pdvepu_en hclk_pdvepu clock gating control. When high, disable clock.
1	RW	0x0	clk_venc_core_en clk_venc_core clock gating control. When high, disable clock.
0	RW	0x0	aclk_pdvepu_en aclk_pdvepu clock gating control. When high, disable clock.

CRU_GATE_CON13

Address: Operational Base + offset (0x02B4)

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

Bit	Attr	Reset Value	Description
14	RW	0x0	hclk_jpeg_en hclk_jpeg clock gating control. When high, disable clock.
13	RW	0x0	aclk_jpeg_en aclk_jpeg clock gating control. When high, disable clock.
12	RW	0x0	hclk_pdjpeg_biu_en hclk_pdjpeg_biu clock gating control. When high, disable clock.
11	RW	0x0	aclk_pdjpeg_biu_en aclk_pdjpeg_biu clock gating control. When high, disable clock.
10	RW	0x0	hclk_pdjpeg_en hclk_pdjpeg clock gating control. When high, disable clock.
9	RW	0x0	aclk_pdjpeg_en aclk_pdjpeg clock gating control. When high, disable clock.
8	RW	0x0	hclk_vdec_en hclk_vdec clock gating control. When high, disable clock.
7	RW	0x0	aclk_vdec_en aclk_vdec clock gating control. When high, disable clock.
6	RW	0x0	hclk_pdvdec_biu_en hclk_pdvdec_biu clock gating control. When high, disable clock.
5	RW	0x0	aclk_pdvdec_biu_en aclk_pdvdec_biu clock gating control. When high, disable clock.
4	RW	0x0	hclk_pdvdec_en hclk_pdvdec clock gating control. When high, disable clock.
3	RW	0x0	clk_vdec_hevc_ca_en clk_vdec_hevc_ca clock gating control. When high, disable clock.
2	RW	0x0	clk_vdec_ca_en clk_vdec_ca clock gating control. When high, disable clock.
1	RW	0x0	clk_vdec_core_en clk_vdec_core clock gating control. When high, disable clock.
0	RW	0x0	aclk_pdvdec_en aclk_pdvdec clock gating control. When high, disable clock.

CRU_GATE_CON14

Address: Operational Base + offset (0x02B8)

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

Bit	Attr	Reset Value	Description
14	RW	0x0	pclk_dsihost_en pclk_dsihost clock gating control. When high, disable clock.
13	RW	0x0	dclk_vop_en dclk_vop clock gating control. When high, disable clock.
12	RW	0x0	dclk_vop_fracdiv_en dclk_vop_fracdiv clock gating control. When high, disable clock.
11	RW	0x0	dclk_vop_div_en dclk_vop_div clock gating control. When high, disable clock.
10	RW	0x0	hclk_vop_en hclk_vop clock gating control. When high, disable clock.
9	RW	0x0	aclk_vop_en aclk_vop clock gating control. When high, disable clock.
8	RW	0x0	clk_rga_core_en clk_rga_core clock gating control. When high, disable clock.
7	RW	0x0	hclk_rga_en hclk_rga clock gating control. When high, disable clock.
6	RW	0x0	aclk_rga_en aclk_rga clock gating control. When high, disable clock.
5	RW	0x0	pclk_pdvo_biu_en pclk_pdvo_biu clock gating control. When high, disable clock.
4	RW	0x0	hclk_pdvo_biu_en hclk_pdvo_biu clock gating control. When high, disable clock.
3	RW	0x0	aclk_pdvo_biu_en aclk_pdvo_biu clock gating control. When high, disable clock.
2	RW	0x0	pclk_pdvo_en pclk_pdvo clock gating control. When high, disable clock.
1	RW	0x0	hclk_pdvo_en hclk_pdvo clock gating control. When high, disable clock.
0	RW	0x0	aclk_pdvo_en aclk_pdvo clock gating control. When high, disable clock.

CRU_GATE_CON15

Address: Operational Base + offset (0x02BC)

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

Bit	Attr	Reset Value	Description
15	RW	0x0	pclk_csihost_en pclk_csihost clock gating control. When high, disable clock.
14	RW	0x0	clk_cif_out2io_en clk_cif_out2io clock gating control. When high, disable clock.
13	RW	0x0	clk_cif_out2io_fracdiv_en clk_cif_out2io_fracdiv clock gating control. When high, disable clock.
12	RW	0x0	clk_cif_out2io_div_en clk_cif_out2io_div clock gating control. When high, disable clock.
11	RW	0x0	dclk_cif_en dclk_cif clock gating control. When high, disable clock.
10	RW	0x0	hclk_cif_en hclk_cif clock gating control. When high, disable clock.
9	RW	0x0	aclk_cif_en aclk_cif clock gating control. When high, disable clock.
8	RW	0x0	clk_isp_div_en clk_isp_div clock gating control. When high, disable clock.
7	RW	0x0	hclk_isp_en hclk_isp clock gating control. When high, disable clock.
6	RW	0x0	aclk_isp_en aclk_isp clock gating control. When high, disable clock.
5	RW	0x0	pclk_pdvi_biu_en pclk_pdvi_biu clock gating control. When high, disable clock.
4	RW	0x0	hclk_pdvi_biu_en hclk_pdvi_biu clock gating control. When high, disable clock.
3	RW	0x0	aclk_pdvi_biu_en aclk_pdvi_biu clock gating control. When high, disable clock.
2	RW	0x0	pclk_pdvi_en pclk_pdvi clock gating control. When high, disable clock.
1	RW	0x0	hclk_pdvi_en hclk_pdvi clock gating control. When high, disable clock.
0	RW	0x0	aclk_pdvi_div_en aclk_pdvi_div clock gating control. When high, disable clock.

CRU_GATE_CON16

Address: Operational Base + offset (0x02C0)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14	RW	0x0	clk_isp_np5_en clk_isp_np5 clock gating control. When high, disable clock.
13	RW	0x0	aclk_pdvi_np5_en aclk_pdvi_np5 clock gating control. When high, disable clock.
12	RW	0x0	dclk_ciflite_en dclk_ciflite clock gating control. When high, disable clock.
11	RW	0x0	hclk_ciflite_en hclk_ciflite clock gating control. When high, disable clock.
10	RW	0x0	aclk_ciflite_en aclk_ciflite clock gating control. When high, disable clock.
9	RO	0x0	reserved
8	RW	0x0	aclk_pdispp_np5_en aclk_pdispp_np5 clock gating control. When high, disable clock.
7	RW	0x0	clk_ispp_np5_en clk_ispp_np5 clock gating control. When high, disable clock.
6	RW	0x0	clk_ispp_div_en clk_ispp_div clock gating control. When high, disable clock.
5	RW	0x0	hclk_ispp_en hclk_ispp clock gating control. When high, disable clock.
4	RW	0x0	aclk_ispp_en aclk_ispp clock gating control. When high, disable clock.
3	RW	0x0	hclk_pdispp_biu_en hclk_pdispp_biu clock gating control. When high, disable clock.
2	RW	0x0	aclk_pdispp_biu_en aclk_pdispp_biu clock gating control. When high, disable clock.
1	RW	0x0	hclk_pdispp_en hclk_pdispp clock gating control. When high, disable clock.
0	RW	0x0	aclk_pdispp_div_en aclk_pdispp_div clock gating control. When high, disable clock.

CRU_GATE_CON17

Address: Operational Base + offset (0x02C4)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:10	RO	0x00	reserved
9	RW	0x0	hclk_pdsdio_biu_en hclk_pdsdio_biu clock gating control. When high, disable clock.
8	RW	0x0	hclk_pdsdio_en hclk_pdsdio clock gating control. When high, disable clock.
7	RW	0x0	hclk_pdsdcard_biu_en hclk_pdsdcard_biu clock gating control. When high, disable clock.
6	RW	0x0	hclk_pdsdcard_en hclk_pdsdcard clock gating control. When high, disable clock.
5	RW	0x0	hclk_pdphpmid_biu_en hclk_pdphpmid_biu clock gating control. When high, disable clock.
4	RW	0x0	aclk_pdphpmid_biu_en aclk_pdphpmid_biu clock gating control. When high, disable clock.
3	RW	0x0	hclk_pdphpmid_en hclk_pdphpmid clock gating control. When high, disable clock.
2	RW	0x0	aclk_pdphpmid_en aclk_pdphpmid clock gating control. When high, disable clock.
1	RW	0x0	hclk_pdphp_en hclk_pdphp clock gating control. When high, disable clock.
0	RW	0x0	aclk_pdphp_en aclk_pdphp clock gating control. When high, disable clock.

CRU_GATE_CON18

Address: Operational Base + offset (0x02C8)

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

Bit	Attr	Reset Value	Description
11	RW	0x0	hclk_fspixip_en hclk_fspixip clock gating control. When high, disable clock.
10	RW	0x0	hclk_fspi_en hclk_fspi clock gating control. When high, disable clock.
9	RW	0x0	cclkin_emmc_en cclkin_emmc clock gating control. When high, disable clock.
8	RW	0x0	hclk_emmc_en hclk_emmc clock gating control. When high, disable clock.
7	RW	0x0	cclkin_sdio_en cclkin_sdio clock gating control. When high, disable clock.
6	RW	0x0	hclk_sdio_en hclk_sdio clock gating control. When high, disable clock.
5	RW	0x0	cclkin_sdmmc_en cclkin_sdmmc clock gating control. When high, disable clock.
4	RW	0x0	hclk_sdmmc_en hclk_sdmmc clock gating control. When high, disable clock.
3	RW	0x0	hclk_pdnvm_biu_en hclk_pdnvm_biu clock gating control. When high, disable clock.
2	RO	0x0	reserved
1	RW	0x0	hclk_pdnvm_en hclk_pdnvm clock gating control. When high, disable clock.
0	RO	0x0	reserved

CRU_GATE_CON19

Address: Operational Base + offset (0x02CC)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13	RW	0x0	pclk_usbphy_host_en pclk_usbphy_host clock gating control. When high, disable clock.
12	RW	0x0	pclk_usbphy_otg_en pclk_usbphy_otg clock gating control. When high, disable clock.
11:9	RO	0x0	reserved
8	RW	0x0	clk_usbotg_ref_en clk_usbotg_ref clock gating control. When high, disable clock.
7	RW	0x0	aclk_usbotg_en aclk_usbotg clock gating control. When high, disable clock.

Bit	Attr	Reset Value	Description
6	RW	0x0	clk_usbhost_utmi_ohci_en clk_usbhost_utmi_ohci clock gating control. When high, disable clock.
5	RW	0x0	hclk_usbhost_arb_en hclk_usbhost_arb clock gating control. When high, disable clock.
4	RW	0x0	hclk_usbhost_en hclk_usbhost clock gating control. When high, disable clock.
3	RW	0x0	hclk_pdusb_biu_en hclk_pdusb_biu clock gating control. When high, disable clock.
2	RW	0x0	aclk_pdusb_biu_en aclk_pdusb_biu clock gating control. When high, disable clock.
1	RW	0x0	hclk_pdusb_en hclk_pdusb clock gating control. When high, disable clock.
0	RW	0x0	aclk_pdusb_en aclk_pdusb clock gating control. When high, disable clock.

CRU_GATE_CON20

Address: Operational Base + offset (0x02D0)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	clk_ddr_mon_en clk_ddr_mon clock gating control. When high, disable clock.
14	RO	0x0	reserved
13	RW	0x0	clk_gmac_rgmii_m1_en clk_gmac_rgmii_m1_g clock gating control. When high, disable clock.
12	RW	0x0	clk_gmac_rgmii_m0_en clk_gmac_rgmii_m0_g clock gating control. When high, disable clock.
11	RW	0x0	clk_gmac_ethernet_out2io_en clk_gmac_ethernet_out2io clock gating control. When high, disable clock.
10	RW	0x0	clk_mac_ptpref_en clk_mac_ptpref clock gating control. When high, disable clock.
9	RW	0x0	clk_mac_tx_en clk_mac_tx_src clock gating control. When high, disable clock.
8	RW	0x0	clk_mac_rx_en clk_mac_rx_src clock gating control. When high, disable clock.
7	RW	0x0	clk_mac_ref_en clk_mac_ref clock gating control. When high, disable clock.

Bit	Attr	Reset Value	Description
6	RW	0x0	clk_gmac_divout_en clk_gmac_divout clock gating control. When high, disable clock.
5	RW	0x0	pclk_gmac_en pclk_gmac clock gating control. When high, disable clock.
4	RW	0x0	aclk_gmac_en aclk_gmac clock gating control. When high, disable clock.
3	RW	0x0	pclk_pdgmac_biu_en pclk_pdgmac_biu clock gating control. When high, disable clock.
2	RW	0x0	aclk_pdgmac_biu_en aclk_pdgmac_biu clock gating control. When high, disable clock.
1	RW	0x0	pclk_pdgmac_en pclk_pdgmac clock gating control. When high, disable clock.
0	RW	0x0	aclk_pdgmac_en aclk_pdgmac clock gating control. When high, disable clock.

CRU_GATE_CON21

Address: Operational Base + offset (0x02D4)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	pclk_pdddr_en pclk_pdddr clock gating control. When high, disable clock.
14	RW	0x0	clk_ddr_standby_en clk_ddr_standby clock gating control. When high, disable clock.
13	RW	0x0	clk_ddr_dfictl_en clk_ddr_dfictl clock gating control. When high, disable clock.
12:11	RO	0x0	reserved
10	RW	0x0	clk_ddr_msch_en clk_ddr_msch clock gating control. When high, disable clock.
9	RW	0x0	aclk_ddr_split_en aclk_ddr_split clock gating control. When high, disable clock.
8	RW	0x0	clk_ddrphy_en clk_ddrphy_com clock gating control. When high, disable clock.
7	RW	0x0	tmclk_ddr_mon_en tmclk_ddr_mon_ndft clock gating control. When high, disable clock.
6	RW	0x0	pclk_ddr_msch_en pclk_ddr_msch clock gating control. When high, disable clock.

Bit	Attr	Reset Value	Description
5	RW	0x0	pclk_ddr_grf_en pclk_ddr_grf clock gating control. When high, disable clock.
4	RW	0x0	pclk_ddr_standby_en pclk_ddr_standby clock gating control. When high, disable clock.
3	RW	0x0	pclk_ddr_mon_en pclk_ddr_mon clock gating control. When high, disable clock.
2	RW	0x0	pclk_ddr_dfictl_en pclk_ddr_dfictl clock gating control. When high, disable clock.
1	RO	0x0	reserved
0	RW	0x0	pclk_pdddr_pre_en pclk_pdddr_pre clock gating control. When high, disable clock.

CRU_GATE_CON22

Address: Operational Base + offset (0x02D8)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14	RW	0x0	clk_core_npupvtm_en clk_core_npupvtm clock gating control. When high, disable clock.
13	RW	0x0	clk_npupvtm_en clk_npupvtm clock gating control. When high, disable clock.
12	RW	0x0	pclk_npupvtm_en pclk_npupvtm clock gating control. When high, disable clock.
11	RW	0x0	clk_npu_div2_en clk_npu_div2 clock gating control. When high, disable clock.
10	RW	0x0	clk_npu_np5_en clk_npu_np5 clock gating control. When high, disable clock.
9	RW	0x0	clk_npu_div_en clk_npu_div clock gating control. When high, disable clock.
8	RW	0x0	hclk_npu_en hclk_npu clock gating control. When high, disable clock.
7	RW	0x0	aclk_npu_en aclk_npu clock gating control. When high, disable clock.
6	RW	0x0	pclk_pdnpu_biu_en pclk_pdnpu_biu clock gating control. When high, disable clock.
5	RW	0x0	hclk_pdnpu_biu_en hclk_pdnpu_biu clock gating control. When high, disable clock.

Bit	Attr	Reset Value	Description
4	RW	0x0	aclk_pdnpu_biu_en aclk_pdnpu_biu clock gating control. When high, disable clock.
3	RW	0x0	pclk_pdnpu_en pclk_pdnpu clock gating control. When high, disable clock.
2	RW	0x0	hclk_pdnpu_en hclk_pdnpu clock gating control. When high, disable clock.
1	RW	0x0	aclk_pdnpu_np5_en aclk_pdnpu_np5 clock gating control. When high, disable clock.
0	RW	0x0	aclk_pdnpu_div_en aclk_pdnpu_div clock gating control. When high, disable clock.

CRU_GATE_CON23

Address: Operational Base + offset (0x02DC)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13	RW	0x0	clk_testout_en clk_testout clock gating control. When high, disable clock.
12	RW	0x0	pclk_cpuemadet_en pclk_cpuemadet clock gating control. When high, disable clock.
11	RW	0x0	pclk_topgrf_en pclk_topgrf clock gating control. When high, disable clock.
10	RW	0x0	pclk_topcru_en pclk_topcru clock gating control. When high, disable clock.
9	RW	0x0	pclk_topbiu_en pclk_topbiu clock gating control. When high, disable clock.
8	RW	0x0	pclk_pdtop_en pclk_pdtop clock gating control. When high, disable clock.
7	RW	0x0	mipicsi_clk_out2io_en mipicsi_clk_out2io clock gating control. When high, disable clock.
6	RW	0x0	mipicsi_clk_out2io_fracdiv_en mipicsi_clk_out2io_fracdiv clock gating control. When high, disable clock.
5	RW	0x0	mipicsi_clk_out2io_div_en mipicsi_clk_out2io_div clock gating control. When high, disable clock.
4	RW	0x0	pclk_dsiphy_en pclk_dsiphy clock gating control. When high, disable clock.

Bit	Attr	Reset Value	Description
3	RW	0x0	pclk_csiphy1_en pclk_csiphy1 clock gating control. When high, disable clock.
2	RW	0x0	pclk_csiphy0_en pclk_csiphy0 clock gating control. When high, disable clock.
1	RW	0x0	clk1x_ddrphy_en clk1x_ddrphy clock gating control. When high, disable clock.
0	RW	0x0	pclk_ddrphy_en pclk_ddrphy clock gating control. When high, disable clock.

CRU_GATE_CON24

Address: Operational Base + offset (0x02E0)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:7	RO	0x000	reserved
6	RW	0x0	clk_npu_mcu_biu_en clk_npu_mcu_biu clock gating control. When high, disable clock.
5	RW	0x0	clk_npu_tsadcphy_en clk_npu_tsadcphy clock gating control. When high, disable clock.
4	RW	0x0	clk_npu_tsadc_en clk_npu_tsadc clock gating control. When high, disable clock.
3	RW	0x0	pclk_npu_tsadc_en pclk_npu_tsadc clock gating control. When high, disable clock.
2	RW	0x0	clk_cpu_tsadcphy_en clk_cpu_tsadcphy clock gating control. When high, disable clock.
1	RW	0x0	clk_cpu_tsadc_en clk_cpu_tsadc clock gating control. When high, disable clock.
0	RW	0x0	pclk_cpu_tsadc_en pclk_cpu_tsadc clock gating control. When high, disable clock.

CRU_SOFTRST_CON00

Address: Operational Base + offset (0x0300)

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

Bit	Attr	Reset Value	Description
13	RW	0x0	aresetn_core_biu When high, reset relative logic.
12	RW	0x0	nl2reset When high, reset relative logic.
11	RW	0x0	ndbgreset3 When high, reset relative logic.
10	RW	0x0	ndbgreset2 When high, reset relative logic.
9	RW	0x0	ndbgreset1 When high, reset relative logic.
8	RW	0x0	ndbgreset0 When high, reset relative logic.
7	RW	0x0	ncoreset3 When high, reset relative logic.
6	RW	0x0	ncoreset2 When high, reset relative logic.
5	RW	0x0	ncoreset1 When high, reset relative logic.
4	RW	0x0	ncoreset0 When high, reset relative logic.
3	RW	0x0	ncoreporeset3 When high, reset relative logic.
2	RW	0x0	ncoreporeset2 When high, reset relative logic.
1	RW	0x0	ncoreporeset1 When high, reset relative logic.
0	RW	0x0	ncoreporeset0 When high, reset relative logic.

CRU_SOFTRST_CON01

Address: Operational Base + offset (0x0304)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	aresetn_system_sram When high, reset relative logic.
14	RW	0x0	presetn_dcf When high, reset relative logic.
13	RW	0x0	aresetn_dcf When high, reset relative logic.
12:11	RO	0x0	reserved
10	RW	0x0	resetn_mcu_biu When high, reset relative logic.
9	RW	0x0	hresetn_pdcore_biu When high, reset relative logic.
8	RW	0x0	presetn_dbg_biu When high, reset relative logic.
7	RW	0x0	aresetn_pdbus_hold_biu1 When high, reset relative logic.
6	RW	0x0	hresetn_pdbus_biu3 When high, reset relative logic.

Bit	Attr	Reset Value	Description
5	RW	0x0	aresetn_pdbus_biu3 When high, reset relative logic.
4	RW	0x0	hresetn_pdbus_biu2 When high, reset relative logic.
3	RW	0x0	aresetn_pdbus_biu2 When high, reset relative logic.
2	RW	0x0	presetn_pdbus_biu1 When high, reset relative logic.
1	RW	0x0	hresetn_pdbus_biu1 When high, reset relative logic.
0	RW	0x0	aresetn_pdbus_biu1 When high, reset relative logic.

CRU_SOFTRST_CON02

Address: Operational Base + offset (0x0308)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14	RW	0x0	aresetn_spinlock When high, reset relative logic.
13	RW	0x0	resetn_pwm2 When high, reset relative logic.
12	RW	0x0	presetn_pwm2 When high, reset relative logic.
11	RO	0x0	reserved
10	RW	0x1	resetn_mcu_core When high, reset relative logic.
9	RW	0x0	resetn_spi1 When high, reset relative logic.
8	RW	0x0	presetn_spi1 When high, reset relative logic.
7	RW	0x0	resetn_i2c5 When high, reset relative logic.
6	RW	0x0	presetn_i2c5 When high, reset relative logic.
5	RW	0x0	resetn_i2c4 When high, reset relative logic.
4	RW	0x0	presetn_i2c4 When high, reset relative logic.
3	RW	0x0	resetn_i2c3 When high, reset relative logic.
2	RW	0x0	presetn_i2c3 When high, reset relative logic.
1	RW	0x0	resetn_i2c1 When high, reset relative logic.
0	RW	0x0	presetn_i2c1 When high, reset relative logic.

CRU_SOFTRST_CON03

Address: Operational Base + offset (0x030C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	presetn_mailbox When high, reset relative logic.
14	RW	0x0	presetn_timer When high, reset relative logic.
13	RW	0x0	presetn_grf When high, reset relative logic.
12	RO	0x0	reserved
11	RW	0x0	presetn_saradc When high, reset relative logic.
10	RW	0x0	presetn_wdt When high, reset relative logic.
9	RW	0x0	sresetn_uart5 When high, reset relative logic.
8	RW	0x0	presetn_uart5 When high, reset relative logic.
7	RW	0x0	sresetn_uart4 When high, reset relative logic.
6	RW	0x0	presetn_uart4 When high, reset relative logic.
5	RW	0x0	sresetn_uart3 When high, reset relative logic.
4	RW	0x0	presetn_uart3 When high, reset relative logic.
3	RW	0x0	sresetn_uart2 When high, reset relative logic.
2	RW	0x0	presetn_uart2 When high, reset relative logic.
1	RW	0x0	sresetn_uart0 When high, reset relative logic.
0	RW	0x0	presetn_uart0 When high, reset relative logic.

CRU_SOFTRST_CON04

Address: Operational Base + offset (0x0310)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	dbresetn_gpio4 When high, reset relative logic.
14	RW	0x0	presetn_gpio4 When high, reset relative logic.
13	RW	0x0	dbresetn_gpio3 When high, reset relative logic.
12	RW	0x0	presetn_gpio3 When high, reset relative logic.
11	RW	0x0	dbresetn_gpio2 When high, reset relative logic.

Bit	Attr	Reset Value	Description
10	RW	0x0	presetn_gpio2 When high, reset relative logic.
9	RW	0x0	dbresetn_gpio1 When high, reset relative logic.
8	RW	0x0	presetn_gpio1 When high, reset relative logic.
7	RO	0x0	reserved
6	RW	0x0	presetn_intmux When high, reset relative logic.
5	RW	0x0	resetn_timer5 When high, reset relative logic.
4	RW	0x0	resetn_timer4 When high, reset relative logic.
3	RW	0x0	resetn_timer3 When high, reset relative logic.
2	RW	0x0	resetn_timer2 When high, reset relative logic.
1	RW	0x0	resetn_timer1 When high, reset relative logic.
0	RW	0x0	resetn_timer0 When high, reset relative logic.

CRU_SOFTRST_CON05

Address: Operational Base + offset (0x0314)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	presetn_sgrf When high, reset relative logic.
14	RO	0x0	reserved
13	RW	0x0	resetn_crypto_pka When high, reset relative logic.
12	RW	0x0	resetn_crypto_core When high, reset relative logic.
11	RW	0x0	hresetn_crypto When high, reset relative logic.
10	RW	0x0	aresetn_crypto When high, reset relative logic.
9	RW	0x0	hresetn_pdcrypt_biu When high, reset relative logic.
8	RW	0x0	aresetn_pdcrypt_biu When high, reset relative logic.
7	RW	0x0	dresetn_decom When high, reset relative logic.
6	RW	0x0	presetn_decom When high, reset relative logic.
5	RW	0x0	aresetn_decom When high, reset relative logic.
4:2	RO	0x0	reserved
1	RW	0x0	resetn_can When high, reset relative logic.

Bit	Attr	Reset Value	Description
0	RW	0x0	presetn_can When high, reset relative logic.

CRU_SOFTRST_CON06

Address: Operational Base + offset (0x0318)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14	RW	0x0	resetn_acdcdig When high, reset relative logic.
13	RW	0x0	presetn_acdcdig When high, reset relative logic.
12	RW	0x0	sresetn_audpwm When high, reset relative logic.
11	RW	0x0	hresetn_audpwm When high, reset relative logic.
10	RW	0x0	mresetn_pdm When high, reset relative logic.
9	RW	0x0	hresetn_pdm When high, reset relative logic.
8	RW	0x0	mresetn_i2s2 When high, reset relative logic.
7	RW	0x0	hresetn_i2s2 When high, reset relative logic.
6	RW	0x0	mresetn_i2s1 When high, reset relative logic.
5	RW	0x0	hresetn_i2s1 When high, reset relative logic.
4	RW	0x0	mresetn_i2s0_rx When high, reset relative logic.
3	RW	0x0	mresetn_i2s0_tx When high, reset relative logic.
2	RW	0x0	hresetn_i2s0 When high, reset relative logic.
1	RW	0x0	presetn_pdaudio_biu When high, reset relative logic.
0	RW	0x0	hresetn_pdaudio_biu When high, reset relative logic.

CRU_SOFTRST_CON07

Address: Operational Base + offset (0x031C)

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

Bit	Attr	Reset Value	Description
13	RW	0x0	hresetn_pdjpeg_biu When high, reset relative logic.
12	RW	0x0	aresetn_pdjpeg_biu When high, reset relative logic.
11	RW	0x0	resetn_vdec_hevc_ca When high, reset relative logic.
10	RW	0x0	resetn_vdec_ca When high, reset relative logic.
9	RW	0x0	resetn_vdec_core When high, reset relative logic.
8	RW	0x0	hresetn_vdec When high, reset relative logic.
7	RW	0x0	aresetn_vdec When high, reset relative logic.
6	RW	0x0	hresetn_pdvdec_biu When high, reset relative logic.
5	RW	0x0	aresetn_pdvdec_biu When high, reset relative logic.
4	RW	0x0	resetn_venc_core When high, reset relative logic.
3	RW	0x0	hresetn_venc When high, reset relative logic.
2	RW	0x0	aresetn_venc When high, reset relative logic.
1	RW	0x0	hresetn_pdvepu_biu When high, reset relative logic.
0	RW	0x0	aresetn_pdvepu_biu When high, reset relative logic.

CRU_SOFTRST_CON08

Address: Operational Base + offset (0x0320)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14	RW	0x0	rxpresetn_isp When high, reset relative logic.
13	RW	0x0	resetn_iep_core When high, reset relative logic.
12	RW	0x0	hresetn_iep When high, reset relative logic.
11	RW	0x0	aresetn_iep When high, reset relative logic.
10	RW	0x0	presetn_dsihost When high, reset relative logic.
9	RW	0x0	resetn_txbytehs_dsihost When high, reset relative logic.
8	RW	0x0	dresetn_vop When high, reset relative logic.
7	RW	0x0	hresetn_vop When high, reset relative logic.

Bit	Attr	Reset Value	Description
6	RW	0x0	aresetn_vop When high, reset relative logic.
5	RW	0x0	resetn_rga_core When high, reset relative logic.
4	RW	0x0	hresetn_rga When high, reset relative logic.
3	RW	0x0	aresetn_rga When high, reset relative logic.
2	RW	0x0	presetn_pdvo_biu When high, reset relative logic.
1	RW	0x0	hresetn_pdvo_biu When high, reset relative logic.
0	RW	0x0	aresetn_pdvo_biu When high, reset relative logic.

CRU_SOFTRST_CON09

Address: Operational Base + offset (0x0324)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	presetn_csihost When high, reset relative logic.
14	RW	0x0	resetn_ispp When high, reset relative logic.
13	RW	0x0	hresetn_ispp When high, reset relative logic.
12	RW	0x0	aresetn_ispp When high, reset relative logic.
11	RW	0x0	hresetn_pdispp_biu When high, reset relative logic.
10	RW	0x0	aresetn_pdispp_biu When high, reset relative logic.
9	RW	0x0	rxpresetn_cif When high, reset relative logic.
8	RW	0x0	iresetn_cif When high, reset relative logic.
7	RW	0x0	presetn_cif When high, reset relative logic.
6	RW	0x0	dresetn_cif When high, reset relative logic.
5	RW	0x0	hresetn_cif When high, reset relative logic.
4	RW	0x0	aresetn_cif When high, reset relative logic.
3	RW	0x0	resetn_isp When high, reset relative logic.
2	RW	0x0	presetn_pdvi_biu When high, reset relative logic.
1	RW	0x0	hresetn_pdvi_biu When high, reset relative logic.
0	RW	0x0	aresetn_pdvi_biu When high, reset relative logic.

CRU_SOFTRST_CON10

Address: Operational Base + offset (0x0328)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14	RW	0x0	hresetn_pdsdio_biu When high, reset relative logic.
13	RW	0x0	hresetn_pdsdcard_biu When high, reset relative logic.
12	RO	0x0	reserved
11	RW	0x0	nresetn_nandc When high, reset relative logic.
10	RW	0x0	hresetn_nandc When high, reset relative logic.
9	RW	0x0	sresetn_fspi When high, reset relative logic.
8	RW	0x0	hresetn_fspixip When high, reset relative logic.
7	RW	0x0	hresetn_fspi When high, reset relative logic.
6	RW	0x0	hresetn_emmc When high, reset relative logic.
5	RW	0x0	hresetn_sdio When high, reset relative logic.
4	RW	0x0	hresetn_sdmmc When high, reset relative logic.
3	RW	0x0	hresetn_pdnvm_biu When high, reset relative logic.
2	RO	0x0	reserved
1	RW	0x0	hresetn_pdphpmid_biu When high, reset relative logic.
0	RW	0x0	aresetn_pdphpmid_biu When high, reset relative logic.

CRU_SOFTRST_CON11

Address: Operational Base + offset (0x032C)

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

Bit	Attr	Reset Value	Description
8	RW	0x0	resetn_usbphypor_otg When high, reset relative logic.
7	RW	0x0	presetn_usbphy_host When high, reset relative logic.
6	RW	0x0	presetn_usbphy_otg When high, reset relative logic.
5	RW	0x0	aresetn_usbotg When high, reset relative logic.
4	RW	0x0	resetn_usbhost_utmi When high, reset relative logic.
3	RW	0x0	hresetn_usbhost_arb When high, reset relative logic.
2	RW	0x0	hresetn_usbhost When high, reset relative logic.
1	RW	0x0	hresetn_pdusb_biu When high, reset relative logic.
0	RW	0x0	aresetn_pdusb_biu When high, reset relative logic.

CRU_SOFTRST_CON12

Address: Operational Base + offset (0x0330)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	resetn_ddrphy When high, reset relative logic.
14	RW	0x0	presetn_ddrphy When high, reset relative logic.
13	RW	0x0	resetn_npumcu_biu When high, reset relative logic.
12	RO	0x1	reserved
11	RW	0x0	resetn_ddr_standby When high, reset relative logic.
10	RW	0x0	resetn_ddr_dfictl When high, reset relative logic.
9:8	RO	0x0	reserved
7	RW	0x0	resetn_ddr_msch When high, reset relative logic.
6	RW	0x0	aresetn_ddr_split When high, reset relative logic.
5	RW	0x0	presetn_ddr_msch When high, reset relative logic.
4	RW	0x0	presetn_ddr_grf When high, reset relative logic.
3	RW	0x0	presetn_ddr_standby When high, reset relative logic.
2	RW	0x0	presetn_ddr_mon When high, reset relative logic.
1	RW	0x0	presetn_ddr_dfictl When high, reset relative logic.
0	RO	0x0	reserved

CRU_SOFTRST_CON13

Address: Operational Base + offset (0x0334)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	rxpresetn_ciflite When high, reset relative logic.
14	RW	0x0	dresetn_ciflite When high, reset relative logic.
13	RW	0x0	hresetn_ciflite When high, reset relative logic.
12	RW	0x0	aresetn_ciflite When high, reset relative logic.
11	RO	0x0	reserved
10	RW	0x0	resetn_npu_tsadcphy When high, reset relative logic.
9	RW	0x0	resetn_npu_tsadc When high, reset relative logic.
8	RW	0x0	presetn_npu_tsadc When high, reset relative logic.
7	RW	0x0	resetn_npupvtm When high, reset relative logic.
6	RW	0x0	presetn_npupvtm When high, reset relative logic.
5	RW	0x0	resetn_npu When high, reset relative logic.
4	RW	0x0	hresetn_npu When high, reset relative logic.
3	RW	0x0	aresetn_npu When high, reset relative logic.
2	RW	0x0	presetn_pdnpu_biu When high, reset relative logic.
1	RW	0x0	hresetn_pdnpu_biu When high, reset relative logic.
0	RW	0x0	aresetn_pdnpu_biu When high, reset relative logic.

CRU_SOFTRST_CON14

Address: Operational Base + offset (0x0338)

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

Bit	Attr	Reset Value	Description
8	RW	0x0	presetn_cpu_tsadc When high, reset relative logic.
7	RO	0x0	reserved
6	RW	0x0	presetn_dsiphy When high, reset relative logic.
5	RW	0x0	presetn_csiphy1 When high, reset relative logic.
4	RW	0x0	presetn_csiphy0 When high, reset relative logic.
3	RW	0x0	presetn_cpuemadet When high, reset relative logic.
2	RW	0x0	presetn_topgrf When high, reset relative logic.
1	RW	0x0	presetn_topcru When high, reset relative logic.
0	RW	0x0	presetn_topbiu When high, reset relative logic.

CRU_SSCGTBL_CON0

Address: Operational Base + offset (0x0380)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	ssgtbl3 External wave table 3
23:16	RW	0x00	ssgtbl2 External wave table 2
15:8	RW	0x00	ssgtbl1 External wave table 1
7:0	RW	0x00	ssgtbl0 External wave table 0

CRU_SSCGTBL_CON1

Address: Operational Base + offset (0x0384)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	ssgtbl7 External wave table 7
23:16	RW	0x00	ssgtbl6 External wave table 6
15:8	RW	0x00	ssgtbl5 External wave table 5
7:0	RW	0x00	ssgtbl4 External wave table 4

CRU_SSCGTBL_CON2

Address: Operational Base + offset (0x0388)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	ssgtbl11 External wave table 11
23:16	RW	0x00	ssgtbl10 External wave table 10
15:8	RW	0x00	ssgtbl9 External wave table 9
7:0	RW	0x00	ssgtbl8 External wave table 8

CRU_SSCGTBL_CON3

Address: Operational Base + offset (0x038C)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	ssgtbl15 External wave table 15
23:16	RW	0x00	ssgtbl14 External wave table 14
15:8	RW	0x00	ssgtbl13 External wave table 13
7:0	RW	0x00	ssgtbl12 External wave table 12

CRU_SSCGTBL_CON4

Address: Operational Base + offset (0x0390)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	ssgtbl19 External wave table 19
23:16	RW	0x00	ssgtbl18 External wave table 18
15:8	RW	0x00	ssgtbl17 External wave table 17
7:0	RW	0x00	ssgtbl16 External wave table 16

CRU_SSCGTBL_CON5

Address: Operational Base + offset (0x0394)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	ssgtbl23 External wave table 23
23:16	RW	0x00	ssgtbl22 External wave table 22
15:8	RW	0x00	ssgtbl21 External wave table 21
7:0	RW	0x00	ssgtbl20 External wave table 20

CRU_SSCGTBL_CON6

Address: Operational Base + offset (0x0398)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	ssgtbl27 External wave table 27
23:16	RW	0x00	ssgtbl26 External wave table 26
15:8	RW	0x00	ssgtbl25 External wave table 25
7:0	RW	0x00	ssgtbl24 External wave table 24

CRU_SSCGTBL_CON7

Address: Operational Base + offset (0x039C)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	ssgtbl31 External wave table 31
23:16	RW	0x00	ssgtbl30 External wave table 30

Bit	Attr	Reset Value	Description
15:8	RW	0x00	ssgtbl29 External wave table 29
7:0	RW	0x00	ssgtbl28 External wave table 28

CRU_SSCGTBL_CON8

Address: Operational Base + offset (0x03A0)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	ssgtbl35 External wave table 35
23:16	RW	0x00	ssgtbl34 External wave table 34
15:8	RW	0x00	ssgtbl33 External wave table 33
7:0	RW	0x00	ssgtbl32 External wave table 32

CRU_SSCGTBL_CON9

Address: Operational Base + offset (0x03A4)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	ssgtbl39 External wave table 39
23:16	RW	0x00	ssgtbl38 External wave table38
15:8	RW	0x00	ssgtbl37 External wave table 37
7:0	RW	0x00	ssgtbl36 External wave table 36

CRU_SSCGTBL_CON10

Address: Operational Base + offset (0x03A8)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	ssgtbl43 External wave table 43
23:16	RW	0x00	ssgtbl42 External wave table 42
15:8	RW	0x00	ssgtbl41 External wave table 41
7:0	RW	0x00	ssgtbl40 External wave table 40

CRU_SSCGTBL_CON11

Address: Operational Base + offset (0x03AC)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	ssgtbl47 External wave table 47
23:16	RW	0x00	ssgtbl46 External wave table 46
15:8	RW	0x00	ssgtbl45 External wave table 45
7:0	RW	0x00	ssgtbl44 External wave table 44

CRU_SSCGTBL_CON12

Address: Operational Base + offset (0x03B0)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	ssgtbl51 External wave table 51
23:16	RW	0x00	ssgtbl50 External wave table 50
15:8	RW	0x00	ssgtbl49 External wave table 49
7:0	RW	0x00	ssgtbl48 External wave table 48

CRU_SSCGTBL_CON13

Address: Operational Base + offset (0x03B4)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	ssgtbl55 External wave table 55
23:16	RW	0x00	ssgtbl54 External wave table 54
15:8	RW	0x00	ssgtbl53 External wave table 53
7:0	RW	0x00	ssgtbl52 External wave table 52

CRU_SSCGTBL_CON14

Address: Operational Base + offset (0x03B8)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	ssgtbl59 External wave table 59
23:16	RW	0x00	ssgtbl58 External wave table 58
15:8	RW	0x00	ssgtbl57 External wave table 57
7:0	RW	0x00	ssgtbl56 External wave table 56

CRU_SSCGTBL_CON15

Address: Operational Base + offset (0x03BC)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	ssgtbl63 External wave table 63
23:16	RW	0x00	ssgtbl62 External wave table 62
15:8	RW	0x00	ssgtbl61 External wave table 61
7:0	RW	0x00	ssgtbl60 External wave table 60

CRU_SSCGTBL_CON16

Address: Operational Base + offset (0x03C0)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	ssgtbl67 External wave table 67
23:16	RW	0x00	ssgtbl66 External wave table 66

Bit	Attr	Reset Value	Description
15:8	RW	0x00	ssgtbl65 External wave table 65
7:0	RW	0x00	ssgtbl64 External wave table 64

CRU_SSCGTBL_CON17

Address: Operational Base + offset (0x03C4)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	ssgtbl71 External wave table 71
23:16	RW	0x00	ssgtbl70 External wave table 70
15:8	RW	0x00	ssgtbl69 External wave table 69
7:0	RW	0x00	ssgtbl68 External wave table 68

CRU_SSCGTBL_CON18

Address: Operational Base + offset (0x03C8)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	ssgtbl75 External wave table 75
23:16	RW	0x00	ssgtbl74 External wave table 74
15:8	RW	0x00	ssgtbl73 External wave table 73
7:0	RW	0x00	ssgtbl72 External wave table 72

CRU_SSCGTBL_CON19

Address: Operational Base + offset (0x03CC)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	ssgtbl79 External wave table 79
23:16	RW	0x00	ssgtbl78 External wave table 78
15:8	RW	0x00	ssgtbl77 External wave table 77
7:0	RW	0x00	ssgtbl76 External wave table 76

CRU_SSCGTBL_CON20

Address: Operational Base + offset (0x03D0)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	ssgtbl83 External wave table 83
23:16	RW	0x00	ssgtbl82 External wave table 82
15:8	RW	0x00	ssgtbl81 External wave table 81
7:0	RW	0x00	ssgtbl80 External wave table 80

CRU_SSCGTBL_CON21

Address: Operational Base + offset (0x03D4)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	ssgtbl87 External wave table 87
23:16	RW	0x00	ssgtbl86 External wave table 86
15:8	RW	0x00	ssgtbl85 External wave table 85
7:0	RW	0x00	ssgtbl84 External wave table 84

CRU_SSCGTBL_CON22

Address: Operational Base + offset (0x03D8)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	ssgtbl91 External wave table 91
23:16	RW	0x00	ssgtbl90 External wave table 90
15:8	RW	0x00	ssgtbl89 External wave table 89
7:0	RW	0x00	ssgtbl88 External wave table 88

CRU_SSCGTBL_CON23

Address: Operational Base + offset (0x03DC)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	ssgtbl95 External wave table 95
23:16	RW	0x00	ssgtbl94 External wave table 94
15:8	RW	0x00	ssgtbl93 External wave table 93
7:0	RW	0x00	ssgtbl92 External wave table 92

CRU_SSCGTBL_CON24

Address: Operational Base + offset (0x03E0)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	ssgtbl99 External wave table 99
23:16	RW	0x00	ssgtbl98 External wave table 98
15:8	RW	0x00	ssgtbl97 External wave table 97
7:0	RW	0x00	ssgtbl96 External wave table 96

CRU_SSCGTBL_CON25

Address: Operational Base + offset (0x03E4)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	ssgtbl103 External wave table 103
23:16	RW	0x00	ssgtbl102 External wave table 102

Bit	Attr	Reset Value	Description
15:8	RW	0x00	ssgtbl101 External wave table 101
7:0	RW	0x00	ssgtbl100 External wave table 100

CRU_SSCGTBL_CON26

Address: Operational Base + offset (0x03E8)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	ssgtbl107 External wave table 107
23:16	RW	0x00	ssgtbl106 External wave table 106
15:8	RW	0x00	ssgtbl105 External wave table 105
7:0	RW	0x00	ssgtbl104 External wave table 104

CRU_SSCGTBL_CON27

Address: Operational Base + offset (0x03EC)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	ssgtbl111 External wave table 111
23:16	RW	0x00	ssgtbl110 External wave table 110
15:8	RW	0x00	ssgtbl109 External wave table 109
7:0	RW	0x00	ssgtbl108 External wave table 108

CRU_SSCGTBL_CON28

Address: Operational Base + offset (0x03F0)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	ssgtbl115 External wave table 115
23:16	RW	0x00	ssgtbl114 External wave table 114
15:8	RW	0x00	ssgtbl113 External wave table 113
7:0	RW	0x00	ssgtbl112 External wave table 112

CRU_SSCGTBL_CON29

Address: Operational Base + offset (0x03F4)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	ssgtbl119 External wave table 119
23:16	RW	0x00	ssgtbl118 External wave table 118
15:8	RW	0x00	ssgtbl117 External wave table 117
7:0	RW	0x00	ssgtbl116 External wave table 116

CRU_SSCGTBL_CON30

Address: Operational Base + offset (0x03F8)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	ssgtbl123 External wave table 123
23:16	RW	0x00	ssgtbl122 External wave table 122
15:8	RW	0x00	ssgtbl121 External wave table 121
7:0	RW	0x00	ssgtbl120 External wave table 120

CRU_SSCGTBL_CON31

Address: Operational Base + offset (0x03FC)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	ssgtbl127 External wave table 127
23:16	RW	0x00	ssgtbl126 External wave table 126
15:8	RW	0x00	ssgtbl125 External wave table 125
7:0	RW	0x00	ssgtbl124 External wave table 124

CRU_GLB_CNT_TH

Address: Operational Base + offset (0x0400)

Bit	Attr	Reset Value	Description
31:0	RW	0x00640064	global_reset_counter_threshold Global soft reset, WDT reset or tsadc_shut reset asserted time counter threshold. Measured in OSC clock cycles

CRU_GLB_RST_ST

Address: Operational Base + offset (0x0404)

Bit	Attr	Reset Value	Description
31:7	RO	0x0000000	reserved
6	RW	0x0	wdt_rst_pos WDT reset flag. 1'b0: WDT reset inactive 1'b1: WDT reset active
5	RW	0x0	snd_glb_wdt_rst_st Second global WDT triggered reset flag. 1'b0: Last hot reset is not second global WDT triggered reset 1'b1: Last hot reset is second global WDT triggered reset
4	RW	0x0	fst_glb_wdt_rst_st First global WDT triggered reset flag. 1'b0: Last hot reset is not first global WDT triggered reset 1'b1: Last hot reset is first global WDT triggered reset
3	RW	0x0	snd_glb_tsadc_rst_st Second global TSADC triggered reset flag. 1'b0: Last hot reset is not second global TSADC triggered reset 1'b1: Last hot reset is second global TSADC triggered reset
2	RW	0x0	fst_glb_tsadc_rst_st First global TSADC triggered reset flag. 1'b0: Last hot reset is not first global TSADC triggered reset 1'b1: Last hot reset is first global TSADC triggered reset

Bit	Attr	Reset Value	Description
1	RW	0x0	snd_glb_rst_st Second global rst flag. 1'b0: Last hot reset is not second global reset 1'b1: Last hot reset is second global reset
0	RW	0x0	fst_glb_rst_st First global rst flag. 1'b0: Last hot reset is not first global reset 1'b1: Last hot reset is first global reset

CRU_GLB_SRST_FST_VALUE

Address: Operational Base + offset (0x0408)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15:0	RW	0x0000	glb_srst_fst_value Global software reset first value

CRU_GLB_SRST_SND_VALUE

Address: Operational Base + offset (0x040C)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15:0	RW	0x0000	glb_srst_snd_value Global software reset second value

CRU_GLB_RST_CON

Address: Operational Base + offset (0x0410)

Bit	Attr	Reset Value	Description
31:13	RO	0x00000	reserved
12	RW	0x0	cru_wdt_con WDT reset selection. 1'b0: Second reset 1'b1: First reset
11:9	RO	0x0	reserved
8	RW	0x0	cru_wdt_en 1'b0: Disable WDT reset 1'b1: Enable WDT reset
7:5	RO	0x0	reserved
4	RW	0x0	pmu_srst_wdt_rst_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_srst_en 1'b0: Enable global reset as PMU reset source 1'b1: Disable global reset as PMU reset source
2	RW	0x0	pmu_srst_glb_srst_sel 1'b0: First global reset as PMU reset source 1'b1: Second global reset as PMU reset source
1	RO	0x0	reserved
0	RW	0x0	tsadc_glb_srst_ctrl 1'b0: TSADC trigger second global reset 1'b1: TSADC trigger first global reset

CRU_SDMMC_CON0

Address: Operational Base + offset (0x0440)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RO	0x0	reserved
11	RW	0x0	drv_sel Driver clock source selection. 1'b0: Use clock after phase shift 1'b1: Use clock after phase shift and delay line
10:3	RW	0x00	drv_delaynum Element number in delay line for driver clock
2:1	RW	0x2	drv_degree Phase shift for driver clock. 2'b00: 0 degree 2'b01: 90 degree 2'b10: 180 degree 2'b11: 270 degree
0	RW	0x0	init_state Soft initial state for phase shift, high active

CRU_SDMMC_CON1

Address: Operational Base + offset (0x0444)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RO	0x0	reserved
11	RW	0x0	smp_sel Sample clock source selection. 1'b0: Use clock after phase shift 1'b1: Use clock after phase shift and delay line
10:3	RW	0x00	smp_delaynum Element number in delay line for sample clock
2:1	RW	0x2	smp_degree Phase shift for sample clock. 2'b00: 0 degree 2'b01: 90 degree 2'b10: 180 degree 2'b11: 270 degree
0	RO	0x0	reserved

CRU_SDIO_CON0

Address: Operational Base + offset (0x0448)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RO	0x0	reserved
11	RW	0x0	drv_sel Driver clock source selection. 1'b0: Use clock after phase shift 1'b1: Use clock after phase shift and delay line

Bit	Attr	Reset Value	Description
10:3	RW	0x00	drv_delaynum Element number in delay line for driver clock
2:1	RW	0x2	drv_degree Phase shift for driver clock. 2'b00: 0 degree 2'b01: 90 degree 2'b10: 180 degree 2'b11: 270 degree
0	RW	0x0	init_state Soft initial state for phase shift, high active

CRU_SDIO_CON1

Address: Operational Base + offset (0x044C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RO	0x0	reserved
11	RW	0x0	smp_sel Sample clock source selection. 1'b0: Use clock after phase shift 1'b1: Use clock after phase shift and delay line
10:3	RW	0x00	smp_delaynum Element number in delay line for sample clock
2:1	RW	0x2	smp_degree Phase shift for sample clock. 2'b00: 0 degree 2'b01: 90 degree 2'b10: 180 degree 2'b11: 270 degree
0	RO	0x0	reserved

CRU_EMMC_CON0

Address: Operational Base + offset (0x0450)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RO	0x0	reserved
11	RW	0x0	drv_sel Driver clock source selection. 1'b0: Use clock after phase shift 1'b1: Use clock after phase shift and delay line
10:3	RW	0x00	drv_delaynum Element number in delay line for driver clock
2:1	RW	0x2	drv_degree Phase shift for driver clock. 2'b00: 0 degree 2'b01: 90 degree 2'b10: 180 degree 2'b11: 270 degree
0	RW	0x0	init_state Soft initial state for phase shift, high active

CRU_EMMC_CON1

Address: Operational Base + offset (0x0454)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RO	0x0	reserved
11	RW	0x0	smp_sel Sample clock source selection. 1'b0: Use clock after phase shift 1'b1: Use clock after phase shift and delay line
10:3	RW	0x00	smp_delaynum Element number in delay line for sample clock
2:1	RW	0x2	smp_degree Phase shift for sample clock. 2'b00: 0 degree 2'b01: 90 degree 2'b10: 180 degree 2'b11: 270 degree
0	RO	0x0	reserved

CRU_GMAC_CON

Address: Operational Base + offset (0x0460)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:6	RO	0x000	reserved
5	RW	0x0	ext_sel_m1 GMAC RMII clock source selection 1. 1'b0: From PLL 1'b1: From GPIO2B7
4	RW	0x0	rmii_mode RMII mode selection. 1'b0: RGMII mode 1'b1: RMII mode
3:2	RW	0x0	mii_txclk_sel MII TX clock selection. 2'b10: clk_gmac_src divided by 50 2'b11: clk_gmac_src divided by 5 Others: clk_gmac_src
1	RW	0x0	rmii_clk_sel RMII clock selection. 1'b0: clk_gmac_src divided by 20 1'b1: clk_gmac_src divided by 2
0	RW	0x0	ext_sel_m0 GMAC RMII clock source selection 0. 1'b0: From PLL 1'b1: From GPIO3C0

CRU_MISC_CON0

Address: Operational Base + offset (0x0464)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14	RW	0x0	dis_pd_sdio_dwn_clk_en Disable pd_sdio_dwn_clk_en, i.e. clocks cannot be gated by PMU. 1'b0: Enable 1'b1: Disable
13	RO	0x0	reserved
12	RW	0x0	dis_pd_crypt_dwn_clk_en Disable pd_crypt_dwn_clk_en, i.e. clocks cannot be gated by PMU. 1'b0: Enable 1'b1: Disable
11	RO	0x0	reserved
10	RW	0x0	dis_pd_usb_dwn_clk_en Disable pd_usb_dwn_clk_en, i.e. clocks cannot be gated by PMU. 1'b0: Enable 1'b1: Disable
9	RW	0x0	dis_pd_nvm_dwn_clk_en Disable pd_nvm_dwn_clk_en, i.e. clocks cannot be gated by PMU. 1'b0: Enable 1'b1: Disable
8	RO	0x0	reserved
7	RW	0x0	dis_pd_npu_dwn_clk_en Disable pd_npu_dwn_clk_en, i.e. clocks cannot be gated by PMU. 1'b0: Enable 1'b1: Disable
6	RW	0x0	dis_pd_vdpu_dwn_clk_en Disable pd_vdpu_dwn_clk_en, i.e. clocks cannot be gated by PMU. 1'b0: Enable 1'b1: Disable
5	RW	0x0	dis_pd_vepu_dwn_clk_en Disable pd_vepu_dwn_clk_en, i.e. clocks cannot be gated by PMU. 1'b0: Enable 1'b1: Disable
4	RW	0x0	dis_pd_ispp_dwn_clk_en Disable pd_ispp_dwn_clk_en, i.e. clocks cannot be gated by PMU. 1'b0: Enable 1'b1: Disable
3	RW	0x0	dis_pd_vo_dwn_clk_en Disable pd_vo_dwn_clk_en, i.e. clocks cannot be gated by PMU. 1'b0: Enable 1'b1: Disable
2	RW	0x0	dis_pd_vi_dwn_clk_en Disable pd_vi_dwn_clk_en, i.e. clocks cannot be gated by PMU. 1'b0: Enable 1'b1: Disable
1	RW	0x0	dis_pd_ddr_dwn_clk_en Disable pd_ddr_dwn_clk_en, i.e. clocks cannot be gated by PMU. 1'b0: Enable 1'b1: Disable

Bit	Attr	Reset Value	Description
0	RW	0x0	dis_pd_core_dwn_clk_en Disable pd_core_dwn_clk_en, i.e. clocks cannot be gated by PMU. 1'b0: Enable 1'b1: Disable

CRU_MISC_CON1

Address: Operational Base + offset (0x0468)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:10	RO	0x00	reserved
9	RW	0x0	corepo_srst_wfien CPU power-on reset enable when CPU is in WFI state. 1'b0: Disable 1'b1: Enable
8	RW	0x0	core_srst_wfien CPU reset enable when CPU is in WFI state. 1'b0: Disable 1'b1: Enable
7:0	RO	0x00	reserved

5.6 Application Notes

5.6.1 PLL Usage

5.6.1.1 PLL Frequency Configuration

FBDIV, POSTDIV1, BYPASS can be configured by programming CRU_xPLL_CON0.
DSMPD, REFDIV, POSTDIV2 can be configured by programming CRU_xPLL_CON1.
FRAC can be configured by programming CRU_xPLL_CON2.(x=A,D,C,H,G)

If DSMPD = 1 (DSM is disabled, "integer mode")
FOUTVCO = (FREF / REFDIV) * FBDIV
FOUTPOSTDIV = FOUTVCO / (POSTDIV1*POSTDIV2)
When FREF is 24MHz, and if 700MHz FOUTPOSTDIV is needed. The configuration can be:

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

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

If DSMPD = 0 (DSM is enabled, "fractional mode")
FOUTVCO = (FREF / REFDIV) * (FBDIV + FRAC / (2^24))
FOUTPOSTDIV = FOUTVCO / (POSTDIV1*POSTDIV2)
When FREF is 24MHz, and if 491.52MHz FOUTPOSTDIV is needed. The configuration can be:

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

And then

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

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

5.6.1.2 PLL Setting Consideration

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

5.6.1.3 PLL Frequency Change And Lock Check

The PLL programming supports changed on-the-fly and the PLL will simply slew to the new frequency.
PLL lock state can be checked in CRU_xPLL_CON1[10] (x=A,D,C,H,G) register. The lock state is high when both original hardware PLL lock and PLL counter lock are high.
The max delay time is 1000*REFDIV/FREF.
PLL locking consists of three phases.

- Phase 1 is control voltage slewing. During this phase one of the clocks (reference or divide) is much faster than the other, and the PLL frequency adjusts almost continuously. When locking from power down, the divide clock is initially very slow and steadily increases frequency. It will take slightly longer for faster VCO settings when locking from power down, since the PLL must slew further.
- Phase 2 is small signal phase acquisition. During this phase, the internal up/down signals alternate semi-chaotically as the phase slowly adjusts until the two signals are aligned. The duration of this phase depends on the loop bandwidth and is faster with higher bandwidth. Bandwidth can be estimated as FREF / REFDIV / 20 for integer mode and FREF /REFDIV / 40 for fractional mode. The duration of small signal locking is about 1/Bandwidth.
- Phase 3 is the digital cycle count. After the last cycle slip is detected, an internal counter waits 256 FREF / REFDIV cycles before the lock signal goes high. This is frequently the dominant factor in lock time – especially for slower reference clock signals or large reference divide settings. This time can be calculated as 256*REFDIV/FREF.

5.6.2 PLL Offset Calibration Usage Options

Multiple usage options are available for offset calibration. All options result in systematic period jitter contribution below noise floor.
Following parameters can be configured by programming CRU_xPLL_CON5 and CRU_xPLL_CON6.
CALBYPASS: CRU_xPLL_CON5[1]

CALEN : CRU_xPLL_CON5[2]
FASTCAL : CRU_xPLL_CON5[3]
CALIN : CRU_xPLL_CON5[15:4]
CALCNT : CRU_xPLL_CON6[2:0]
The status can be monitored in CRU_xPLL_OFFSETCAL_STATUS.
CALOUT : CRU_xPLL_OFFSETCAL_STATUS[11:0]
CALLOCK : CRU_xPLL_OFFSETCAL_STATUS[16]

5.6.2.1 Option 1: CALOUT Permanently Stored

1) Set CALCNT with legal value, set CALEN =1, enable PLL, start calibration.
2) Store CALOUT after CALLOCK goes high, then disable PLL.
3) Preset CALIN to stored value and set CALBYPASS=1 for the remainder of life time. And enable PLL.
4) PLL clock is valid when PLLLOCK goes high.

5.6.2.2 Option 2: CALOUT Not Stored

In many cases, it is impossible or impractical to store the CALOUT value. In such a case, the calibration can be initialized from CALIN=0, and run constantly in the background.

1) It is recommended to run the offset calibration constantly.
2) Set FASTCAL=1. PLL clock is valid after 32 * 2^(5+CALCNT) * PFD cycles.

5.6.3 Divider Usage

CRU supports multi-dividers for different clock requirement.

- Divider free divider
- Fractional divider
- Divfree50 divider
- DivfreeNP5 divider

5.6.3.1 Fractional Divider Usage

To get specific frequency, clocks of I2S, audioPWM, UART, VOP, CIFOUT, MIPICSI can be generated by fractional divider. Generally you should set that denominator 20 times larger than numerator to generate precise clock frequency. So the fractional divider applies only to generate low frequency clock. For implementation issue, the input source clocks of fractional divider should limit to less than 1200MHz, except MIPICSI with the input source clock of fractional divider less than 400MHz.

5.6.3.2 Divfree50 Divider Usage

Some modules like PDM, EMMC, SDIO, SDMMC, FSPI and NANDC need clock of 50% duty cycle, divfree50 can generate clock of 50% duty cycle even in odd value divisor.

5.6.3.3 DivFreeNP5 Divider Usage

Some modules like NPU, ISPP and ISP need some special frequency can use this divider. Frequency of this divider=clk_src/((2*n+1)/2).

5.6.4 Global Software Reset

Two global software resets are designed in RV1109/RV1126, you can program CRU_GLB_SRST_FST_VALUE[15:0] as 0xfdb9 to assert the first global software reset glb_srstn_1 and program CRU_GLB_SRST_SND_VALUE[15:0] as 0xeca8 to assert the second global software reset glb_srstn_2. These two software resets are self-de-asserted by hardware. Resetting hold timing of global software reset (glb_srstn_1, glb_srstn_2, wdt_rstn, tsadc_rstn) can be programed up to 1ms.
glb_srstn_1 resets almost all logic except some registers just supporting hardware reset.
glb_srstn_2 resets almost all logic except GRFs and GPIOs.
Reset for IP in PD_PMU can be hold if its reset_hold_enable in PMUGRF_PMU_SOC_CON1 or PMUGRF_PMU_SOC_CON2 is high even if glb_srstn_1 or glb_srstn_2 active.

5.6.5 SSCG Usage

There are some scenes where SSCG should be enabled. One scene is in communication where a fixed frequency is required. Another scene is a system requiring a clock with low long-term jitter. When SSCG is used, the PLL should be configured to fractional mode firstly for spread spectrum capability.

5.6.5.1 SSCG Use Internal Point Table

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

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

(CRU_XPLL_CON3[3])

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

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

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

The modulation direction is determined by the ssmod_downspread bit.

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

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

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

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

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

Spread spectrum modulator is implemented by repeating as above.

- Setting ssmod_divval (CRU_XPLL_CON3[7:4])

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

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

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

5.6.5.2 SSCG Use External Point Table

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

This enables customization tables in both shape and the number of sample points for the envelope wave form up to 128 data points. The external table of 128 data points can be configured from CRU_SSCGTBL_CON0~CRU_SSCGTBL_CON31.

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

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

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

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

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

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

- Setting ssmod_divval (CRU_XPLL_CON3[7:4])

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

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

Chapter 6 General Register Files (GRF)

6.1 Overview

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

- GRF, used for general system, with base address 0xFE000000.
- PMUGRF, used for always on system, with base address 0xFE020000.
- DDRGRF, used for DDR system, with base address 0xFE030000.

6.2 Function Description

The function of general register file is:

- GPIO IOMUX control
- GPIO PAD pull down and pull up control
- Common system control
- Record the system state

6.3 GRF Register Description

6.3.1 Registers Summary

Name	Offset	Size	Reset Value	Description
GRF_SOC_CON0	0x0000	W	0x00000000	System control register 0
GRF_SOC_CON1	0x0004	W	0x00000000	System control register 1
GRF_SOC_CON2	0x0008	W	0x00000000	System control register 2
GRF_SOC_STATUS0	0x0010	W	0x00380400	System status register 0
GRF_SOC_STATUS1	0x0014	W	0x00000000	System status register 1
GRF_CPU_CON0	0x0020	W	0x00000C0F	CPU control register 0
GRF_CPU_CON1	0x0024	W	0x00000000	CPU control register 1
GRF_CPU_STATUS0	0x0030	W	0x0000C0F0	CPU status register 0
GRF_INTCONNET_CON0	0x0040	W	0x00000000	System internal connection control register 0
GRF_INTCONNET_CON1	0x0044	W	0x00000000	System internal connection control register 1
GRF_INTCONNET_CON2	0x0048	W	0x00000000	System internal connection control register 2
GRF_INTCONNET_CON3	0x004C	W	0x00000000	System internal connection control register 3
GRF_USBHOST_CON0	0x0050	W	0x00000820	USBHOST control register 0
GRF_USBHOST_CON1	0x0054	W	0x000004BC	USBHOST control register 1
GRF_USBHOST_STATUS0	0x0058	W	0x00000000	USBHOST status register 0
GRF_USBOTG_CON0	0x005C	W	0x00002000	USBOTG control register 0
GRF_USBOTG_CON1	0x0060	W	0x00000100	USBOTG control register 1
GRF_USBOTG_STATUS0	0x0064	W	0x00000000	USBOTG status register 0
GRF_USBOTG_STATUS1	0x0068	W	0x00000000	USBOTG status register 1
GRF_USBOTG_STATUS2	0x006C	W	0x000007E8	USBOTG status register 2
GRF_MAC_CON0	0x0070	W	0x00000000	MAC control register 0
GRF_MAC_CON1	0x0074	W	0x00000000	MAC control register 1
GRF_MAC_CON2	0x0078	W	0x00000000	MAC control register 2
GRF_MAC_STATUS0	0x0084	W	0x00000000	MAC status register 0
GRF_MAC_STATUS1	0x0088	W	0x00000000	MAC status register 1
GRF_MAC_STATUS2	0x008C	W	0x00000000	MAC status register 2
GRF_MEM_CON0	0x0090	W	0x00000505	Memory EMA control register 0
GRF_MEM_CON1	0x0094	W	0x00000505	Memory EMA control register 1

Name	Offset	Size	Reset Value	Description
GRF_MEM_CON2	0x0098	W	0x000000A5	Memory EMA control register 2
GRF_MEM_CON3	0x009C	W	0x00002805	Memory EMA control register 3
GRF_TSADC_CON	0x0100	W	0x00000377	TSADC control register
GRF_CHIP_ID	0x0110	W	0x00001109	Chip ID register
GRF_GPIO0C_IOMUX_H	0x10000	W	0x00000000	GPIO0C high bits IOMUX control register
GRF_GPIO0D_IOMUX_L	0x10004	W	0x00000000	GPIO0D low bits IOMUX control register
GRF_GPIO0D_IOMUX_H	0x10008	W	0x00000000	GPIO0D high bits IOMUX control register
GRF_GPIO1A_IOMUX_L	0x10010	W	0x00000000	GPIO1A low bits IOMUX control register
GRF_GPIO1A_IOMUX_H	0x10014	W	0x00000000	GPIO1A high bits IOMUX control register
GRF_GPIO1B_IOMUX_L	0x10018	W	0x00000000	GPIO1B low bits IOMUX control register
GRF_GPIO1B_IOMUX_H	0x1001C	W	0x00000000	GPIO1B high bits IOMUX control register
GRF_GPIO1C_IOMUX_L	0x10020	W	0x00000000	GPIO1C low bits IOMUX control register
GRF_GPIO1C_IOMUX_H	0x10024	W	0x00000000	GPIO1C high bits IOMUX control register
GRF_GPIO1D_IOMUX_L	0x10028	W	0x00000000	GPIO1D low bits IOMUX control register
GRF_GPIO1D_IOMUX_H	0x1002C	W	0x00000000	GPIO1D high bits IOMUX control register
GRF_GPIO2A_IOMUX_L	0x10030	W	0x00000000	GPIO2A low bits IOMUX control register
GRF_GPIO2A_IOMUX_H	0x10034	W	0x00000000	GPIO2A high bits IOMUX control register
GRF_GPIO2B_IOMUX_L	0x10038	W	0x00000000	GPIO2B low bits IOMUX control register
GRF_GPIO2B_IOMUX_H	0x1003C	W	0x00000000	GPIO2B high bits IOMUX control register
GRF_GPIO2C_IOMUX_L	0x10040	W	0x00000000	GPIO2C low bits IOMUX control register
GRF_GPIO2C_IOMUX_H	0x10044	W	0x00000000	GPIO2C high bits IOMUX control register
GRF_GPIO2D_IOMUX_L	0x10048	W	0x00000000	GPIO2D low bits IOMUX control register
GRF_GPIO2D_IOMUX_H	0x1004C	W	0x00000000	GPIO2D high bits IOMUX control register
GRF_GPIO3A_IOMUX_L	0x10050	W	0x00000000	GPIO3A low bits IOMUX control register
GRF_GPIO3A_IOMUX_H	0x10054	W	0x00000000	GPIO3A high bits IOMUX control register
GRF_GPIO3B_IOMUX_L	0x10058	W	0x00000000	GPIO3B low bits IOMUX control register
GRF_GPIO3B_IOMUX_H	0x1005C	W	0x00000000	GPIO3B high bits IOMUX control register
GRF_GPIO3C_IOMUX_L	0x10060	W	0x00000000	GPIO3C low bits IOMUX control register

Name	Offset	Size	Reset Value	Description
GRF_GPIO3C_IOMUX_H	0x10064	W	0x00000000	GPIO3C high bits IOMUX control register
GRF_GPIO3D_IOMUX_L	0x10068	W	0x00000000	GPIO3D low bits IOMUX control register
GRF_GPIO3D_IOMUX_H	0x1006C	W	0x00000000	GPIO3D high bits IOMUX control register
GRF_GPIO4A_IOMUX_L	0x10070	W	0x00000000	GPIO4A low bits IOMUX control register
GRF_GPIO0C_DS_H	0x10080	W	0x00001111	GPIO0C high bits driver strength control register
GRF_GPIO0D_DS_L	0x10084	W	0x00001111	GPIO0D low bits driver strength control register
GRF_GPIO0D_DS_H	0x10088	W	0x00002111	GPIO0D high bits driver strength control register
GRF_GPIO1A_DS_L	0x10090	W	0x00002111	GPIO1A low bits driver strength control register
GRF_GPIO1A_DS_H	0x10094	W	0x00001111	GPIO1A high bits driver strength control register
GRF_GPIO1B_DS_L	0x10098	W	0x00001212	GPIO1B low bits driver strength control register
GRF_GPIO1B_DS_H	0x1009C	W	0x00001111	GPIO1B high bits driver strength control register
GRF_GPIO1C_DS_L	0x100A0	W	0x00001111	GPIO1C low bits driver strength control register
GRF_GPIO1C_DS_H	0x100A4	W	0x00001111	GPIO1C high bits driver strength control register
GRF_GPIO1D_DS_L	0x100A8	W	0x00001111	GPIO1D low bits driver strength control register
GRF_GPIO1D_DS_H	0x100AC	W	0x00001111	GPIO1D high bits driver strength control register
GRF_GPIO2A_DS_L	0x100B0	W	0x00002211	GPIO2A low bits driver strength control register
GRF_GPIO2A_DS_H	0x100B4	W	0x00001111	GPIO2A high bits driver strength control register
GRF_GPIO2B_DS_L	0x100B8	W	0x00001111	GPIO2B low bits driver strength control register
GRF_GPIO2B_DS_H	0x100BC	W	0x00001111	GPIO2B high bits driver strength control register
GRF_GPIO2C_DS_L	0x100C0	W	0x00001111	GPIO2C low bits driver strength control register
GRF_GPIO2C_DS_H	0x100C4	W	0x00001111	GPIO2C high bits driver strength control register
GRF_GPIO2D_DS_L	0x100C8	W	0x00001111	GPIO2D low bits driver strength control register
GRF_GPIO2D_DS_H	0x100CC	W	0x00002111	GPIO2D high bits driver strength control register
GRF_GPIO3A_DS_L	0x100D0	W	0x00001111	GPIO3A low bits driver strength control register
GRF_GPIO3A_DS_H	0x100D4	W	0x00001111	GPIO3A high bits driver strength control register
GRF_GPIO3B_DS_L	0x100D8	W	0x00001111	GPIO3B low bits driver strength control register

Name	Offset	Size	Reset Value	Description
GRF_GPIO3B_DS_H	0x100DC	W	0x00001111	GPIO3B high bits driver strength control register
GRF_GPIO3C_DS_L	0x100E0	W	0x00001111	GPIO3C low bits driver strength control register
GRF_GPIO3C_DS_H	0x100E4	W	0x00001111	GPIO3C high bits driver strength control register
GRF_GPIO3D_DS_L	0x100E8	W	0x00001111	GPIO3D low bits driver strength control register
GRF_GPIO3D_DS_H	0x100EC	W	0x00001111	GPIO3D high bits driver strength control register
GRF_GPIO4A_DS_L	0x100F0	W	0x00000011	GPIO4A low bits driver strength control register
GRF_GPIO0C_P_H	0x10100	W	0x000055AA	GPIO0C high bits PU/PD control register
GRF_GPIO0D_P	0x10104	W	0x0000A555	GPIO0D PU/PD control register
GRF_GPIO1A_P	0x10108	W	0x00005596	GPIO1A PU/PD control register
GRF_GPIO1B_P	0x1010C	W	0x00005565	GPIO1B PU/PD control register
GRF_GPIO1C_P	0x10110	W	0x0000AA55	GPIO1C PU/PD control register
GRF_GPIO1D_P	0x10114	W	0x0000AA5A	GPIO1D PU/PD control register
GRF_GPIO2A_P	0x10118	W	0x0000AAAA	GPIO2A PU/PD control register
GRF_GPIO2B_P	0x1011C	W	0x0000AAAA	GPIO2B PU/PD control register
GRF_GPIO2C_P	0x10120	W	0x0000AAAA	GPIO2C PU/PD control register
GRF_GPIO2D_P	0x10124	W	0x0000AAAA	GPIO2D PU/PD control register
GRF_GPIO3A_P	0x10128	W	0x0000AA55	GPIO3A PU/PD control register
GRF_GPIO3B_P	0x1012C	W	0x0000AAAA	GPIO3B PU/PD control register
GRF_GPIO3C_P	0x10130	W	0x0000AAAA	GPIO3C PU/PD control register
GRF_GPIO3D_P	0x10134	W	0x0000AAAA	GPIO3D PU/PD control register
GRF_GPIO4A_P	0x10138	W	0x0000000A	GPIO4A PU/PD control register
GRF_GPIO0C_IE_H	0x10140	W	0x000000FF	GPIO0C high bits IE control register
GRF_GPIO0D_IE	0x10144	W	0x000000FF	GPIO0D IE control register
GRF_GPIO1A_IE	0x10148	W	0x000000FF	GPIO1A IE control register
GRF_GPIO1B_IE	0x1014C	W	0x000000FF	GPIO1B IE control register
GRF_GPIO1C_IE	0x10150	W	0x000000FF	GPIO1C IE control register
GRF_GPIO1D_IE	0x10154	W	0x000000FF	GPIO1D IE control register
GRF_GPIO2A_IE	0x10158	W	0x000000FF	GPIO2A IE control register
GRF_GPIO2B_IE	0x1015C	W	0x000000FF	GPIO2B IE control register
GRF_GPIO2C_IE	0x10160	W	0x000000FF	GPIO2C IE control register
GRF_GPIO2D_IE	0x10164	W	0x000000FF	GPIO2D IE control register
GRF_GPIO3A_IE	0x10168	W	0x000000FF	GPIO3A IE control register
GRF_GPIO3B_IE	0x1016C	W	0x000000FF	GPIO3B IE control register
GRF_GPIO3C_IE	0x10170	W	0x000000FF	GPIO3C IE control register
GRF_GPIO3D_IE	0x10174	W	0x000000FF	GPIO3D IE control register
GRF_GPIO4A_IE	0x10178	W	0x000000FF	GPIO4A IE control register
GRF_GPIO0C_SMT_H	0x10180	W	0x00000000	GPIO0C high bits SMT control register
GRF_GPIO0D_SMT	0x10184	W	0x00000000	GPIO0D SMT control register
GRF_GPIO1A_SMT	0x10188	W	0x00000000	GPIO1A SMT control register
GRF_GPIO1B_SMT	0x1018C	W	0x00000000	GPIO1B SMT control register
GRF_GPIO1C_SMT	0x10190	W	0x00000000	GPIO1C SMT control register
GRF_GPIO1D_SMT	0x10194	W	0x00000000	GPIO1D SMT control register
GRF_GPIO2A_SMT	0x10198	W	0x00000000	GPIO2A SMT control register

Name	Offset	Size	Reset Value	Description
GRF_GPIO2B_SMT	0x1019C	W	0x00000000	GPIO2B SMT control register
GRF_GPIO2C_SMT	0x101A0	W	0x00000000	GPIO2C SMT control register
GRF_GPIO2D_SMT	0x101A4	W	0x00000000	GPIO2D SMT control register
GRF_GPIO3A_SMT	0x101A8	W	0x00000000	GPIO3A SMT control register
GRF_GPIO3B_SMT	0x101AC	W	0x00000000	GPIO3B SMT control register
GRF_GPIO3C_SMT	0x101B0	W	0x00000000	GPIO3C SMT control register
GRF_GPIO3D_SMT	0x101B4	W	0x00000000	GPIO3D SMT control register
GRF_GPIO4A_SMT	0x101B8	W	0x00000000	GPIO4A SMT control register
GRF_CSIPHY0_CON	0x10200	W	0x00000000	CSIPHY0 control register
GRF_CSIPHY0_STATUS	0x10208	W	0x00000000	CSIPHY0 status register
GRF_CSIPHY1_CON	0x10210	W	0x00000000	CSIPHY1 control register
GRF_CSIPHY1_STATUS	0x10218	W	0x00000000	CSIPHY1 status register
GRF_DSIPHY_CON	0x10220	W	0x00000000	DSIPHY control register
GRF_USBPHY_CON0	0x10230	W	0x00008C52	USBPHY control register 0
GRF_USBPHY_CON1	0x10234	W	0x00000080	USBPHY control register 1
GRF_USBPHY_CON2	0x10238	W	0x00000D50	USBPHY control register 2
GRF_USBPHY_STATUS	0x10248	W	0x20082048	USBPHY status register
GRF_CIFIO_CON	0x10250	W	0x00000000	CIF IO control register
GRF_SD_JTAG_SWITCH_DLY_CNT	0x10254	W	0x00000000	JTAG switch delay counter register
GRF_UART2RX_LOW_CON	0x10258	W	0x00000000	UART2 RX pin filter control register
GRF_IOFUNC_CON0	0x10260	W	0x00000000	IO function control register 0
GRF_IOFUNC_CON1	0x10264	W	0x00000000	IO function control register 1
GRF_IOFUNC_CON2	0x10268	W	0x00000000	IO function control register 2
GRF_IOFUNC_CON3	0x1026C	W	0x00000010	IO function control register 3
GRF_USBPHY0_CFG_CON	0x10270	W	0x00000004	USBPHY0 APB configuration control register
GRF_USBPHY0_CFG_ADDRIN	0x10274	W	0x00000000	USBPHY0 APB configuration address register
GRF_USBPHY0_CFG_ADDROUT	0x10278	W	0x00000000	USBPHY0 APB configuration address status register
GRF_USBPHY0_CFG_DLY_CON	0x1027C	W	0x00000000	USBPHY0 APB configuration interface delay control register
GRF_USBPHY1_CFG_CON	0x10280	W	0x00000004	USBPHY1 APB configuration control register
GRF_USBPHY1_CFG_ADDRIN	0x10284	W	0x00000000	USBPHY1 APB configuration address register
GRF_USBPHY1_CFG_ADDROUT	0x10288	W	0x00000000	USBPHY1 APB configuration address status register
GRF_USBPHY1_CFG_DLY_CON	0x1028C	W	0x00000000	USBPHY1 APB configuration interface delay control register
GRF_USB_SIG_DETECT_CON	0x10300	W	0x00000000	USB detection control register
GRF_USB_SIG_DETECT_STATUS	0x10304	W	0x00000000	USB detection status register
GRF_USB_SIG_DETECT_CLR	0x10308	W	0x00000000	USB detection clear register
GRF_USB_LINESTATE_CON	0x10310	W	0x00030100	USB linestate control register
GRF_USB_DISCONNECT_CON	0x10314	W	0x00030100	USB disconnect control register
GRF_USB_BVALID_CON	0x10318	W	0x00030100	USBPHY bvalid control register

Name	Offset	Size	Reset Value	Description
GRF_USB_ID_CON	0x1031C	W	0x00030100	USBPHY id control register

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

6.3.2 Detail Register Description

GRF_SOC_CON0

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RO	0x0	reserved
11	RW	0x0	clk_hpll_ls_sel HPLL clock path selection. 1'b0: Normal 1'b1: Through level-shift cell
10	RW	0x0	clk_cpll_ls_sel CPLL clock path selection. 1'b0: Normal 1'b1: Through level-shift cell
9	RW	0x0	clk_dpll_ls_sel DPLL clock path selection. 1'b0: Normal 1'b1: Through level-shift cell
8	RW	0x0	clk_apll_ls_sel APLL clock path selection. 1'b0: Normal 1'b1: Through level-shift cell
7:6	RO	0x0	reserved
5	RW	0x0	npu_csysreq_sel NPU csysreq source selection. 1'b0: From software (GRF_SOC_CON0[4]) 1'b1: From hardware (npu_pwr_IdleReq)
4	RW	0x0	npu_csysreq NPU system low-power request(csysreq). 1'b0: Inactive 1'b1: Active
3	RW	0x0	mcu_soft_irq MCU software irq. 1'b0: Inactive 1'b1: Active
2	RW	0x0	usbphy_hostport_wakeup_irq_en USBPHY host port wakeup irq enable. 1'b0: Disable 1'b1: Enable
1	RW	0x0	npu_disable_ram_clockgating Disable NPU clock gating of the rams during BIST. 1'b0: Not disable 1'b1: Disable
0	RW	0x0	wdtns_glb_reset_en Non-secure WDT global reset enable. 1'b0: Disable 1'b1: Enable

GRF_SOC_CON1

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RO	0x0	reserved
11	RW	0x0	uart5_cts_inv uart5_cts polarity selection. 1'b0: Low active 1'b1: High active
10	RW	0x0	uart5_rts_inv uart5_rts polarity selection. 1'b0: Low active 1'b1: High active
9	RW	0x0	uart4_cts_inv uart4_cts polarity selection. 1'b0: Low active 1'b1: High active
8	RW	0x0	uart4_rts_inv uart4_rts polarity selection. 1'b0: Low active 1'b1: High active
7	RW	0x0	uart3_cts_inv uart3_cts polarity selection. 1'b0: Low active 1'b1: High active
6	RW	0x0	uart3_rts_inv uart3_rts polarity selection. 1'b0: Low active 1'b1: High active
5	RW	0x0	uart2_cts_inv uart2_cts polarity selection. 1'b0: Low active 1'b1: High active
4	RW	0x0	uart2_rts_inv uart2_rts polarity selection. 1'b0: Low active 1'b1: High active
3:2	RO	0x0	reserved
1	RW	0x0	uart0_cts_inv uart0_cts polarity selection. 1'b0: Low active 1'b1: High active
0	RW	0x0	uart0_rts_inv uart0_rts polarity selection. 1'b0: Low active 1'b1: High active

GRF_SOC_CON2

Address: Operational Base + offset (0x0008)

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

Bit	Attr	Reset Value	Description
15	RW	0x0	mcu_jtag_sel MCU jtag selection. 1'b0: MCU jtag 1'b1: Reserved This bit should be set to 0 for MCU jtag working.
14	RW	0x0	acdcdig_i2c_trans_req ACODEC I2C transfer request. 1'b0: Not request 1'b1: Request
13	RW	0x0	acdcdig_i2s_active I2S0 input source selection. 1'b0: From external ACODEC 1'b1: From internal ACODEC
12	RW	0x0	vopm_dma_finish_en vop_dma_finish enable for dma_finish generation. 1'b0: Disable 1'b1: Enable
11:8	RO	0x0	reserved
7	RW	0x0	vdec_fuse_vp9 Need tie to 0 for VDEC normal work
6	RO	0x0	reserved
5	RW	0x0	dsi0_dpiupdatecfg DSIHOST dsi0_dpiupdatecfg configuration
4	RW	0x0	dsi0_dpishutdn DSIHOST dsi0_dpishutdn configuration
3	RW	0x0	dsi0_dpicolorm DSIHOST dsi0_dpicolorm configuration
2	RW	0x0	vop_flow_switch 1'b0: VOP access DDR through VPU memory scheduler port 1'b1: VOP access DDR through VI memory scheduler port
1:0	RW	0x0	vop_press QoS control for VOP

GRF_SOC_STATUS0

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:24	RO	0x00	npu_debug_out NPU debug output
23:22	RO	0x0	reserved
21	RO	0x1	nandc_master_idle NANDC master idle status. 1'b0: Not idle 1'b1: Idle
20	RO	0x1	pmu_pwr_IdleAck PMU idle acknowledge status. 1'b0: Not acknowledge 1'b1: Acknowledge
19	RO	0x1	pmu_pwr_Idle PMU idle status. 1'b0: Not idle 1'b1: Idle
18	RO	0x0	acdcdig_i2c_trans_ack ACODEC I2C transfer acknowledge. 1'b0: Not acknowledge 1'b1: Acknowledge

Bit	Attr	Reset Value	Description
17	RO	0x0	vopm_dma_finish VOP dma finish status. 1'b0: Not finish 1'b1: Finish
16	RO	0x0	timer5_en_status Timer5 enable status. 1'b0: Disable 1'b1: Enable
15	RO	0x0	timer4_en_status Timer4 enable status. 1'b0: Disable 1'b1: Enable
14	RO	0x0	timer3_en_status Timer3 enable status. 1'b0: Disable 1'b1: Enable
13	RO	0x0	timer2_en_status Timer2 enable status. 1'b0: Disable 1'b1: Enable
12	RO	0x0	timer1_en_status Timer1 enable status. 1'b0: Disable 1'b1: Enable
11	RO	0x0	timer0_en_status Timer0 enable status. 1'b0: Disable 1'b1: Enable
10	RO	0x1	iep_core_idle IEP core idle status. 1'b0: Not idle 1'b0: Idle
9	RO	0x0	reserved
8	RO	0x0	scramble_shift_ready Scramble shift status. 1'b0: Not ready 1'b1: Ready
7	RO	0x0	mcu_rst_n_out MCU reset status. 1'b0: MCU in reset state 1'b1: MCU in normal state
6	RO	0x0	ddrphy_plllock DDRPHY PLL lock status. 1'b0: Not locked 1'b1: Locked
5	RO	0x0	reserved
4	RO	0x0	hpll_lock HPLL lock status. 1'b0: Not locked 1'b1: Locked
3	RO	0x0	gpll_lock GPLL lock status. 1'b0: Not locked 1'b1: Locked

Bit	Attr	Reset Value	Description
2	RO	0x0	cpll_lock CPLL lock status. 1'b0: Not locked 1'b1: Locked
1	RO	0x0	dpll_lock DPLL lock status. 1'b0: Not locked 1'b1: Locked
0	RO	0x0	apll_lock APLL lock status. 1'b0: Not locked 1'b1: Locked

GRF_SOC_STATUS1

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:5	RO	0x0000000	reserved
4	RO	0x0	pmu_global_int_disable3 CPU3 interrupt disable by PMU. 1'b0: Enable 1'b1: Disable
3	RO	0x0	pmu_global_int_disable2 CPU2 interrupt disable by PMU. 1'b0: Enable 1'b1: Disable
2	RO	0x0	pmu_global_int_disable1 CPU1 interrupt disable by PMU. 1'b0: Enable 1'b1: Disable
1	RO	0x0	pmu_global_int_disable0 CPU0 interrupt disable by PMU. 1'b0: Enable 1'b1: Disable
0	RO	0x0	pmu_global_int_disable SCU interrupt disable by PMU. 1'b0: Enable 1'b1: Disable

GRF_CPU_CON0

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13	RW	0x0	cpu_ema_detect_en CPU EMA detection enable. If enabled, hardware will set CPU bus idle and stop CPU clock until finish EMA modification. 1'b0: Disable 1'b1: Enable
12	RW	0x0	evento_clear EVENTO clear. 1'b0: Not clear 1'b1: Clear

Bit	Attr	Reset Value	Description
11	RW	0x1	dbgselfaddrv Debug self-address offset valid. 1'b0: Not valid 1'b1: Valid
10	RW	0x1	dbgromaddrv Debug ROM physical address valid. 1'b0: Not valid 1'b1: Valid
9	RW	0x0	eventi Event input for processor wake-up from WFE state. When this signal is asserted, it acts as a WFE wake-up event to all the processors in the multiprocessor device.
8	RW	0x0	l2rstdisable Disable automatic L2 cache invalidated at reset. 1'b0: Enable L2 cache invalidated at reset 1'b1: Disable L2 cache invalidated at reset (soft reset) Only change this signal when the processor is in the reset state.
7:4	RW	0x0	l1rstdisable Disable automatic data cache, instruction cache and TLB invalidated at reset for each processor. Each bit represents each processor respectively. 1'b0: Enable data cache invalidated, instruction cache invalidated and TLB invalidated at reset. 1'b1: Disable data cache invalidated, instruction cache invalidated, and TLB invalidated at reset (soft reset). Only change these signals when the processor is in the reset state.
3:0	RW	0xf	vinithi Set the base address of the vector table for each processor. Each bit represents each processor respectively. 1'b0: Vector table base address of 0x00000000 1'b1: Vector table base address of 0xFFFF0000 Only change this signal when the processor is in the reset state.

GRF_CPU_CON1

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7:4	RW	0x0	cfgte Controls processor state for exception handling (TE bit) at reset for each processor. Each bit represents each processor respectively. 1'b0: ARM instruction set for exception handling 1'b1: Thumb instruction set for exception handling Only change this signal when the processor is in the reset state.

Bit	Attr	Reset Value	Description
3:0	RW	0x0	cfgend Control endianness during data exception handling (EE-bit) at reset for each processor. Each bit represents each processor respectively. 1'b0: Little-endian data during exception handling 1'b1: Big-endian data during exception handling Only change this signal when the processor is in the reset state.

GRF_CPU_STATUS0

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15	RO	0x1	jtagnsw JTAGNSW output
14	RO	0x1	jtagtop JTAGTOP output
13	WO	0x0	evento_rising_edge Event output rising edge indication. 1'b0: Inactive 1'b1: Active
12	RO	0x0	standbywfil2 Indicates if the L2 memory system is in WFI standby mode. 1'b0: Normal state 1'b1: WFI state
11:8	RO	0x0	smpnamp Signals Symmetric MultiProcessing (SMP) mode or Asymmetric MultiProcessing (AMP) for each processor. Each bit represents each processor respectively.
7:4	RO	0xf	standbywfi Indicates if a processor is in WFI state for each processor. 1'b0: Normal state 1'b1: WFI state Each bit represents each processor respectively.
3:0	RO	0x0	standbywfe Indicates if a processor is in WFE state for each processor. Each bit represents each processor respectively. 1'b0: Normal state 1'b1: WFE state

GRF_INTCONNET_CON0

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:6	RO	0x000	reserved
5:4	RW	0x0	press_vdec QoS control for VDEC
3:2	RO	0x0	reserved

Bit	Attr	Reset Value	Description
1:0	RW	0x0	split Decide the splitting size of the interconnection for a transaction from a master. 2'b00: Splitting size is 64 bytes 2'b01: Splitting size is 32 bytes 2'b10: Splitting size is 16 bytes 2'b11: Reserved. Do not set to this value; otherwise, the system will crash.

GRF_INTCONNET_CON1

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	vi_bus_to_ddr_stall Response type when BIU_BUS or BIU_DDR in idle state and ISPP/ISP/CIF/IEP access DDR. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the BIU idle state is de-asserted
14	RW	0x0	bus_to_ddr_stall Response type when BIU_BUS or BIU_DDR in idle state and masters in BUS or in PHP or NPU access DDR. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the BIU idle state is de-asserted
13	RW	0x0	mcu_to_bushold_stall Response type when BIU_BUS or BIU_BUSHOLD in idle state and MCU access DDR. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the BIU idle state is de-asserted
12	RW	0x0	phpmid_to_bus_stall Response type when BIU_PHPMID or BIU_BUS in idle state and SDMMC/SDIO/EMMC/FSPI/NANDC/USBHOST/USBOTG/GMAC access DDR or SYSTEM_SRAM. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the BIU idle state is de-asserted
11	RW	0x0	phpmid_to_usb_stall Response type when BIU_PHPMID or BIU_USB in idle state and CPU or MCU access USBOTG/USBHOST register. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the BIU idle state is de-asserted
10	RW	0x0	phpmid_to_nvm_stall Response type when BIU_PHPMID or BIU_NVM in idle state and CPU or MCU access FSPI/EMMC/NANDC register. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the BIU idle state is de-asserted
9	RW	0x0	phpmid_to_gmac_stall Response type when BIU_PHPMID or BIU_GMAC in idle state and CPU or MCU access GMAC register. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the BIU idle state is de-asserted

Bit	Attr	Reset Value	Description
8	RW	0x0	usb_to_phpmid_stall Response type when BIU_USB or BIU_PHPMID in idle state and USBHOST access DDR or USBOTG access SYSTEM_SRAM or DDR. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the BIU idle state is de-asserted
7	RW	0x0	gmac_to_phpmid_stall Response type when BIU_GMAC or BIU_PHPMID in idle state and GMAC access DDR. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the BIU idle state is de-asserted
6	RW	0x0	nvm_to_phpmid_stall Response type when BIU_NVM or BIU_PHPMID in idle state and FSPI/EMMC/NANDC access DDR. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the BIU idle state is de-asserted
5	RW	0x0	vepu_to_ddr_stall Response type when BIU_VEPU or BIU_DDR in idle state and VEPU access DDR. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the BIU idle state is de-asserted
4	RW	0x0	vdpu_to_ddr_stall Response type when BIU_VDPU or BIU_DDR in idle state and VDEC/JPEG access DDR. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the BIU idle state is de-asserted
3	RW	0x0	ispp_to_bus_stall Response type when BIU_ISPP or BIU_BUS in idle state and ISPP access DDR. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the BIU idle state is de-asserted
2	RW	0x0	vo_to_bus_stall Response type when BIU_VO or BIU_BUS in idle state and RGA/IEP/VOP access DDR. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the BIU idle state is de-asserted
1	RW	0x0	cpu_to_bus_stall Response type when BIU_CORE or BIU_BUS in idle state and CPU access other slave or SYSTEM_SRAM. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the BIU idle state is de-asserted
0	RW	0x0	cpu_to_ddr_stall Response type when BIU_CORE or BIU_DDR in idle state and CPU access DDR. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the BIU idle state is de-asserted

GRF_INTCONNET_CON2

Address: Operational Base + offset (0x0048)

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

Bit	Attr	Reset Value	Description
15	RW	0x0	bus_to_vi_stall Response type when BIU_BUS or BIU_VI in idle state and CPU/MCU access ISP/CIF. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the BIU idle state is de-asserted
14	RW	0x0	bus_to_vepu_stall Response type when BIU_BUS or BIU_VEPU in idle state and CPU/MCU access VENC register. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the BIU idle state is de-asserted
13	RW	0x0	bus_to_vdpu_stall Response type when BIU_BUS or BIU_VDPU in idle state and CPU/MCU access VDEC/JPEG register. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the BIU idle state is de-asserted
12	RW	0x0	bus_to_ispp_stall Response type when BIU_BUS or BIU_ISPP in idle state and CPU/MCU access ISPP register. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the BIU idle state is de-asserted
11	RW	0x0	npu_to_bus_stall Response type when BIU_NPU or BIU_BUS in idle state and NPU access DDR. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the BIU idle state is de-asserted
10	RW	0x0	bus_to_top_stall Response type when BIU_BUS or BIU_TOP in idle state and CPU/MCU access all PHYs/CRU register. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the BIU idle state is de-asserted
9	RW	0x0	vi_to_bus_stall Response type when BIU_VI or BIU_BUS in idle state and ISPP/ISP/CIF/IEP access DDR. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the BIU idle state is de-asserted
8	RW	0x0	bus_to_phpmid_stall Response type when BIU_BUS or BIU_PHPMID in idle state and CPU/MCU access SDMMC/SDIO/EMMC/FSPI/NANDC/USBHOST/USBOTG/GMAC. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the BIU idle state is de-asserted
7	RW	0x0	bus_to_npu_stall Response type when BIU_BUS or BIU_NPU in idle state and CPU/MCU access NPU register. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the BIU idle state is de-asserted
6	RW	0x0	hold_to_bus_stall Response type when BIU_BUSHOLD or BIU_DDR in idle state and MCU/DMAC/CRYPTO access DDR. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the BIU idle state is de-asserted

Bit	Attr	Reset Value	Description
5	RW	0x0	busmst_to_hold_stall Response type when BIU_BUS or BIU_BUSHOLD in idle state and DMAC/CRYPTO access DDR. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the BIU idle state is de-asserted
4	RW	0x0	bus_to_pmu_stall Response type when BIU_BUS or BIU_PMU in idle state and CPU/MCU access IPs in PMU. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the BIU idle state is de-asserted
3	RW	0x0	bus_to_ddrc_stall Response type when BIU_BUS or BIU_DDR in idle state and CPU/MCU access DDR controller register. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the BIU idle state is de-asserted
2	RW	0x0	bus_to_audio_stall Response type when BIU_BUS or BIU_AUDIO in idle state and CPU/MCU access I2S/PDM/audPWM/ACDC_DIG register. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the BIU idle state is de-asserted
1	RW	0x0	bus_to_mschreg_stall Response type when BIU_BUS or BIU_DDR in idle state and CPU/MCU access firewall or scheduler register. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the BIU idle state is de-asserted
0	RW	0x0	vo_bus_to_ddr_stall Response type when BIU_BUS or BIU_DDR in idle state and RGA/VOP/IEP access DDR. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the BIU idle state is de-asserted

GRF_INTCONNET_CON3

Address: Operational Base + offset (0x004C)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x0	bus_to_cpu_stall bus_to_cpu_stall Response type when BIU_BUS or BIU_CORE in idle state and MCU access debug/PVTM register. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the BIU idle state is de-asserted
6	RW	0x0	sdio_to_phpmid_stall Response type when BIU_SDIO or BIU_PHPMID in idle state and SDIO access DDR. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the BIU idle state is de-asserted
5	RW	0x0	sdcard_to_phpmid_stall Response type when BIU_SDCARD or BIU_PHPMID in idle state and SDMMC access DDR. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the BIU idle state is de-asserted

Bit	Attr	Reset Value	Description
4	RW	0x0	phpmid_to_sdio_stall Response type when BIU_PHPMID or BIU_SDIO in idle state and CPU/MCU access SDIO. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the BIU idle state is de-asserted
3	RW	0x0	phpmid_to_sdcard_stall Response type when BIU_PHPMID or BIU_SDCARD in idle state and CPU/MCU access SDMMC. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the BIU idle state is de-asserted
2	RW	0x0	bus_to_crypto_stall Response type when BIU_BUS or BIU_CRYPT in idle state and CPU/ MCU access CRYPTO. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the BIU idle state is de-asserted
1	RW	0x0	crypto_to_bus_stall Response type when BIU_CRYPT or BIU_BUS in idle state and CRYPTO access DDR or SYSTEM_SRAM. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the BIU idle state is de-asserted
0	RW	0x0	bus_to_vo_stall Response type when BIU_BUS or BIU_VO in idle state and CPU/MCU access VOP/RGA/IEP register. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the BIU idle state is de-asserted

GRF_USBHOST_CON0

Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RO	0x0	reserved
11:6	RW	0x20	usbhost_flasj_val_common USBHOST fladj_val_common bit control. 1'b0: Disable 1'b1: Enable
5:0	RW	0x20	usbhost_fladj_val USBHOST fladj bit control. 1'b0: Disable 1'b1: Enable

GRF_USBHOST_CON1

Address: Operational Base + offset (0x0054)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13	RW	0x0	usbhost_arb_pause USBHOST ehci/ohci arbiter pause control. 1'b0: Disable 1'b1: Enable

Bit	Attr	Reset Value	Description
12	RW	0x0	usbhost_ohci_susp_lgcy USBHOST ohci_susp_lgcy bit control. 1'b0: Disable 1'b1: Enable
11	RW	0x0	usbhost_ohci_cntsel USBHOST ohci_cntsel bit control. 1'b0: Disable 1'b1: Enable
10	RW	0x1	usbhost_ohci_clkcktrst USBHOST ohci_clkcktrst bit control. 1'b0: Disable 1'b1: Enable
9	RW	0x0	usbhost_app_prt_ovrcur USBHOST app_prt_ovrcur bit control. 1'b0: Disable 1'b1: Enable
8	RW	0x0	usbhost_autopd_on_overcur_en USBHOST autoppd_on_overcur_en bit control. 1'b0: Disable 1'b1: Enable
7	RW	0x1	usbhost_word_if USBHOST word_if bit control. 1'b0: Disable 1'b1: Enable
6	RW	0x0	usbhost_sim_mode USBHOST sim_mode bit control. 1'b0: Disable 1'b1: Enable
5	RW	0x1	usbhost_incrx_en USBHOST incrx_en bit control. 1'b0: Disable 1'b1: Enable
4	RW	0x1	usbhost_incr8_en USBHOST incr8_en bit control. 1'b0: Disable 1'b1: Enable
3	RW	0x1	usbhost_incr4_en USBHOST incr4_en bit control. 1'b0: Disable 1'b1: Enable
2	RW	0x1	usbhost_incr16_en USBHOST incr16_en bit control. 1'b0: Disable 1'b1: Enable
1	RW	0x0	usbhost_hubsetup_min USBHOST bubsetup_min bit control. 1'b0: Disable 1'b1: Enable
0	RW	0x0	usbhost_app_start_clk USBHOST app_start_clk bit control. 1'b0: Disable 1'b1: Enable

GRF_USBHOST_STATUS0

Address: Operational Base + offset (0x0058)

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30	RW	0x0	usbhost_ehci_power_state_ack USBHOST ehci_power_state_ack status
29	RW	0x0	usbhost_ehci_pme_status USBHOST ehci_pme_status status
28	RW	0x0	usbhost_ehci_bufacc USBHOST ehci_bufacc status
27	RW	0x0	usbhost_ehci_xfer_prdc USBHOST ehci_xfer_prdc status
26	RW	0x0	usbhost_ohci_ccs USBHOST ohci_ccs status
25	RW	0x0	usbhost_ohci_rwe USBHOST ohci_rwe status
24	RW	0x0	usbhost_ohci_drwe USBHOST ohci_drwe status
23	RW	0x0	usbhost_ohci_globalsuspend USBHOST ohci_globalsuspend status
22	RW	0x0	usbhost_ohci_bufacc USBHOST ohci_bufacc status
21	RW	0x0	usbhost_ohci_rmtwkp USBHOST ohci_rmtwkp status
20:17	RW	0x0	usbhost_ehci_lpsmc_state USBHOST ehci_lpsmc_state status
16:11	RW	0x00	usbhost_ehci_usbsts USBHOST ehci_usbsts status
10:0	RW	0x000	usbhost_ehci_xfer_cnt USBHOST ehci_xfer_cnt status

GRF_USBOTG_CON0

Address: Operational Base + offset (0x005C)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	usbotg_host_u2_port_disable USB2.0 port disable control. 1'b0: Port enabled 1'b1: Port disabled, stops reporting connect/disconnect events the port and keeps the port in disabled state.
14	RW	0x0	usbotg_host_port_power_control_present This indicates whether the host controller implementation includes port power control. 1'b0: Indicates that the port does not have port power switches 1'b1: Indicates that the port has port power switches
13:8	RW	0x20	usbotg_fladj_30mhz_reg USB OTG fladj_30mhz_reg configuration
7	RO	0x0	reserved
6	RW	0x0	usbotg_hub_port_perm_attach Indicates if the device attached to a downstream port is permanently attached or not. 1'b0: Not permanently attached 1'b1: Permanently attached
5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4	RW	0x0	usbotg_hub_port_overcurrent Overcurrent indication of the root-hub ports. 1'b0: No Overcurrent 1'b1: Overcurrent
3:0	RW	0x0	usbotg_bus_filter_bypass The function of each bit is: usbotg_bus_filter_bypass[3]: Bypass the filter for utmiotg_iddig usbotg_bus_filter_bypass[2]: Bypass the filters for utmisrp_bvalid and utmisrp_sessend usbotg_bus_filter_bypass[1]: Bypass the filter for pipe3_PowerPresent all U3 ports usbotg_bus_filter_bypass[0]: Bypass the filter for utmiotg_vbusvalid all U2 ports In non-OTG Host-only mode, internal bus filters are not needed. Values: 1'b0: Bus filter(s) enabled 1'b1: Bus filter(s) disabled It is expected that this signal is set or reset at power-on reset and is not changed during the normal operation of the core.

GRF_USBOTG_CON1

Address: Operational Base + offset (0x0060)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RO	0x0	reserved
11:8	RW	0x1	usbotg_host_num_u2_port xHCI host USB2 Port number, default as 1.
7:6	RO	0x0	reserved
5	RW	0x0	usbotg_host_legacy_smi_bar Use this register to support SMI on BAR defined in xHCI spec. SW must set this register, then clear this register to indicate Base Address Register written.
4	RW	0x0	usbotg_host_legacy_smi_pci_cmd Use this register to support SMI on PCI Command defined in xHCI spec. SW must set this register, then clear this register to indicate PCI command register written.
3:2	RO	0x0	reserved
1	RW	0x0	usbotg_pme_en Enable the core to assert pme_generation. 1'b0: Disable 1'b1: Enable
0	RO	0x0	reserved

GRF_USBOTG_STATUS0

Address: Operational Base + offset (0x0064)

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

GRF_USBOTG_STATUS1

Address: Operational Base + offset (0x0068)

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

GRF_USBOTG_STATUS2

Address: Operational Base + offset (0x006C)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:0	RO	0x00007e8	usbotg_host_current_belt usbotg_host_current_belt bit[11:0] status

GRF_MAC_CON0

Address: Operational Base + offset (0x0070)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:9	RO	0x00	reserved
8	RW	0x0	mac_ptp_aux_ts_trig MAC auxiliary timestamp trigger. 1'b0: Disable 1'b1: Enable
7	RW	0x0	mac_sbd_flowctrl MAC sideband flow control. 1'b0: Disable 1'b1: Enable
6:4	RW	0x0	mac_phy_intf_sel MAC PHY interface selection. 3'b000: GMII ot MII 3'b001: RGMII 3'b100: RMII Others: Reserved
3	RW	0x0	mac_m1_rxclk_dly_ena MAC receive clock delay enable for iomux group 1. 1'b0: Disable 1'b1: Enable
2	RW	0x0	mac_m1_txclk_dly_ena MAC transmit clock delay enable for iomux group 1. 1'b0: Disable 1'b1: Enable
1	RW	0x0	mac_m0_rxclk_dly_ena MAC receive clock delay enable for iomux group 0. 1'b0: Disable 1'b1: Enable
0	RW	0x0	mac_m0_txclk_dly_ena MAC transmit clock delay enable for iomux group 0. 1'b0: Disable 1'b1: Enable

GRF_MAC_CON1

Address: Operational Base + offset (0x0074)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RW	0x00	mac_m0_clk_rx_dl_cfg MAC recept clock delay length configuration for iomux group 0
7:0	RW	0x00	mac_m0_clk_tx_dl_cfg GMAC transmit clock delay length configuration for iomux group 0

GRF_MAC_CON2

Address: Operational Base + offset (0x0078)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RW	0x00	mac_m1_clk_rx_dl_cfg MAC receive clock delay length configuration for iomux group 1
7:0	RW	0x00	mac_m1_clk_tx_dl_cfg MAC transmit clock delay length configuration for iomux group 1

GRF_MAC_STATUS0

Address: Operational Base + offset (0x0084)

Bit	Attr	Reset Value	Description
31:4	RO	0x0000000	reserved
3	RO	0x0	gmac_sbd_tx_clk_gating_ctrl MAC LPI transmit clock gating control status
2	RO	0x0	mac_ptp_pps MAC pulse per second status
1:0	RO	0x0	mac_mac_speed MAC sideband speed status. 2'b00: 1000 Mbps (GMII) 2'b01: 2500 Mbps (GMII) 2'b10: 10 Mbps (MII) 2'b11: 100 Mbps (MII)

GRF_MAC_STATUS1

Address: Operational Base + offset (0x0088)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	gmac_ptp_timestamp_31_0 MAC reference time output bit[31:0]

GRF_MAC_STATUS2

Address: Operational Base + offset (0x008C)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	gmac_ptp_timestamp_63_32 MAC reference time output bit[63:32]

GRF_MEM_CON0

Address: Operational Base + offset (0x0090)

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

Bit	Attr	Reset Value	Description
15:14	RO	0x0	reserved
13:8	RW	0x05	mem_romcfg ROM EMA configuration. bit[0]: RME bit[3:1]: RM bit[4]: LS bit[5]: TEST1
7:0	RW	0x05	core_mem_spcfg CORE memory EMA configuration. bit[0]: RME bit[3:1]: RM bit[4]: LS bit[5]: BC1 bit[6]: BC2 bit[7]: TEST1

GRF_MEM_CON1

Address: Operational Base + offset (0x0094)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RW	0x05	mem_hssprfcfg HSSPRF memory EMA configuration. bit[0]: RME bit[3:1]: RM bit[4]: LS bit[5]: BC1 bit[6]: BC2 bit[7]: TEST1
7:0	RW	0x05	mem_sprfcfg SPRF memory EMA configuration. bit[0]: RME bit[3:1]: RM bit[4]: LS bit[5]: BC1 bit[6]: BC2 bit[7]: TEST1

GRF_MEM_CON2

Address: Operational Base + offset (0x0098)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:10	RO	0x00	reserved

Bit	Attr	Reset Value	Description
9:0	RW	0x0a5	mem_dprfcfg DPRF memory EMA configuration. bit[0]: RMEA bit[3:1]: RMA bit[4]: LS bit[5]: RMEB bit[8:6]: RMB bit[9]: TEST1

GRF_MEM_CON3

Address: Operational Base + offset (0x009C)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13:0	RW	0x2805	mem_spracfg SPRA memory EMA configuration. bit[0]: RME bit[3:1]: RM bit[4]: LS bit[5]: BC1 bit[6]: BC2 bit[7]: TEST1 bit[10:8]: WPULSE bit[13:11]: WA

GRF_TSADC_CON

Address: Operational Base + offset (0x0100)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14	RW	0x0	pvt_shut_2gpio_sel pvt_shut_2gpio selection. 1'b0: pvt_shut_2gpio valid if either cpu_pvt_shut_2gpio or npu_pvt_shut_2gpio valid 1'b1: pvt_shut_2gpio valid if both cpu_pvt_shut_2gpio and npu_pvt_shut_2gpio valid
13	RW	0x0	npu_pvt_shut_2gpio_ena Enable NPU pvt_shut_2gpio. 1'b0: Disable 1'b1: Enable
12	RW	0x0	cpu_pvt_shut_2gpio_ena Enable CPU pvt_shut_2gpio. 1'b0: Disable 1'b1: Enable
11	RW	0x0	npupvt_shut_2cru_ena Enable NPU pvt_shut to reset CRU. 1'b0: Disable 1'b0: Enable

Bit	Attr	Reset Value	Description
10	RW	0x0	cpupvt_shut_2cru_ena Enable CPU pvt_shut to reset CRU. 1'b0: Disable 1'b0: Enable
9	RW	0x1	npu_pvt_tdc_mode NPU PVT TDC mode enable signal. 1'b0: VDC(voltage sensor) mode 1'b1: TDC(temp sensor) mode
8	RW	0x1	cpu_pvt_tdc_mode CPU PVT TDC mode enable signal. 1'b0: VDC(voltage sensor) mode 1'b1: TDC(temp sensor) mode
7:4	RW	0x7	npu_pvt_trim NPU PVT bandgap trim signal
3:0	RW	0x7	cpu_pvt_trim CPU PVT bandgap trim signal

GRF_CHIP_ID

Address: Operational Base + offset (0x0110)

Bit	Attr	Reset Value	Description
31:0	RO	0x00001109	chipid Chip ID

GRF_GPIO0C_IOMUX_H

Address: Operational Base + offset (0x10000)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14:12	RW	0x0	gpio0c7_sel 3'h0: GPIO0_C7_u 3'h1: FLASH_D3 3'h2: EMMC_D3 Others: Reserved
11	RO	0x0	reserved
10:8	RW	0x0	gpio0c6_sel 3'h0: GPIO0_C6_u 3'h1: FLASH_D2 3'h2: EMMC_D2 Others: Reserved
7	RO	0x0	reserved
6:4	RW	0x0	gpio0c5_sel 3'h0: GPIO0_C5_u 3'h1: FLASH_D1 3'h2: EMMC_D1 Others: Reserved
3	RO	0x0	reserved
2:0	RW	0x0	gpio0c4_sel 3'h0: GPIO0_C4_u 3'h1: FLASH_D0 3'h2: EMMC_D0 Others: Reserved

GRF_GPIO0D_IOMUX_L

Address: Operational Base + offset (0x10004)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14:12	RW	0x0	gpio0d3_sel 3'h0: GPIO0_D3_u 3'h1: FLASH_D7 3'h2: EMMC_D7 Others: Reserved
11	RO	0x0	reserved
10:8	RW	0x0	gpio0d2_sel 3'h0: GPIO0_D2_u 3'h1: FLASH_D6 3'h2: EMMC_D6 Others: Reserved
7	RO	0x0	reserved
6:4	RW	0x0	gpio0d1_sel 3'h0: GPIO0_D1_u 3'h1: FLASH_D5 3'h2: EMMC_D5 3'h3: FSPI_CS1N Others: Reserved
3	RO	0x0	reserved
2:0	RW	0x0	gpio0d0_sel 3'h0: GPIO0_D0_u 3'h1: FLASH_D4 3'h2: EMMC_D4 Others: Reserved

GRF_GPIO0D_IOMUX_H

Address: Operational Base + offset (0x10008)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14:12	RW	0x0	gpio0d7_sel 3'h0: GPIO0_D7_d 3'h1: FLASH_CLE 3'h2: EMMC_CLKO Others: Reserved
11	RO	0x0	reserved
10:8	RW	0x0	gpio0d6_sel 3'h0: GPIO0_D6_d 3'h3: FSPI_D2 3'h4: I2S1_SDO_M0 Others: Reserved
7	RO	0x0	reserved

Bit	Attr	Reset Value	Description
6:4	RW	0x0	gpio0d5_sel 3'h0: GPIO0_D5_u 3'h1: FLASH_WRN 3'h2: EMMC_CMD Others: Reserved
3	RO	0x0	reserved
2:0	RW	0x0	gpio0d4_sel 3'h0: GPIO0_D4_u 3'h1: FLASH_CS0N 3'h3: FSPI_CS0N 3'h4: I2S1_MCLK_M0 Others: Reserved

GRF_GPIO1A_IOMUX_L

Address: Operational Base + offset (0x10010)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14:12	RW	0x0	gpio1a3_sel 3'h0: GPIO1_A3_d 3'h1: FLASH_WPN 3'h2: EMMC_RSTn 3'h3: FSPI_CLK Others: Reserved
11	RO	0x0	reserved
10:8	RW	0x0	gpio1a2_sel 3'h0: GPIO1_A2_u 3'h1: FLASH_RDN 3'h3: FSPI_D3 3'h4: I2S1_SDI_M0 Others: Reserved
7	RO	0x0	reserved
6:4	RW	0x0	gpio1a1_sel 3'h0: GPIO1_A1_u 3'h1: FLASH_RDYN 3'h3: FSPI_D1 3'h4: I2S1_SCLK_M0 Others: Reserved
3	RO	0x0	reserved
2:0	RW	0x0	gpio1a0_sel 3'h0: GPIO1_A0_d 3'h1: FLASH_ALE 3'h3: FSPI_D0 3'h4: I2S1_LRCK_M0 Others: Reserved

GRF_GPIO1A_IOMUX_H

Address: Operational Base + offset (0x10014)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14:12	RW	0x0	gpio1a7_sel 3'h0: GPIO1_A7_u 3'h1: SDMMC0_D3 3'h2: UART3_TX_M1 3'h3: A7_JTAG_TMS_M0 3'h4: MCU_JTAG_TMS Others: Reserved
11	RO	0x0	reserved
10:8	RW	0x0	gpio1a6_sel 3'h0: GPIO1_A6_u 3'h1: SDMMC0_D2 3'h2: UART3_RX_M1 3'h3: A7_JTAG_TCK_M0 3'h4: MCU_JTAG_TCK Others: Reserved
7	RO	0x0	reserved
6:4	RW	0x0	gpio1a5_sel 3'h0: GPIO1_A5_u 3'h1: SDMMC0_D1 3'h2: TEST_CLK0_OUT 3'h3: UART2_TX_M0 3'h4: MCU_JTAG_TRSTN Others: Reserved
3	RO	0x0	reserved
2:0	RW	0x0	gpio1a4_sel 3'h0: GPIO1_A4_u 3'h1: SDMMC0_D0 3'h2: TEST_CLK1_OUT 3'h3: UART2_RX_M0 Others: Reserved

GRF_GPIO1B_IOMUX_L

Address: Operational Base + offset (0x10018)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14:12	RW	0x0	gpio1b3_sel 3'h0: GPIO1_B3_u 3'h1: SDIO_CMD Others: Reserved
11	RO	0x0	reserved
10:8	RW	0x0	gpio1b2_sel 3'h0: GPIO1_B2_d 3'h1: SDIO_CLK Others: Reserved
7	RO	0x0	reserved

Bit	Attr	Reset Value	Description
6:4	RW	0x0	gpio1b1_sel 3'h0: GPIO1_B1_u 3'h1: SDMMC0_CMD 3'h2: UART3_CTSN_M1 3'h4: MCU_JTAG_TDI Others: Reserved
3	RO	0x0	reserved
2:0	RW	0x0	gpio1b0_sel 3'h0: GPIO1_B0_u 3'h1: SDMMC0_CLK 3'h2: UART3_RTSN_M1 3'h4: MCU_JTAG_TDO Others: Reserved

GRF_GPIO1B_IOMUX_H

Address: Operational Base + offset (0x1001C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14:12	RW	0x0	gpio1b7_sel 3'h0: GPIO1_B7_u 3'h1: SDIO_D3 Others: Reserved
11	RO	0x0	reserved
10:8	RW	0x0	gpio1b6_sel 3'h0: GPIO1_B6_u 3'h1: SDIO_D2 Others: Reserved
7	RO	0x0	reserved
6:4	RW	0x0	gpio1b5_sel 3'h0: GPIO1_B5_u 3'h1: SDIO_D1 Others: Reserved
3	RO	0x0	reserved
2:0	RW	0x0	gpio1b4_sel 3'h0: GPIO1_B4_u 3'h1: SDIO_D0 Others: Reserved

GRF_GPIO1C_IOMUX_L

Address: Operational Base + offset (0x10020)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14:12	RW	0x0	gpio1c3_sel 3'h0: GPIO1_C3_u 3'h1: UART0_TX Others: Reserved
11	RO	0x0	reserved

Bit	Attr	Reset Value	Description
10:8	RW	0x0	gpio1c2_sel 3'h0: GPIO1_C2_u 3'h1: UART0_RX Others: Reserved
7	RO	0x0	reserved
6:4	RW	0x0	gpio1c1_sel 3'h0: GPIO1_C1_u 3'h1: UART0_CTSN Others: Reserved
3	RO	0x0	reserved
2:0	RW	0x0	gpio1c0_sel 3'h0: GPIO1_C0_u 3'h1: UART0_RTSN Others: Reserved

GRF_GPIO1C_IOMUX_H

Address: Operational Base + offset (0x10024)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14:12	RW	0x0	gpio1c7_sel 3'h0: GPIO1_C7_d 3'h1: I2S2_LRCK_M0 3'h3: SPI1_CS0N_M1 3'h5: UART1_CTSN_M1 Others: Reserved
11	RO	0x0	reserved
10:8	RW	0x0	gpio1c6_sel 3'h0: GPIO1_C6_d 3'h1: I2S2_SCLK_M0 3'h3: SPI1_CLK_M1 3'h4: PRELIGHT_TRIG_OUT 3'h5: UART1_RTSN_M1 Others: Reserved
7	RO	0x0	reserved
6:4	RW	0x0	gpio1c5_sel 3'h0: GPIO1_C5_d 3'h1: I2S2_SDI_M0 3'h3: SPI1_MISO_M1 3'h4: FLASH_TRIG_IN Others: Reserved
3	RO	0x0	reserved
2:0	RW	0x0	gpio1c4_sel 3'h0: GPIO1_C4_d 3'h1: I2S2_SDO_M0 3'h3: SPI1_MOSI_M1 3'h4: FLASH_TRIG_OUT Others: Reserved

GRF_GPIO1D_IOMUX_L

Address: Operational Base + offset (0x10028)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14:12	RW	0x0	gpio1d3_sel 3'h0: GPIO1_D3_u 3'h1: I2C1_SCL 3'h3: UART4_CTSN_M2 Others: Reserved
11	RO	0x0	reserved
10:8	RW	0x0	gpio1d2_sel 3'h0: GPIO1_D2_u 3'h1: I2C1_SDA 3'h3: UART4_RTSN_M2 Others: Reserved
7	RO	0x0	reserved
6:4	RW	0x0	gpio1d1_sel 3'h0: GPIO1_D1_d 3'h2: SDIO_PWR 3'h4: I2C5_SDA_M2 3'h5: UART1_RX_M1 Others: Reserved
3	RO	0x0	reserved
2:0	RW	0x0	gpio1d0_sel 3'h0: GPIO1_D0_d 3'h1: I2S2_MCLK_M0 3'h2: SDIO_DET 3'h3: SPI1_CS1N_M1 3'h4: I2C5_SCL_M2 3'h5: UART1_TX_M1 Others: Reserved

GRF_GPIO1D_IOMUX_H

Address: Operational Base + offset (0x1002C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14:12	RW	0x0	gpio1d7_sel 3'h0: GPIO1_D7_d 3'h1: SPI0_MISO_M1 3'h2: I2S1_LRCK_M1 3'h3: I2C3_SDA_M2 Others: Reserved
11	RO	0x0	reserved
10:8	RW	0x0	gpio1d6_sel 3'h0: GPIO1_D6_d 3'h1: SPI0_MOSI_M1 3'h2: I2S1_SCLK_M1 3'h3: I2C3_SCL_M2 Others: Reserved
7	RO	0x0	reserved

Bit	Attr	Reset Value	Description
6:4	RW	0x0	gpio1d5_sel 3'h0: GPIO1_D5_d 3'h1: SPI0_CS1N_M1 3'h2: I2S1_MCLK_M1 3'h3: UART4_TX_M2 Others: Reserved
3	RO	0x0	reserved
2:0	RW	0x0	gpio1d4_sel 3'h0: GPIO1_D4_d 3'h3: UART4_RX_M2 Others: Reserved

GRF_GPIO2A_IOMUX_L

Address: Operational Base + offset (0x10030)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14:12	RW	0x0	gpio2a3_sel 3'h0: GPIO2_A3_d 3'h1: MIPICSI_CLK0 3'h3: UART5_CTSN_M2 Others: Reserved
11	RO	0x0	reserved
10:8	RW	0x0	gpio2a2_sel 3'h0: GPIO2_A2_d 3'h1: MIPICSI_CLK1 3'h3: UART5_RTSN_M2 Others: Reserved
7	RO	0x0	reserved
6:4	RW	0x0	gpio2a1_sel 3'h0: GPIO2_A1_d 3'h1: SPI0_CLK_M1 3'h2: I2S1_SDO_M1 3'h3: UART5_RX_M2 Others: Reserved
3	RO	0x0	reserved
2:0	RW	0x0	gpio2a0_sel 3'h0: GPIO2_A0_d 3'h1: SPI0_CS0N_M1 3'h2: I2S1_SDI_M1 3'h3: UART5_TX_M2 Others: Reserved

GRF_GPIO2A_IOMUX_H

Address: Operational Base + offset (0x10034)

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

Bit	Attr	Reset Value	Description
14:12	RW	0x0	gpio2a7_sel 3'h0: GPIO2_A7_d 3'h1: LCDC_D3 3'h2: I2S2_SDO_M1 3'h4: UART4_RX_M1 3'h5: PWM4_PIN_M1 3'h6: SPI0_CS0N_M2 Others: Reserved
11	RO	0x0	reserved
10:8	RW	0x0	gpio2a6_sel 3'h0: GPIO2_A6_d 3'h1: LCDC_D2 3'h2: RGMII_COL_M1 3'h3: CIF_D2_M1 3'h4: UART4_TX_M1 3'h5: PWM5_PIN_M1 Others: Reserved
7	RO	0x0	reserved
6:4	RW	0x0	gpio2a5_sel 3'h0: GPIO2_A5_d 3'h1: LCDC_D1 3'h2: RGMII_CRS_M1 3'h3: CIF_D1_M1 3'h4: UART4_CTSN_M1 3'h7: I2C5_SCL_M0 Others: Reserved
3	RO	0x0	reserved
2:0	RW	0x0	gpio2a4_sel 3'h0: GPIO2_A4_d 3'h1: LCDC_D0 3'h2: RGMII_TXD3_M1 3'h3: CIF_D0_M1 3'h4: UART4_RTSN_M1 Others: Reserved

GRF_GPIO2B_IOMUX_L

Address: Operational Base + offset (0x10038)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14:12	RW	0x0	gpio2b3_sel 3'h0: GPIO2_B3_d 3'h1: LCDC_D7 3'h2: I2S2_MCLK_M1 3'h3: CIF_D3_M1 3'h4: UART5_CTSN_M1 3'h5: PWM0_PIN_M1 3'h6: SPI0_CS1N_M2 3'h7: I2C5_SDA_M0
11	RO	0x0	reserved

Bit	Attr	Reset Value	Description
10:8	RW	0x0	gpio2b2_sel 3'h0: GPIO2_B2_d 3'h1: LCDC_D6 3'h2: I2S2_LRCK_M1 3'h4: UART5_RTSN_M1 3'h5: PWM1_PIN_M1 3'h6: SPI0_CLK_M2 Others: Reserved
7	RO	0x0	reserved
6:4	RW	0x0	gpio2b1_sel 3'h0: GPIO2_B1_d 3'h1: LCDC_D5 3'h2: I2S2_SCLK_M1 3'h4: UART5_RX_M1 3'h5: PWM2_PIN_M1 3'h6: SPI0_MISO_M2 Others: Reserved
3	RO	0x0	reserved
2:0	RW	0x0	gpio2b0_sel 3'h0: GPIO2_B0_d 3'h1: LCDC_D4 3'h2: I2S2_SDI_M1 3'h4: UART5_TX_M1 3'h5: PWM3_PIN_M1 3'h6: SPI0_MOSI_M2 Others: Reserved

GRF_GPIO2B_IOMUX_H

Address: Operational Base + offset (0x1003C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14:12	RW	0x0	gpio2b7_sel 3'h0: GPIO2_B7_d 3'h1: LCDC_D11 3'h2: RGMII_CLK_M1 3'h3: CIF_D7_M1 Others: Reserved
11	RO	0x0	reserved
10:8	RW	0x0	gpio2b6_sel 3'h0: GPIO2_B6_d 3'h1: LCDC_D10 3'h2: RGMII_RXD1_M1 3'h3: CIF_D6_M1 Others: Reserved
7	RO	0x0	reserved
6:4	RW	0x0	gpio2b5_sel 3'h0: GPIO2_B5_d 3'h1: LCDC_D9 3'h2: RGMII_RXD0_M1 3'h3: CIF_D5_M1 Others: Reserved

Bit	Attr	Reset Value	Description
3	RO	0x0	reserved
2:0	RW	0x0	gpio2b4_sel 3'h0: GPIO2_B4_d 3'h1: LCDC_D8 3'h2: RGMII_RXDV_M1 3'h3: CIF_D4_M1 Others: Reserved

GRF_GPIO2C_IOMUX_L

Address: Operational Base + offset (0x10040)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14:12	RW	0x0	gpio2c3_sel 3'h0: GPIO2_C3_d 3'h1: LCDC_D15 3'h2: RGMII_TXD0_M1 3'h3: CIF_D11_M1 Others: Reserved
11	RO	0x0	reserved
10:8	RW	0x0	gpio2c2_sel 3'h0: GPIO2_C2_d 3'h1: LCDC_D14 3'h2: RGMII_MDC_M1 3'h3: CIF_D10_M1 Others: Reserved
7	RO	0x0	reserved
6:4	RW	0x0	gpio2c1_sel 3'h0: GPIO2_C1_d 3'h1: LCDC_D13 3'h2: RGMII_MDIO_M1 3'h3: CIF_D9_M1 Others: Reserved
3	RO	0x0	reserved
2:0	RW	0x0	gpio2c0_sel 3'h0: GPIO2_C0_d 3'h1: LCDC_D12 3'h2: RGMII_RXER_M1 3'h3: CIF_D8_M1 Others: Reserved

GRF_GPIO2C_IOMUX_H

Address: Operational Base + offset (0x10044)

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

Bit	Attr	Reset Value	Description
14:12	RW	0x0	gpio2c7_sel 3'h0: GPIO2_C7_d 3'h1: LCDC_D19 3'h2: RGMII_RXD2_M1 3'h3: CIF_D15_M1 3'h6: I2S1_MCLK_M2 Others: Reserved
11	RO	0x0	reserved
10:8	RW	0x0	gpio2c6_sel 3'h0: GPIO2_C6_d 3'h1: LCDC_D18 3'h2: RGMII_TXEN_M1 3'h3: CIF_D14_M1 Others: Reserved
7	RO	0x0	reserved
6:4	RW	0x0	gpio2c5_sel 3'h0: GPIO2_C5_d 3'h1: LCDC_D17 3'h2: CLK_OUT_ETHERNET_M1 3'h3: CIF_D13_M1 Others: Reserved
3	RO	0x0	reserved
2:0	RW	0x0	gpio2c4_sel 3'h0: GPIO2_C4_d 3'h1: LCDC_D16 3'h2: RGMII_TXD1_M1 3'h3: CIF_D12_M1 Others: Reserved

GRF_GPIO2D_IOMUX_L

Address: Operational Base + offset (0x10048)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14:12	RW	0x0	gpio2d3_sel 3'h0: GPIO2_D3_d 3'h1: LCDC_D23 3'h2: RGMII_RXCLK_M1 3'h3: CIF_HSYNC_M1 3'h6: I2S1_SDI_M2 Others: Reserved
11	RO	0x0	reserved
10:8	RW	0x0	gpio2d2_sel 3'h0: GPIO2_D2_d 3'h1: LCDC_D22 3'h2: RGMII_TXCLK_M1 3'h3: CIF_CLKIN_M1 3'h6: I2S1_LRCK_M2 Others: Reserved
7	RO	0x0	reserved

Bit	Attr	Reset Value	Description
6:4	RW	0x0	gpio2d1_sel 3'h0: GPIO2_D1_d 3'h1: LCDC_D21 3'h2: RGMII_TXD2_M1 3'h3: CIF_CLKOUT_M1 3'h6: I2S1_SCLK_M2 Others: Reserved
3	RO	0x0	reserved
2:0	RW	0x0	gpio2d0_sel 3'h0: GPIO2_D0_d 3'h1: LCDC_D20 3'h2: RGMII_RXD3_M1 3'h3: CIF_VSYNC_M1 3'h6: I2S1_SDO_M2 Others: Reserved

GRF_GPIO2D_IOMUX_H

Address: Operational Base + offset (0x1004C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14:12	RW	0x0	gpio2d7_sel 3'h0: GPIO2_D7_d 3'h1: LCDC_CLK 3'h4: UART3_CTSN_M2 3'h5: PWM8_PIN_M1 3'h6: SPI1_MISO_M2 Others: Reserved
11	RO	0x0	reserved
10:8	RW	0x0	gpio2d6_sel 3'h0: GPIO2_D6_d 3'h1: LCDC_VSYNC 3'h4: UART3_RTSN_M2 3'h5: PWM9_PIN_M1 3'h6: SPI1_MOSI_M2 Others: Reserved
7	RO	0x0	reserved
6:4	RW	0x0	gpio2d5_sel 3'h0: GPIO2_D5_d 3'h1: LCDC_HSYNC 3'h5: PWM10_PIN_M1 3'h6: SPI1_CLK_M2 3'h7: I2C3_SDA_M1 Others: Reserved
3	RO	0x0	reserved
2:0	RW	0x0	gpio2d4_sel 3'h0: GPIO2_D4_d 3'h1: LCDC_DEN 3'h5: PWM6_PIN_M1 3'h6: SPI1_CS0N_M2 3'h7: I2C3_SCL_M1 Others: Reserved

GRF_GPIO3A_IOMUX_L

Address: Operational Base + offset (0x10050)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14:12	RW	0x0	gpio3a3_sel 3'h0: GPIO3_A3_u 3'h1: UART2_RX_M1 3'h2: A7_JTAG_TMS_M1 Others: Reserved
11	RO	0x0	reserved
10:8	RW	0x0	gpio3a2_sel 3'h0: GPIO3_A2_u 3'h1: UART2_TX_M1 3'h2: A7_JTAG_TCK_M1 Others: Reserved
7	RO	0x0	reserved
6:4	RW	0x0	gpio3a1_sel 3'h0: GPIO3_A1_u 3'h3: CAN_TXD_M0 3'h4: UART3_RX_M2 3'h5: PWM11_PIN_M1 3'h7: I2C4_SDA_M0 Others: Reserved
3	RO	0x0	reserved
2:0	RW	0x0	gpio3a0_sel 3'h0: GPIO3_A0_u 3'h3: CAN_RXD_M0 3'h4: UART3_TX_M2 3'h5: PWM7_PIN_M1 3'h6: SPI1_CS1N_M2 3'h7: I2C4_SCL_M0 Others: Reserved

GRF_GPIO3A_IOMUX_H

Address: Operational Base + offset (0x10054)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14:12	RW	0x0	gpio3a7_sel 3'h0: GPIO3_A7_d 3'h1: CIF_D3_M0 3'h2: RGMII_RXD2_M0 3'h3: I2S0_SDI0_M1 3'h4: UART5_RX_M0 3'h5: CAN_TXD_M1 3'h6: PWM11_PIN_M0 Others: Reserved
11	RO	0x0	reserved

Bit	Attr	Reset Value	Description
10:8	RW	0x0	gpio3a6_sel 3'h0: GPIO3_A6_d 3'h1: CIF_D2_M0 3'h2: RGMII_COL_M0 3'h3: I2S0_SDO0_M1 3'h4: UART5_TX_M0 3'h5: CAN_RXD_M1 3'h6: PWM10_PIN_M0 Others: Reserved
7	RO	0x0	reserved
6:4	RW	0x0	gpio3a5_sel 3'h0: GPIO3_A5_d 3'h1: CIF_D1_M0 3'h2: RGMII_CRS_M0 3'h3: I2S0_LRCK_TX_M1 3'h4: UART4_RX_M0 3'h5: I2C3_SDA_M0 3'h6: PWM9_PIN_M0 Others: Reserved
3	RO	0x0	reserved
2:0	RW	0x0	gpio3a4_sel 3'h0: GPIO3_A4_d 3'h1: CIF_D0_M0 3'h3: I2S0_SCLK_TX_M1 3'h4: UART4_TX_M0 3'h5: I2C3_SCL_M0 3'h6: PWM8_PIN_M0 Others: Reserved

GRF_GPIO3B_IOMUX_L

Address: Operational Base + offset (0x10058)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14:12	RW	0x0	gpio3b3_sel 3'h0: GPIO3_B3_d 3'h1: CIF_D7_M0 3'h2: RGMII_TXD0_M0 3'h3: I2S0_SDO1_SDI3_M1 3'h4: UART4_CTSN_M0 Others: Reserved
11	RO	0x0	reserved
10:8	RW	0x0	gpio3b2_sel 3'h0: GPIO3_B2_d 3'h1: CIF_D6_M0 3'h2: RGMII_TXD3_M0 3'h3: I2S0_LRCK_RX_M1 3'h4: UART4_RTSN_M0 Others: Reserved
7	RO	0x0	reserved

Bit	Attr	Reset Value	Description
6:4	RW	0x0	gpio3b1_sel 3'h0: GPIO3_B1_d 3'h1: CIF_D5_M0 3'h2: RGMII_TXD2_M0 3'h3: I2S0_SCLK_RX_M1 3'h4: UART5_CTSN_M0 3'h5: I2C5_SDA_M1 Others: Reserved
3	RO	0x0	reserved
2:0	RW	0x0	gpio3b0_sel 3'h0: GPIO3_B0_d 3'h1: CIF_D4_M0 3'h2: RGMII_RXD3_M0 3'h3: I2S0_MCLK_M1 3'h4: UART5_RTSN_M0 3'h5: I2C5_SCL_M1 Others: Reserved

GRF_GPIO3B_IOMUX_H

Address: Operational Base + offset (0x1005C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14:12	RW	0x0	gpio3b7_sel 3'h0: GPIO3_B7_d 3'h1: CIF_D11_M0 3'h2: RGMII_RXD1_M0 3'h3: PDM_SDI3_M1 3'h5: SPI1_MISO_M0 Others: Reserved
11	RO	0x0	reserved
10:8	RW	0x0	gpio3b6_sel 3'h0: GPIO3_B6_d 3'h1: CIF_D10_M0 3'h2: RGMII_RXD0_M0 3'h3: PDM_SDI2_M1 3'h5: SPI1_MOSI_M0 Others: Reserved
7	RO	0x0	reserved
6:4	RW	0x0	gpio3b5_sel 3'h0: GPIO3_B5_d 3'h1: CIF_D9_M0 3'h2: RGMII_TXEN_M0 3'h3: I2S0_SDO3_SDI1_M1 3'h5: SPI1_CS0N_M0 Others: Reserved
3	RO	0x0	reserved

Bit	Attr	Reset Value	Description
2:0	RW	0x0	gpio3b4_sel 3'h0: GPIO3_B4_d 3'h1: CIF_D8_M0 3'h2: RGMII_TXD1_M0 3'h3: I2S0_SDO2_SDI2_M1 3'h5: SPI1_CS1N_M0 Others: Reserved

GRF_GPIO3C_IOMUX_L

Address: Operational Base + offset (0x10060)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14:12	RW	0x0	gpio3c3_sel 3'h0: GPIO3_C3_d 3'h1: CIF_D15_M0 3'h2: RGMII_MDIO_M0 3'h3: PDM_CLK1_M1 Others: Reserved
11	RO	0x0	reserved
10:8	RW	0x0	gpio3c2_sel 3'h0: GPIO3_C2_d 3'h1: CIF_D14_M0 3'h2: RGMII_RXER_M0 3'h3: PDM_SDI1_M1 Others: Reserved
7	RO	0x0	reserved
6:4	RW	0x0	gpio3c1_sel 3'h0: GPIO3_C1_d 3'h1: CIF_D13_M0 3'h2: RGMII_RXDV_M0 3'h3: PDM_SDI0_M1 Others: Reserved
3	RO	0x0	reserved
2:0	RW	0x0	gpio3c0_sel 3'h0: GPIO3_C0_d 3'h1: CIF_D12_M0 3'h2: RGMII_CLK_M0 3'h3: PDM_CLK0_M1 3'h5: SPI1_CLK_M0 Others: Reserved

GRF_GPIO3C_IOMUX_H

Address: Operational Base + offset (0x10064)

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

Bit	Attr	Reset Value	Description
14:12	RW	0x0	gpio3c7_sel 3'h0: GPIO3_C7_d 3'h1: CIF_HSYNC_M0 3'h2: RGMII_RXCLK_M0 3'h4: UART3_RX_M0 Others: Reserved
11	RO	0x0	reserved
10:8	RW	0x0	gpio3c6_sel 3'h0: GPIO3_C6_d 3'h1: CIF_CLKOUT_M0 3'h2: RGMII_TXCLK_M0 3'h4: UART3_TX_M0 Others: Reserved
7	RO	0x0	reserved
6:4	RW	0x0	gpio3c5_sel 3'h0: GPIO3_C5_d 3'h1: CIF_CLKIN_M0 3'h2: CLK_OUT_ETHERNET_M0 3'h4: UART3_CTSN_M0 Others: Reserved
3	RO	0x0	reserved
2:0	RW	0x0	gpio3c4_sel 3'h0: GPIO3_C4_d 3'h1: CIF_VSYNC_M0 3'h2: RGMII_MDC_M0 3'h4: UART3_RTSN_M0 Others: Reserved

GRF_GPIO3D_IOMUX_L

Address: Operational Base + offset (0x10068)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14:12	RW	0x0	gpio3d3_sel 3'h0: GPIO3_D3_d 3'h1: I2S0_LRCK_TX_M0 3'h3: ACODEC_DAC_SYNC 3'h4: AUDPWM_L_M1 3'h5: AUDDSM_LN Others: Reserved
11	RO	0x0	reserved
10:8	RW	0x0	gpio3d2_sel 3'h0: GPIO3_D2_d 3'h1: I2S0_MCLK_M0 Others: Reserved
7	RO	0x0	reserved
6:4	RW	0x0	gpio3d1_sel 3'h0: GPIO3_D1_d 3'h1: I2S0_SCLK_RX_M0 3'h2: PDM_CLK1_M0 3'h3: ACODEC_ADC_CLK Others: Reserved

Bit	Attr	Reset Value	Description
3	RO	0x0	reserved
2:0	RW	0x0	gpio3d0_sel 3'h0: GPIO3_D0_d 3'h1: I2S0_SCLK_TX_M0 3'h3: ACODEC_DAC_CLK Others: Reserved

GRF_GPIO3D_IOMUX_H

Address: Operational Base + offset (0x1006C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14:12	RW	0x0	gpio3d7_sel 3'h0: GPIO3_D7_d 3'h1: I2S0_SDO1_SDI3_M0 3'h2: PDM_SDI3_M0 3'h3: ACODEC_ADC_DATA Others: Reserved
11	RO	0x0	reserved
10:8	RW	0x0	gpio3d6_sel 3'h0: GPIO3_D6_d 3'h1: I2S0_SDI0_M0 3'h2: PDM_SDI0_M0 3'h3: ACODEC_DAC_DATAL Others: Reserved
7	RO	0x0	reserved
6:4	RW	0x0	gpio3d5_sel 3'h0: GPIO3_D5_d 3'h1: I2S0_SDO0_M0 3'h3: ACODEC_DAC_DATAR 3'h4: AUDPWM_R_M1 3'h5: AUDDSM_LP Others: Reserved
3	RO	0x0	reserved
2:0	RW	0x0	gpio3d4_sel 3'h0: GPIO3_D4_d 3'h1: I2S0_LRCK_RX_M0 3'h2: PDM_CLK0_M0 3'h3: ACODEC_ADC_SYNC Others: Reserved

GRF_GPIO4A_IOMUX_L

Address: Operational Base + offset (0x10070)

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

Bit	Attr	Reset Value	Description
6:4	RW	0x0	gpio4a1_sel 3'h0: GPIO4_A1_d 3'h1: I2S0_SDO3_SDI1_M0 3'h2: PDM_SDI1_M0 3'h3: AUDPWM_R_M0 3'h4: I2C4_SDA_M1 3'h5: AUDDSM_RP Others: Reserved
3	RO	0x0	reserved
2:0	RW	0x0	gpio4a0_sel 3'h0: GPIO4_A0_d 3'h1: I2S0_SDO2_SDI2_M0 3'h2: PDM_SDI2_M0 3'h3: AUDPWM_L_M0 3'h4: I2C4_SCL_M1 3'h5: AUDDSM_RN Others: Reserved

GRF_GPIO0C_DS_H

Address: Operational Base + offset (0x10080)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RW	0x1	gpio0c7_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
11:8	RW	0x1	gpio0c6_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
7:4	RW	0x1	gpio0c5_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
3:0	RW	0x1	gpio0c4_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved

GRF_GPIO0D_DS_L

Address: Operational Base + offset (0x10084)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RW	0x1	gpio0d3_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
11:8	RW	0x1	gpio0d2_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
7:4	RW	0x1	gpio0d1_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
3:0	RW	0x1	gpio0d0_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved

GRF_GPIO0D_DS_H

Address: Operational Base + offset (0x10088)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RW	0x2	gpio0d7_ds 4'd0: Level 0 strength 4'd1: Level 1 strength 4'd2: Level 2 strength 4'd3: Level 3 strength 4'd4: Level 4 strength 4'd5: Level 5 strength 4'd6: Level 6 strength 4'd7: Level 7 strength 4'd8: Level 8 strength 4'd9: Level 9 strength 4'd10: Level 10 strength 4'd11: Level 11 strength 4'd12: Level 12 strength 4'd13: Level 13 strength 4'd14: Level 14 strength 4'd15: Level 15 strength

Bit	Attr	Reset Value	Description
11:8	RW	0x1	gpio0d6_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
7:4	RW	0x1	gpio0d5_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
3:0	RW	0x1	gpio0d4_ds 4'd0: Level 0 strength 4'd1: Level 1 strength 4'd2: Level 2 strength 4'd3: Level 3 strength 4'd4: Level 4 strength 4'd5: Level 5 strength 4'd6: Level 6 strength 4'd7: Level 7 strength 4'd8: Level 8 strength 4'd9: Level 9 strength 4'd10: Level 10 strength 4'd11: Level 11 strength 4'd12: Level 12 strength 4'd13: Level 13 strength 4'd14: Level 14 strength 4'd15: Level 15 strength

GRF_GPIO1A_DS_L

Address: Operational Base + offset (0x10090)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RW	0x2	gpio1a3_ds 4'd0: Level 0 strength 4'd1: Level 1 strength 4'd2: Level 2 strength 4'd3: Level 3 strength 4'd4: Level 4 strength 4'd5: Level 5 strength 4'd6: Level 6 strength 4'd7: Level 7 strength 4'd8: Level 8 strength 4'd9: Level 9 strength 4'd10: Level 10 strength 4'd11: Level 11 strength 4'd12: Level 12 strength 4'd13: Level 13 strength 4'd14: Level 14 strength 4'd15: Level 15 strength

Bit	Attr	Reset Value	Description
11:8	RW	0x1	gpio1a2_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
7:4	RW	0x1	gpio1a1_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
3:0	RW	0x1	gpio1a0_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved

GRF_GPIO1A_DS_H

Address: Operational Base + offset (0x10094)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RW	0x1	gpio1a7_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
11:8	RW	0x1	gpio1a6_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
7:4	RW	0x1	gpio1a5_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
3:0	RW	0x1	gpio1a4_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved

GRF_GPIO1B_DS_L

Address: Operational Base + offset (0x10098)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RW	0x1	gpio1b3_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
11:8	RW	0x2	gpio1b2_ds 4'd0: Level 0 strength 4'd1: Level 1 strength 4'd2: Level 2 strength 4'd3: Level 3 strength 4'd4: Level 4 strength 4'd5: Level 5 strength 4'd6: Level 6 strength 4'd7: Level 7 strength 4'd8: Level 8 strength 4'd9: Level 9 strength 4'd10: Level 10 strength 4'd11: Level 11 strength 4'd12: Level 12 strength 4'd13: Level 13 strength 4'd14: Level 14 strength 4'd15: Level 15 strength
7:4	RW	0x1	gpio1b1_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
3:0	RW	0x2	gpio1b0_ds 4'd0: Level 0 strength 4'd1: Level 1 strength 4'd2: Level 2 strength 4'd3: Level 3 strength 4'd4: Level 4 strength 4'd5: Level 5 strength 4'd6: Level 6 strength 4'd7: Level 7 strength 4'd8: Level 8 strength 4'd9: Level 9 strength 4'd10: Level 10 strength 4'd11: Level 11 strength 4'd12: Level 12 strength 4'd13: Level 13 strength 4'd14: Level 14 strength 4'd15: Level 15 strength

GRF_GPIO1B_DS_H

Address: Operational Base + offset (0x1009C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RW	0x1	gpio1b7_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
11:8	RW	0x1	gpio1b6_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
7:4	RW	0x1	gpio1b5_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
3:0	RW	0x1	gpio1b4_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved

GRF_GPIO1C_DS_L

Address: Operational Base + offset (0x100A0)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RW	0x1	gpio1c3_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
11:8	RW	0x1	gpio1c2_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
7:4	RW	0x1	gpio1c1_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved

Bit	Attr	Reset Value	Description
3:0	RW	0x1	gpio1c0_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved

GRF_GPIO1C_DS_H

Address: Operational Base + offset (0x100A4)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RW	0x1	gpio1c7_ds 4'd0: Level 0 strength 4'd1: Level 1 strength 4'd2: Level 2 strength 4'd3: Level 3 strength 4'd4: Level 4 strength 4'd5: Level 5 strength 4'd6: Level 6 strength 4'd7: Level 7 strength 4'd8: Level 8 strength 4'd9: Level 9 strength 4'd10: Level 10 strength 4'd11: Level 11 strength 4'd12: Level 12 strength 4'd13: Level 13 strength 4'd14: Level 14 strength 4'd15: Level 15 strength
11:8	RW	0x1	gpio1c6_ds 4'd0: Level 0 strength 4'd1: Level 1 strength 4'd2: Level 2 strength 4'd3: Level 3 strength 4'd4: Level 4 strength 4'd5: Level 5 strength 4'd6: Level 6 strength 4'd7: Level 7 strength 4'd8: Level 8 strength 4'd9: Level 9 strength 4'd10: Level 10 strength 4'd11: Level 11 strength 4'd12: Level 12 strength 4'd13: Level 13 strength 4'd14: Level 14 strength 4'd15: Level 15 strength
7:4	RW	0x1	gpio1c5_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved

Bit	Attr	Reset Value	Description
3:0	RW	0x1	gpio1c4_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved

GRF_GPIO1D_DS_L

Address: Operational Base + offset (0x100A8)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RW	0x1	gpio1d3_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
11:8	RW	0x1	gpio1d2_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
7:4	RW	0x1	gpio1d1_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
3:0	RW	0x1	gpio1d0_ds 4'd0: Level 0 strength 4'd1: Level 1 strength 4'd2: Level 2 strength 4'd3: Level 3 strength 4'd4: Level 4 strength 4'd5: Level 5 strength 4'd6: Level 6 strength 4'd7: Level 7 strength 4'd8: Level 8 strength 4'd9: Level 9 strength 4'd10: Level 10 strength 4'd11: Level 11 strength 4'd12: Level 12 strength 4'd13: Level 13 strength 4'd14: Level 14 strength 4'd15: Level 15 strength

GRF_GPIO1D_DS_H

Address: Operational Base + offset (0x100AC)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RW	0x1	gpio1d7_ds 4'd0: Level 0 strength 4'd1: Level 1 strength 4'd2: Level 2 strength 4'd3: Level 3 strength 4'd4: Level 4 strength 4'd5: Level 5 strength 4'd6: Level 6 strength 4'd7: Level 7 strength 4'd8: Level 8 strength 4'd9: Level 9 strength 4'd10: Level 10 strength 4'd11: Level 11 strength 4'd12: Level 12 strength 4'd13: Level 13 strength 4'd14: Level 14 strength 4'd15: Level 15 strength
11:8	RW	0x1	gpio1d6_ds 4'd0: Level 0 strength 4'd1: Level 1 strength 4'd2: Level 2 strength 4'd3: Level 3 strength 4'd4: Level 4 strength 4'd5: Level 5 strength 4'd6: Level 6 strength 4'd7: Level 7 strength 4'd8: Level 8 strength 4'd9: Level 9 strength 4'd10: Level 10 strength 4'd11: Level 11 strength 4'd12: Level 12 strength 4'd13: Level 13 strength 4'd14: Level 14 strength 4'd15: Level 15 strength
7:4	RW	0x1	gpio1d5_ds 4'd0: Level 0 strength 4'd1: Level 1 strength 4'd2: Level 2 strength 4'd3: Level 3 strength 4'd4: Level 4 strength 4'd5: Level 5 strength 4'd6: Level 6 strength 4'd7: Level 7 strength 4'd8: Level 8 strength 4'd9: Level 9 strength 4'd10: Level 10 strength 4'd11: Level 11 strength 4'd12: Level 12 strength 4'd13: Level 13 strength 4'd14: Level 14 strength 4'd15: Level 15 strength

Bit	Attr	Reset Value	Description
3:0	RW	0x1	gpio1d4_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved

GRF_GPIO2A_DS_L

Address: Operational Base + offset (0x100B0)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RW	0x2	gpio2a3_ds 4'd0: Level 0 strength 4'd1: Level 1 strength 4'd2: Level 2 strength 4'd3: Level 3 strength 4'd4: Level 4 strength 4'd5: Level 5 strength 4'd6: Level 6 strength 4'd7: Level 7 strength 4'd8: Level 8 strength 4'd9: Level 9 strength 4'd10: Level 10 strength 4'd11: Level 11 strength 4'd12: Level 12 strength 4'd13: Level 13 strength 4'd14: Level 14 strength 4'd15: Level 15 strength
11:8	RW	0x2	gpio2a2_ds 4'd0: Level 0 strength 4'd1: Level 1 strength 4'd2: Level 2 strength 4'd3: Level 3 strength 4'd4: Level 4 strength 4'd5: Level 5 strength 4'd6: Level 6 strength 4'd7: Level 7 strength 4'd8: Level 8 strength 4'd9: Level 9 strength 4'd10: Level 10 strength 4'd11: Level 11 strength 4'd12: Level 12 strength 4'd13: Level 13 strength 4'd14: Level 14 strength 4'd15: Level 15 strength

Bit	Attr	Reset Value	Description
7:4	RW	0x1	gpio2a1_ds 4'd0: Level 0 strength 4'd1: Level 1 strength 4'd2: Level 2 strength 4'd3: Level 3 strength 4'd4: Level 4 strength 4'd5: Level 5 strength 4'd6: Level 6 strength 4'd7: Level 7 strength 4'd8: Level 8 strength 4'd9: Level 9 strength 4'd10: Level 10 strength 4'd11: Level 11 strength 4'd12: Level 12 strength 4'd13: Level 13 strength 4'd14: Level 14 strength 4'd15: Level 15 strength
3:0	RW	0x1	gpio2a0_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved

GRF_GPIO2A_DS_H

Address: Operational Base + offset (0x100B4)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RW	0x1	gpio2a7_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
11:8	RW	0x1	gpio2a6_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
7:4	RW	0x1	gpio2a5_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
3:0	RW	0x1	gpio2a4_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved

GRF_GPIO2B_DS_L

Address: Operational Base + offset (0x100B8)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RW	0x1	gpio2b3_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
11:8	RW	0x1	gpio2b2_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
7:4	RW	0x1	gpio2b1_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
3:0	RW	0x1	gpio2b0_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved

GRF_GPIO2B_DS_H

Address: Operational Base + offset (0x100BC)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RW	0x1	gpio2b7_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
11:8	RW	0x1	gpio2b6_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved

Bit	Attr	Reset Value	Description
7:4	RW	0x1	gpio2b5_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
3:0	RW	0x1	gpio2b4_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved

GRF_GPIO2C_DS_L

Address: Operational Base + offset (0x100C0)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RW	0x1	gpio2c3_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
11:8	RW	0x1	gpio2c2_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
7:4	RW	0x1	gpio2c1_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
3:0	RW	0x1	gpio2c0_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved

GRF_GPIO2C_DS_H

Address: Operational Base + offset (0x100C4)

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

Bit	Attr	Reset Value	Description
15:12	RW	0x1	gpio2c7_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
11:8	RW	0x1	gpio2c6_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
7:4	RW	0x1	gpio2c5_ds 4'd0: Level 0 strength 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
3:0	RW	0x1	gpio2c4_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved

GRF_GPIO2D_DS_L

Address: Operational Base + offset (0x100C8)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RW	0x1	gpio2d3_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
11:8	RW	0x1	gpio2d2_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
7:4	RW	0x1	gpio2d1_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved

Bit	Attr	Reset Value	Description
3:0	RW	0x1	gpio2d0_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved

GRF_GPIO2D_DS_H

Address: Operational Base + offset (0x100CC)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RW	0x2	gpio2d7_ds 4'd0: Level 0 strength 4'd1: Level 1 strength 4'd2: Level 2 strength 4'd3: Level 3 strength 4'd4: Level 4 strength 4'd5: Level 5 strength 4'd6: Level 6 strength 4'd7: Level 7 strength 4'd8: Level 8 strength 4'd9: Level 9 strength 4'd10: Level 10 strength 4'd11: Level 11 strength 4'd12: Level 12 strength 4'd13: Level 13 strength 4'd14: Level 14 strength 4'd15: Level 15 strength
11:8	RW	0x1	gpio2d6_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
7:4	RW	0x1	gpio2d5_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
3:0	RW	0x1	gpio2d4_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved

GRF_GPIO3A_DS_L

Address: Operational Base + offset (0x100D0)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RW	0x1	gpio3a3_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
11:8	RW	0x1	gpio3a2_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
7:4	RW	0x1	gpio3a1_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
3:0	RW	0x1	gpio3a0_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved

GRF_GPIO3A_DS_H

Address: Operational Base + offset (0x100D4)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RW	0x1	gpio3a7_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
11:8	RW	0x1	gpio3a6_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
7:4	RW	0x1	gpio3a5_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved

Bit	Attr	Reset Value	Description
3:0	RW	0x1	gpio3a4_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved

GRF_GPIO3B_DS_L

Address: Operational Base + offset (0x100D8)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RW	0x1	gpio3b3_ds x'bxxx(bits < 4), x'hxxxx(bits >= 4)
11:8	RW	0x1	gpio3b2_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
7:4	RW	0x1	gpio3b1_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
3:0	RW	0x1	gpio3b0_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved

GRF_GPIO3B_DS_H

Address: Operational Base + offset (0x100DC)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RW	0x1	gpio3b7_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
11:8	RW	0x1	gpio3b6_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved

Bit	Attr	Reset Value	Description
7:4	RW	0x1	gpio3b5_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
3:0	RW	0x1	gpio3b4_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved

GRF_GPIO3C_DS_L

Address: Operational Base + offset (0x100E0)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RW	0x1	gpio3c3_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
11:8	RW	0x1	gpio3c2_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
7:4	RW	0x1	gpio3c1_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
3:0	RW	0x1	gpio3c0_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved

GRF_GPIO3C_DS_H

Address: Operational Base + offset (0x100E4)

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

Bit	Attr	Reset Value	Description
15:12	RW	0x1	gpio3c7_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
11:8	RW	0x1	gpio3c6_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
7:4	RW	0x1	gpio3c5_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
3:0	RW	0x1	gpio3c4_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved

GRF_GPIO3D_DS_L

Address: Operational Base + offset (0x100E8)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RW	0x1	gpio3d3_ds 4'd0: Level 0 strength 4'd1: Level 1 strength 4'd2: Level 2 strength 4'd3: Level 3 strength 4'd4: Level 4 strength 4'd5: Level 5 strength 4'd6: Level 6 strength 4'd7: Level 7 strength 4'd8: Level 8 strength 4'd9: Level 9 strength 4'd10: Level 10 strength 4'd11: Level 11 strength 4'd12: Level 12 strength 4'd13: Level 13 strength 4'd14: Level 14 strength 4'd15: Level 15 strength

Bit	Attr	Reset Value	Description
11:8	RW	0x1	gpio3d2_ds 4'd0: Level 0 strength 4'd1: Level 1 strength 4'd2: Level 2 strength 4'd3: Level 3 strength 4'd4: Level 4 strength 4'd5: Level 5 strength 4'd6: Level 6 strength 4'd7: Level 7 strength 4'd8: Level 8 strength 4'd9: Level 9 strength 4'd10: Level 10 strength 4'd11: Level 11 strength 4'd12: Level 12 strength 4'd13: Level 13 strength 4'd14: Level 14 strength 4'd15: Level 15 strength
7:4	RW	0x1	gpio3d1_ds 4'd0: Level 0 strength 4'd1: Level 1 strength 4'd2: Level 2 strength 4'd3: Level 3 strength 4'd4: Level 4 strength 4'd5: Level 5 strength 4'd6: Level 6 strength 4'd7: Level 7 strength 4'd8: Level 8 strength 4'd9: Level 9 strength 4'd10: Level 10 strength 4'd11: Level 11 strength 4'd12: Level 12 strength 4'd13: Level 13 strength 4'd14: Level 14 strength 4'd15: Level 15 strength
3:0	RW	0x1	gpio3d0_ds 4'd0: Level 0 strength 4'd1: Level 1 strength 4'd2: Level 2 strength 4'd3: Level 3 strength 4'd4: Level 4 strength 4'd5: Level 5 strength 4'd6: Level 6 strength 4'd7: Level 7 strength 4'd8: Level 8 strength 4'd9: Level 9 strength 4'd10: Level 10 strength 4'd11: Level 11 strength 4'd12: Level 12 strength 4'd13: Level 13 strength 4'd14: Level 14 strength 4'd15: Level 15 strength

GRF_GPIO3D_DS_H

Address: Operational Base + offset (0x100EC)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RW	0x1	gpio3d7_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
11:8	RW	0x1	gpio3d6_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
7:4	RW	0x1	gpio3d5_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
3:0	RW	0x1	gpio3d4_ds 4'd0: Level 0 strength 4'd1: Level 1 strength 4'd2: Level 2 strength 4'd3: Level 3 strength 4'd4: Level 4 strength 4'd5: Level 5 strength 4'd6: Level 6 strength 4'd7: Level 7 strength 4'd8: Level 8 strength 4'd9: Level 9 strength 4'd10: Level 10 strength 4'd11: Level 11 strength 4'd12: Level 12 strength 4'd13: Level 13 strength 4'd14: Level 14 strength 4'd15: Level 15 strength

GRF_GPIO4A_DS_L

Address: Operational Base + offset (0x100F0)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7:4	RW	0x1	gpio4a1_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved

Bit	Attr	Reset Value	Description
3:0	RW	0x1	gpio4a0_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved

GRF_GPIO0C_P_H

Address: Operational Base + offset (0x10100)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RW	0x1	gpio0c7_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
13:12	RW	0x1	gpio0c6_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
11:10	RW	0x1	gpio0c5_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
9:8	RW	0x1	gpio0c4_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
7:0	RO	0xaa	reserved

GRF_GPIO0D_P

Address: Operational Base + offset (0x10104)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RW	0x2	gpio0d7_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
13:12	RW	0x2	gpio0d6_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved

Bit	Attr	Reset Value	Description
11:10	RW	0x1	gpio0d5_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
9:8	RW	0x1	gpio0d4_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
7:6	RW	0x1	gpio0d3_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
5:4	RW	0x1	gpio0d2_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
3:2	RW	0x1	gpio0d1_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
1:0	RW	0x1	gpio0d0_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved

GRF_GPIO1A_P

Address: Operational Base + offset (0x10108)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RW	0x1	gpio1a7_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
13:12	RW	0x1	gpio1a6_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
11:10	RW	0x1	gpio1a5_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved

Bit	Attr	Reset Value	Description
9:8	RW	0x1	gpio1a4_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
7:6	RW	0x2	gpio1a3_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
5:4	RW	0x1	gpio1a2_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
3:2	RW	0x1	gpio1a1_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
1:0	RW	0x2	gpio1a0_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved

GRF_GPIO1B_P

Address: Operational Base + offset (0x1010C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RW	0x1	gpio1b7_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
13:12	RW	0x1	gpio1b6_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
11:10	RW	0x1	gpio1b5_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
9:8	RW	0x1	gpio1b4_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved

Bit	Attr	Reset Value	Description
7:6	RW	0x1	gpio1b3_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
5:4	RW	0x2	gpio1b2_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
3:2	RW	0x1	gpio1b1_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
1:0	RW	0x1	gpio1b0_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved

GRF_GPIO1C_P

Address: Operational Base + offset (0x10110)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RW	0x2	gpio1c7_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
13:12	RW	0x2	gpio1c6_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
11:10	RW	0x2	gpio1c5_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
9:8	RW	0x2	gpio1c4_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
7:6	RW	0x1	gpio1c3_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved

Bit	Attr	Reset Value	Description
5:4	RW	0x1	gpio1c2_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
3:2	RW	0x1	gpio1c1_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
1:0	RW	0x1	gpio1c0_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved

GRF_GPIO1D_P

Address: Operational Base + offset (0x10114)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RW	0x2	gpio1d7_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
13:12	RW	0x2	gpio1d6_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
11:10	RW	0x2	gpio1d5_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
9:8	RW	0x2	gpio1d4_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
7:6	RW	0x1	gpio1d3_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
5:4	RW	0x1	gpio1d2_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved

Bit	Attr	Reset Value	Description
3:2	RW	0x2	gpio1d1_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
1:0	RW	0x2	gpio1d0_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved

GRF_GPIO2A_P

Address: Operational Base + offset (0x10118)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RW	0x2	gpio2a7_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
13:12	RW	0x2	gpio2a6_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
11:10	RW	0x2	gpio2a5_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
9:8	RW	0x2	gpio2a4_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
7:6	RW	0x2	gpio2a3_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
5:4	RW	0x2	gpio2a2_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
3:2	RW	0x2	gpio2a1_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved

Bit	Attr	Reset Value	Description
1:0	RW	0x2	gpio2a0_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved

GRF_GPIO2B_P

Address: Operational Base + offset (0x1011C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RW	0x2	gpio2b7_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
13:12	RW	0x2	gpio2b6_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
11:10	RW	0x2	gpio2b5_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
9:8	RW	0x2	gpio2b4_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
7:6	RW	0x2	gpio2b3_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
5:4	RW	0x2	gpio2b2_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
3:2	RW	0x2	gpio2b1_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
1:0	RW	0x2	gpio2b0_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved

GRF_GPIO2C_P

Address: Operational Base + offset (0x10120)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RW	0x2	gpio2c7_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
13:12	RW	0x2	gpio2c6_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
11:10	RW	0x2	gpio2c5_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
9:8	RW	0x2	gpio2c4_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
7:6	RW	0x2	gpio2c3_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
5:4	RW	0x2	gpio2c2_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
3:2	RW	0x2	gpio2c1_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
1:0	RW	0x2	gpio2c0_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved

GRF_GPIO2D_P

Address: Operational Base + offset (0x10124)

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

Bit	Attr	Reset Value	Description
15:14	RW	0x2	gpio2d7_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
13:12	RW	0x2	gpio2d6_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
11:10	RW	0x2	gpio2d5_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
9:8	RW	0x2	gpio2d4_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
7:6	RW	0x2	gpio2d3_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
5:4	RW	0x2	gpio2d2_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
3:2	RW	0x2	gpio2d1_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
1:0	RW	0x2	gpio2d0_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved

GRF_GPIO3A_P

Address: Operational Base + offset (0x10128)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RW	0x2	gpio3a7_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved

Bit	Attr	Reset Value	Description
13:12	RW	0x2	gpio3a6_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
11:10	RW	0x2	gpio3a5_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
9:8	RW	0x2	gpio3a4_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
7:6	RW	0x1	gpio3a3_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
5:4	RW	0x1	gpio3a2_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
3:2	RW	0x1	gpio3a1_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
1:0	RW	0x1	gpio3a0_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved

GRF_GPIO3B_P

Address: Operational Base + offset (0x1012C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RW	0x2	gpio3b7_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
13:12	RW	0x2	gpio3b6_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved

Bit	Attr	Reset Value	Description
11:10	RW	0x2	gpio3b5_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
9:8	RW	0x2	gpio3b4_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
7:6	RW	0x2	gpio3b3_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
5:4	RW	0x2	gpio3b2_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
3:2	RW	0x2	gpio3b1_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
1:0	RW	0x2	gpio3b0_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved

GRF_GPIO3C_P

Address: Operational Base + offset (0x10130)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RW	0x2	gpio3c7_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
13:12	RW	0x2	gpio3c6_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
11:10	RW	0x2	gpio3c5_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved

Bit	Attr	Reset Value	Description
9:8	RW	0x2	gpio3c4_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
7:6	RW	0x2	gpio3c3_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
5:4	RW	0x2	gpio3c2_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
3:2	RW	0x2	gpio3c1_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
1:0	RW	0x2	gpio3c0_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved

GRF_GPIO3D_P

Address: Operational Base + offset (0x10134)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RW	0x2	gpio3d7_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
13:12	RW	0x2	gpio3d6_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
11:10	RW	0x2	gpio3d5_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
9:8	RW	0x2	gpio3d4_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved

Bit	Attr	Reset Value	Description
7:6	RW	0x2	gpio3d3_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
5:4	RW	0x2	gpio3d2_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
3:2	RW	0x2	gpio3d1_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
1:0	RW	0x2	gpio3d0_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved

GRF_GPIO4A_P

Address: Operational Base + offset (0x10138)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:4	RO	0x000	reserved
3:2	RW	0x2	gpio4a1_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
1:0	RW	0x2	gpio4a0_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved

GRF_GPIO0C_IE_H

Address: Operational Base + offset (0x10140)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x1	gpio0c7_ie 1'b0: Input disable 1'b1: Input enable
6	RW	0x1	gpio0c6_ie 1'b0: Input disable 1'b1: Input enable

Bit	Attr	Reset Value	Description
5	RW	0x1	gpio0c5_ie 1'b0: Input disable 1'b1: Input enable
4	RW	0x1	gpio0c4_ie 1'b0: Input disable 1'b1: Input enable
3:0	RO	0xf	reserved

GRF_GPIO0D_IE

Address: Operational Base + offset (0x10144)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x1	gpio0d7_ie 1'b0: Input disable 1'b1: Input enable
6	RW	0x1	gpio0d6_ie 1'b0: Input disable 1'b1: Input enable
5	RW	0x1	gpio0d5_ie 1'b0: Input disable 1'b1: Input enable
4	RW	0x1	gpio0d4_ie 1'b0: Input disable 1'b1: Input enable
3	RW	0x1	gpio0d3_ie 1'b0: Input disable 1'b1: Input enable
2	RW	0x1	gpio0d2_ie 1'b0: Input disable 1'b1: Input enable
1	RW	0x1	gpio0d1_ie 1'b0: Input disable 1'b1: Input enable
0	RW	0x1	gpio0d0_ie 1'b0: Input disable 1'b1: Input enable

GRF_GPIO1A_IE

Address: Operational Base + offset (0x10148)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x1	gpio1a7_ie 1'b0: Input disable 1'b1: Input enable
6	RW	0x1	gpio1a6_ie 1'b0: Input disable 1'b1: Input enable

Bit	Attr	Reset Value	Description
5	RW	0x1	gpio1a5_ie 1'b0: Input disable 1'b1: Input enable
4	RW	0x1	gpio1a4_ie 1'b0: Input disable 1'b1: Input enable
3	RW	0x1	gpio1a3_ie 1'b0: Input disable 1'b1: Input enable
2	RW	0x1	gpio1a2_ie 1'b0: Input disable 1'b1: Input enable
1	RW	0x1	gpio1a1_ie 1'b0: Input disable 1'b1: Input enable
0	RW	0x1	gpio1a0_ie 1'b0: Input disable 1'b1: Input enable

GRF_GPIO1B_IE

Address: Operational Base + offset (0x1014C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x1	gpio1b7_ie 1'b0: Input disable 1'b1: Input enable
6	RW	0x1	gpio1b6_ie 1'b0: Input disable 1'b1: Input enable
5	RW	0x1	gpio1b5_ie 1'b0: Input disable 1'b1: Input enable
4	RW	0x1	gpio1b4_ie 1'b0: Input disable 1'b1: Input enable
3	RW	0x1	gpio1b3_ie 1'b0: Input disable 1'b1: Input enable
2	RW	0x1	gpio1b2_ie 1'b0: Input disable 1'b1: Input enable
1	RW	0x1	gpio1b1_ie 1'b0: Input disable 1'b1: Input enable
0	RW	0x1	gpio1b0_ie 1'b0: Input disable 1'b1: Input enable

GRF_GPIO1C_IE

Address: Operational Base + offset (0x10150)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x1	gpio1c7_ie 1'b0: Input disable 1'b1: Input enable
6	RW	0x1	gpio1c6_ie 1'b0: Input disable 1'b1: Input enable
5	RW	0x1	gpio1c5_ie 1'b0: Input disable 1'b1: Input enable
4	RW	0x1	gpio1c4_ie 1'b0: Input disable 1'b1: Input enable
3	RW	0x1	gpio1c3_ie 1'b0: Input disable 1'b1: Input enable
2	RW	0x1	gpio1c2_ie 1'b0: Input disable 1'b1: Input enable
1	RW	0x1	gpio1c1_ie 1'b0: Input disable 1'b1: Input enable
0	RW	0x1	gpio1c0_ie 1'b0: Input disable 1'b1: Input enable

GRF_GPIO1D_IE

Address: Operational Base + offset (0x10154)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x1	gpio1d7_ie 1'b0: Input disable 1'b1: Input enable
6	RW	0x1	gpio1d6_ie 1'b0: Input disable 1'b1: Input enable
5	RW	0x1	gpio1d5_ie 1'b0: Input disable 1'b1: Input enable
4	RW	0x1	gpio1d4_ie 1'b0: Input disable 1'b1: Input enable
3	RW	0x1	gpio1d3_ie 1'b0: Input disable 1'b1: Input enable

Bit	Attr	Reset Value	Description
2	RW	0x1	gpio1d2_ie 1'b0: Input disable 1'b1: Input enable
1	RW	0x1	gpio1d1_ie 1'b0: Input disable 1'b1: Input enable
0	RW	0x1	gpio1d0_ie 1'b0: Input disable 1'b1: Input enable

GRF_GPIO2A_IE

Address: Operational Base + offset (0x10158)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x1	gpio2a7_ie 1'b0: Input disable 1'b1: Input enable
6	RW	0x1	gpio2a6_ie 1'b0: Input disable 1'b1: Input enable
5	RW	0x1	gpio2a5_ie 1'b0: Input disable 1'b1: Input enable
4	RW	0x1	gpio2a4_ie 1'b0: Input disable 1'b1: Input enable
3	RW	0x1	gpio2a3_ie 1'b0: Input disable 1'b1: Input enable
2	RW	0x1	gpio2a2_ie 1'b0: Input disable 1'b1: Input enable
1	RW	0x1	gpio2a1_ie 1'b0: Input disable 1'b1: Input enable
0	RW	0x1	gpio2a0_ie 1'b0: Input disable 1'b1: Input enable

GRF_GPIO2B_IE

Address: Operational Base + offset (0x1015C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x1	gpio2b7_ie 1'b0: Input disable 1'b1: Input enable

Bit	Attr	Reset Value	Description
6	RW	0x1	gpio2b6_ie 1'b0: Input disable 1'b1: Input enable
5	RW	0x1	gpio2b5_ie 1'b0: Input disable 1'b1: Input enable
4	RW	0x1	gpio2b4_ie 1'b0: Input disable 1'b1: Input enable
3	RW	0x1	gpio2b3_ie 1'b0: Input disable 1'b1: Input enable
2	RW	0x1	gpio2b2_ie 1'b0: Input disable 1'b1: Input enable
1	RW	0x1	gpio2b1_ie 1'b0: Input disable 1'b1: Input enable
0	RW	0x1	gpio2b0_ie 1'b0: Input disable 1'b1: Input enable

GRF_GPIO2C_IE

Address: Operational Base + offset (0x10160)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x1	gpio2c7_ie 1'b0: Input disable 1'b1: Input enable
6	RW	0x1	gpio2c6_ie 1'b0: Input disable 1'b1: Input enable
5	RW	0x1	gpio2c5_ie 1'b0: Input disable 1'b1: Input enable
4	RW	0x1	gpio2c4_ie 1'b0: Input disable 1'b1: Input enable
3	RW	0x1	gpio2c3_ie 1'b0: Input disable 1'b1: Input enable
2	RW	0x1	gpio2c2_ie 1'b0: Input disable 1'b1: Input enable
1	RW	0x1	gpio2c1_ie 1'b0: Input disable 1'b1: Input enable
0	RW	0x1	gpio2c0_ie 1'b0: Input disable 1'b1: Input enable

GRF_GPIO2D_IE

Address: Operational Base + offset (0x10164)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x1	gpio2d7_ie 1'b0: Input disable 1'b1: Input enable
6	RW	0x1	gpio2d6_ie 1'b0: Input disable 1'b1: Input enable
5	RW	0x1	gpio2d5_ie 1'b0: Input disable 1'b1: Input enable
4	RW	0x1	gpio2d4_ie 1'b0: Input disable 1'b1: Input enable
3	RW	0x1	gpio2d3_ie 1'b0: Input disable 1'b1: Input enable
2	RW	0x1	gpio2d2_ie 1'b0: Input disable 1'b1: Input enable
1	RW	0x1	gpio2d1_ie 1'b0: Input disable 1'b1: Input enable
0	RW	0x1	gpio2d0_ie 1'b0: Input disable 1'b1: Input enable

GRF_GPIO3A_IE

Address: Operational Base + offset (0x10168)

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

Bit	Attr	Reset Value	Description
2	RW	0x1	gpio3a2_ie 1'b0: Input disable 1'b1: Input enable
1	RW	0x1	gpio3a1_ie 1'b0: Input disable 1'b1: Input enable
0	RW	0x1	gpio3a0_ie 1'b0: Input disable 1'b1: Input enable

GRF_GPIO3B_IE

Address: Operational Base + offset (0x1016C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x1	gpio3b7_ie 1'b0: Input disable 1'b1: Input enable
6	RW	0x1	gpio3b6_ie 1'b0: Input disable 1'b1: Input enable
5	RW	0x1	gpio3b5_ie 1'b0: Input disable 1'b1: Input enable
4	RW	0x1	gpio3b4_ie 1'b0: Input disable 1'b1: Input enable
3	RW	0x1	gpio3b3_ie 1'b0: Input disable 1'b1: Input enable
2	RW	0x1	gpio3b2_ie 1'b0: Input disable 1'b1: Input enable
1	RW	0x1	gpio3b1_ie 1'b0: Input disable 1'b1: Input enable
0	RW	0x1	gpio3b0_ie 1'b0: Input disable 1'b1: Input enable

GRF_GPIO3C_IE

Address: Operational Base + offset (0x10170)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x1	gpio3c7_ie 1'b0: Input disable 1'b1: Input enable

Bit	Attr	Reset Value	Description
6	RW	0x1	gpio3c6_ie 1'b0: Input disable 1'b1: Input enable
5	RW	0x1	gpio3c5_ie 1'b0: Input disable 1'b1: Input enable
4	RW	0x1	gpio3c4_ie 1'b0: Input disable 1'b1: Input enable
3	RW	0x1	gpio3c3_ie 1'b0: Input disable 1'b1: Input enable
2	RW	0x1	gpio3c2_ie 1'b0: Input disable 1'b1: Input enable
1	RW	0x1	gpio3c1_ie 1'b0: Input disable 1'b1: Input enable
0	RW	0x1	gpio3c0_ie 1'b0: Input disable 1'b1: Input enable

GRF_GPIO3D_IE

Address: Operational Base + offset (0x10174)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x1	gpio3d7_ie 1'b0: Input disable 1'b1: Input enable
6	RW	0x1	gpio3d6_ie 1'b0: Input disable 1'b1: Input enable
5	RW	0x1	gpio3d5_ie 1'b0: Input disable 1'b1: Input enable
4	RW	0x1	gpio3d4_ie 1'b0: Input disable 1'b1: Input enable
3	RW	0x1	gpio3d3_ie 1'b0: Input disable 1'b1: Input enable
2	RW	0x1	gpio3d2_ie 1'b0: Input disable 1'b1: Input enable
1	RW	0x1	gpio3d1_ie 1'b0: Input disable 1'b1: Input enable
0	RW	0x1	gpio3d0_ie 1'b0: Input disable 1'b1: Input enable

GRF_GPIO4A_IE

Address: Operational Base + offset (0x10178)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:2	RO	0x3f	reserved
1	RW	0x1	gpio4a1_ie 1'b0: Input disable 1'b1: Input enable
0	RW	0x1	gpio4a0_ie 1'b0: Input disable 1'b1: Input enable

GRF_GPIO0C_SMT_H

Address: Operational Base + offset (0x10180)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x0	gpio0c7_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
6	RW	0x0	gpio0c6_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
5	RW	0x0	gpio0c5_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
4	RW	0x0	gpio0c4_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
3:0	RO	0x0	reserved

GRF_GPIO0D_SMT

Address: Operational Base + offset (0x10184)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x0	gpio0d7_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
6	RW	0x0	gpio0d6_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
5	RW	0x0	gpio0d5_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled

Bit	Attr	Reset Value	Description
4	RW	0x0	gpio0d4_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
3	RW	0x0	gpio0d3_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
2	RW	0x0	gpio0d2_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
1	RW	0x0	gpio0d1_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
0	RW	0x0	gpio0d0_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled

GRF_GPIO1A_SMT

Address: Operational Base + offset (0x10188)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x0	gpio1a7_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
6	RW	0x0	gpio1a6_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
5	RW	0x0	gpio1a5_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
4	RW	0x0	gpio1a4_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
3	RW	0x0	gpio1a3_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
2	RW	0x0	gpio1a2_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
1	RW	0x0	gpio1a1_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
0	RW	0x0	gpio1a0_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled

GRF_GPIO1B_SMT

Address: Operational Base + offset (0x1018C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x0	gpio1b7_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
6	RW	0x0	gpio1b6_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
5	RW	0x0	gpio1b5_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
4	RW	0x0	gpio1b4_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
3	RW	0x0	gpio1b3_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
2	RW	0x0	gpio1b2_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
1	RW	0x0	gpio1b1_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
0	RW	0x0	gpio1b0_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled

GRF_GPIO1C_SMT

Address: Operational Base + offset (0x10190)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x0	gpio1c7_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
6	RW	0x0	gpio1c6_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
5	RW	0x0	gpio1c5_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
4	RW	0x0	gpio1c4_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
3	RW	0x0	gpio1c3_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled

Bit	Attr	Reset Value	Description
2	RW	0x0	gpio1c2_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
1	RW	0x0	gpio1c1_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
0	RW	0x0	gpio1c0_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled

GRF_GPIO1D_SMT

Address: Operational Base + offset (0x10194)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x0	gpio1d7_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
6	RW	0x0	gpio1d6_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
5	RW	0x0	gpio1d5_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
4	RW	0x0	gpio1d4_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
3	RW	0x0	gpio1d3_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
2	RW	0x0	gpio1d2_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
1	RW	0x0	gpio1d1_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
0	RW	0x0	gpio1d0_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled

GRF_GPIO2A_SMT

Address: Operational Base + offset (0x10198)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x0	gpio2a7_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled

Bit	Attr	Reset Value	Description
6	RW	0x0	gpio2a6_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
5	RW	0x0	gpio2a5_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
4	RW	0x0	gpio2a4_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
3	RW	0x0	gpio2a3_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
2	RW	0x0	gpio2a2_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
1	RW	0x0	gpio2a1_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
0	RW	0x0	gpio2a0_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled

GRF_GPIO2B_SMT

Address: Operational Base + offset (0x1019C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x0	gpio2b7_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
6	RW	0x0	gpio2b6_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
5	RW	0x0	gpio2b5_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
4	RW	0x0	gpio2b4_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
3	RW	0x0	gpio2b3_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
2	RW	0x0	gpio2b2_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
1	RW	0x0	gpio2b1_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
0	RW	0x0	gpio2b0_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled

GRF_GPIO2C_SMT

Address: Operational Base + offset (0x101A0)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x0	gpio2c7_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
6	RW	0x0	gpio2c6_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
5	RW	0x0	gpio2c5_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
4	RW	0x0	gpio2c4_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
3	RW	0x0	gpio2c3_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
2	RW	0x0	gpio2c2_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
1	RW	0x0	gpio2c1_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
0	RW	0x0	gpio2c0_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled

GRF_GPIO2D_SMT

Address: Operational Base + offset (0x101A4)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x0	gpio2d7_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
6	RW	0x0	gpio2d6_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
5	RW	0x0	gpio2d5_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
4	RW	0x0	gpio2d4_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
3	RW	0x0	gpio2d3_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled

Bit	Attr	Reset Value	Description
2	RW	0x0	gpio2d2_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
1	RW	0x0	gpio2d1_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
0	RW	0x0	gpio2d0_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled

GRF_GPIO3A_SMT

Address: Operational Base + offset (0x101A8)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x0	gpio3a7_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
6	RW	0x0	gpio3a6_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
5	RW	0x0	gpio3a5_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
4	RW	0x0	gpio3a4_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
3	RW	0x0	gpio3a3_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
2	RW	0x0	gpio3a2_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
1	RW	0x0	gpio3a1_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
0	RW	0x0	gpio3a0_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled

GRF_GPIO3B_SMT

Address: Operational Base + offset (0x101AC)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x0	gpio3b7_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled

Bit	Attr	Reset Value	Description
6	RW	0x0	gpio3b6_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
5	RW	0x0	gpio3b5_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
4	RW	0x0	gpio3b4_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
3	RW	0x0	gpio3b3_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
2	RW	0x0	gpio3b2_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
1	RW	0x0	gpio3b1_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
0	RW	0x0	gpio3b0_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled

GRF_GPIO3C_SMT

Address: Operational Base + offset (0x101B0)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x0	gpio3c7_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
6	RW	0x0	gpio3c6_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
5	RW	0x0	gpio3c5_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
4	RW	0x0	gpio3c4_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
3	RW	0x0	gpio3c3_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
2	RW	0x0	gpio3c2_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
1	RW	0x0	gpio3c1_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
0	RW	0x0	gpio3c0_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled

GRF_GPIO3D_SMT

Address: Operational Base + offset (0x101B4)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x0	gpio3d7_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
6	RW	0x0	gpio3d6_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
5	RW	0x0	gpio3d5_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
4	RW	0x0	gpio3d4_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
3	RW	0x0	gpio3d3_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
2	RW	0x0	gpio3d2_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
1	RW	0x0	gpio3d1_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
0	RW	0x0	gpio3d0_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled

GRF_GPIO4A_SMT

Address: Operational Base + offset (0x101B8)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:2	RO	0x0000	reserved
1	RW	0x0	gpio4a1_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
0	RW	0x0	gpio4a0_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled

GRF_CSIPHY0_CON

Address: Operational Base + offset (0x10200)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:10	RO	0x00	reserved

Bit	Attr	Reset Value	Description
9	RW	0x0	csiphy0_clk_inv CSIPHY0 clock inverted. 1'b0: Disable 1'b1: Enable
8	RW	0x0	csiphy0_clklane_en CSIPHY0 clock lane enable. 1'b0: Disable 1'b1: Enable
7	RW	0x0	csiphy0_enable_dat_3 CSIPHY0 lane3 data enable. 1'b0: Disable 1'b1: Enable
6	RW	0x0	csiphy0_enable_dat_2 CSIPHY0 lane2 data enable. 1'b0: Disable 1'b1: Enable
5	RW	0x0	csiphy0_enable_dat_1 CSIPHY0 lane1 data enable. 1'b0: Disable 1'b1: Enable
4	RW	0x0	csiphy0_enable_dat_0 CSIPHY0 lane0 data enable. 1'b0: Disable 1'b1: Enable
3	RW	0x0	csiphy0_forcerxmode_3 CSIPHY0 lane3 force to RX mode. 1'b0: Disable 1'b1: Enable
2	RW	0x0	csiphy0_forcerxmode_2 CSIPHY0 lane2 force to RX mode. 1'b0: Disable 1'b1: Enable
1	RW	0x0	csiphy0_forcerxmode_1 CSIPHY0 lane1 force to RX mode. 1'b0: Disable 1'b1: Enable
0	RW	0x0	csiphy0_forcerxmode_0 CSIPHY0 lane0 force to RX mode. 1'b0: Disable 1'b1: Enable

GRF_CSIPHY0_STATUS

Address: Operational Base + offset (0x10208)

Bit	Attr	Reset Value	Description
31:11	RO	0x000000	reserved
10	RW	0x0	csiphy0_pin_errcontentionlp1_0 LP1 Contention error status
9	RW	0x0	csiphy0_pin_errcontentionlp0_0 LP0 Contention error status
8	RW	0x0	csiphy0_pin_rxskewcalhs_3 Lane3 high-speed receive skew calibration status
7	RW	0x0	csiphy0_pin_rxskewcalhs_2 Lane2 high-speed receive skew calibration status
6	RW	0x0	csiphy0_pin_rxskewcalhs_1 Lane1 high-speed receive skew calibration status

Bit	Attr	Reset Value	Description
5	RW	0x0	csiphy0_pin_rxskewcalhs_0 Lane0 high-speed receive skew calibration status
4	RW	0x0	csiphy0_direction Transmit/receive direction. 1'b0: Transmit mode 1'b1: Receive mode
3	RW	0x0	csiphy0_ulpsactivenot_3 Lane3 ULP active status. This active low signal is asserted to indicate that the lane is in ULP state.
2	RW	0x0	csiphy0_ulpsactivenot_2 Lane2 ULP active status. This active low signal is asserted to indicate that the lane is in ULP state.
1	RW	0x0	csiphy0_ulpsactivenot_1 Lane1 ULP active status. This active low signal is asserted to indicate that the lane is in ULP state.
0	RW	0x0	csiphy0_ulpsactivenot_0 Lane0 ULP active status. This active low signal is asserted to indicate that the lane is in ULP state.

GRF_CSIPHY1_CON

Address: Operational Base + offset (0x10210)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:10	RO	0x00	reserved
9	RW	0x0	csiphy_clk_inv CSIPHY1 clock inverted. 1'b0: Disable 1'b1: Enable
8	RW	0x0	csiphy_clklane_en CSIPHY1 clock lane enable. 1'b0: Disable 1'b1: Enable
7	RW	0x0	csiphy1_enable_dat_3 CSIPHY1 lane3 data enable. 1'b0: Disable 1'b1: Enable
6	RW	0x0	csiphy1_enable_dat_2 CSIPHY1 lane2 data enable. 1'b0: Disable 1'b1: Enable
5	RW	0x0	csiphy1_enable_dat_1 CSIPHY1 lane1 data enable. 1'b0: Disable 1'b1: Enable
4	RW	0x0	csiphy1_enable_dat_0 CSIPHY1 lane0 data enable. 1'b0: Disable 1'b1: Enable

Bit	Attr	Reset Value	Description
3	RW	0x0	csiphy1_forcerxmode_3 CSIPHY1 lane3 force to RX mode. 1'b0: Disable 1'b1: Enable
2	RW	0x0	csiphy1_forcerxmode_2 CSIPHY1 lane2 force to RX mode. 1'b0: Disable 1'b1: Enable
1	RW	0x0	csiphy1_forcerxmode_1 CSIPHY1 lane1 force to RX mode. 1'b0: Disable 1'b1: Enable
0	RW	0x0	csiphy1_forcerxmode_0 CSIPHY1 lane 0force to RX mode. 1'b0: Disable 1'b1: Enable

GRF_CSIPHY1_STATUS

Address: Operational Base + offset (0x10218)

Bit	Attr	Reset Value	Description
31:11	RO	0x000000	reserved
10	RW	0x0	csiphy1_pin_errcontentionlp1_0 LP1 Contention error status
9	RW	0x0	csiphy1_pin_errcontentionlp0_0 LP0 Contention error status
8	RW	0x0	csiphy1_pin_rxskewcalhs_3 Lane3 high-speed receive skew calibration status
7	RW	0x0	csiphy1_pin_rxskewcalhs_2 Lane2 high-speed receive skew calibration status
6	RW	0x0	csiphy1_pin_rxskewcalhs_1 Lane1 high-speed receive skew calibration status
5	RW	0x0	csiphy1_pin_rxskewcalhs_0 Lane0 high-speed receive skew calibration status
4	RW	0x0	csiphy1_direction Transmit/Receive direction. 1'b0: Transmit mode 1'b1: Receive mode
3	RW	0x0	csiphy1_ulpsactivenot_3 Lane3 ULP active status. This active low signal is asserted to indicate that the lane is in ULP state.
2	RW	0x0	csiphy1_ulpsactivenot_2 Lane2 ULP active status. This active low signal is asserted to indicate that the lane is in ULP state.
1	RW	0x0	csiphy1_ulpsactivenot_1 Lane1 ULP active status. This active low signal is asserted to indicate that the lane is in ULP state.
0	RW	0x0	csiphy1_ulpsactivenot_0 Lane0 ULP active status. This active low signal is asserted to indicate that the lane is in ULP state.

GRF_DSIPHY_CON

Address: Operational Base + offset (0x10220)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	dsiphy_txskewcalhs_3 Request to transmit skew calibration on lane3. 1'b0: Disable 1'b1: Enable
14	RW	0x0	dsiphy_txskewcalhs_2 Request to transmit skew calibration on lane2. 1'b0: Disable 1'b1: Enable
13	RW	0x0	dsiphy_txskewcalhs_1 Request to transmit skew calibration on lane1. 1'b0: Disable 1'b1: Enable
12	RW	0x0	dsiphy_txskewcalhs_0 Request to transmit skew calibration on lane0. 1'b0: Disable 1'b1: Enable
11	RW	0x0	dsiphy_txskewcalhs_ck Request to transmit skew calibration on clock lane. 1'b0: Disable 1'b1: Enable
10:8	RO	0x0	reserved
7	RW	0x0	dsiphy_lane3_frctxstpm Force DSIPHY lane3 into transmit mode and generate stop state. 1'b0: Disable 1'b1: Enable
6	RW	0x0	dsiphy_lane2_frctxstpm Force DSIPHY lane2 into transmit mode and generate stop state. 1'b0: Disable 1'b1: Enable
5	RW	0x0	dsiphy_lane1_frctxstpm Force DSIPHY lane1 into transmit mode and generate stop state. 1'b0: Disable 1'b1: Enable
4	RW	0x0	dsiphy_lane0_frctxstpm Force DSIPHY lane0 into transmit mode and generate stop state. 1'b0: Disable 1'b1: Enable
3	RO	0x0	reserved
2	RW	0x0	dsiphy_lane0_turndisable Disable Turn-around. This signal is used to prevent lane from going into transmit mode, even if it observes a turn-around request on the Lane interconnect
1	RW	0x0	dsiphy_lvds_mode DSIPHY LVDS mode enable. 1'b0: Disable 1'b1: Enable

Bit	Attr	Reset Value	Description
0	RW	0x0	dsiphy_forcerxmode Force DSIPHY TX lane module into receive mode, wait for stop state. 1'b0: Disable 1'b1: Enable

GRF_USBPHY_CON0

Address: Operational Base + offset (0x10230)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x1	usbotg_data_bus16_8 Software configure for OTGPHY data_bus16_8
14	RW	0x0	usbotg_pll_en Software configure for OTGPHY pll_en
13	RW	0x0	usbotg_dischrgvbus Software configure for OTGPHY dischrgvbus
12	RW	0x0	usbotg_chrgvbus Software configure for OTGPHY chrgvbus
11	RW	0x1	usbotg_idpulup Software configure for OTGPHY idpulup
10	RW	0x1	usbotg_utmi_iddig Software configure for OTGPHY utmi_iddig
9	RW	0x0	usbotg_utmi_iddig_sel IDDIG selection. 1'b0: USBPHY output 1'b1: Software output, USBPHY_CON0[10]
8	RW	0x0	usbotg_dmpulldown Software configure for OTGPHY dmpulldown
7	RW	0x0	usbotg_dppilldown Software configure for OTGPHY dppilldown
6	RW	0x1	usbotg_termselect Software configure for OTGPHY termselect
5:4	RW	0x1	usbotg_xcvrselect Software configure for OTGPHY xcvrselect
3:2	RW	0x0	usbotg_opmode Software configure for OTGPHY opmode
1	RW	0x1	usbotg_suspend_n Software configure for OTGPHY suspend_n
0	RW	0x0	otgphy_input_sel OTGPHY input selection. 1'b0: Hardware input 1'b1: Software input

GRF_USBPHY_CON1

Address: Operational Base + offset (0x10234)

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

Bit	Attr	Reset Value	Description
14	RW	0x0	usbotg_chg_en Software configure for OTGPHY chg_en
13	RW	0x0	usbhost_suspendm_byps Software configure for HOSTPHY suspendm_byps
12	RW	0x0	usbotg_suspendm_byps Software configure for OTGPHY suspendm_byps
11	RW	0x0	usbotg_txbitstuff_enable Software configure for OTGPHY txbitstuff_enable
10	RW	0x0	usbotg_suspendm Software configure for OTGPHY suspendm
9	RW	0x0	usbotg_fs_se0 Software configure for OTGPHY fs_se0
8	RW	0x0	usbotg_fs_data Software configure for OTGPHY fs_data
7	RW	0x1	usbotg_fs_oe Software configure for OTGPHY fs_oe
6	RW	0x0	usbotg_fs_xver_own Software configure for OTGPHY fs_xver_own
5:3	RO	0x0	reserved
2	RW	0x0	usbotg_suspend_n_1 Software configure for OTGPHY suspend_n
1:0	RW	0x0	usbotg_suspend_n_sel usbotg_suspend_n selection. 2'b00: ~usbotg_utmi_suspend_com_n & ~usbotg_utmi_l1_suspend_com_n 2'b01: USBPHY_CON1[2] 2'b10: ~usbotg_utmi_suspend_com_n & ~usbotg_utmi_l1_suspend_com_n 2'b11: ~usbotg_uspend_n & ~usbotg_utmi_l1_suspend_n

GRF_USBPHY_CON2

Address: Operational Base + offset (0x10238)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	usbhost_id_pull_up Software configure for HOSTPHY id_pull_up
14	RW	0x0	usbhost_dischrgvbus Software configure for HOSTPHY dischrgvbus
13	RW	0x0	usbhost_chrgvbus Software configure for HOSTPHY chrgvbus
12	RW	0x0	usbhost_txbitstuff_enable Software configure for HOSTPHY txbitstuff_enable
11	RW	0x1	usbhost_otg_suspendm Software configure for HOSTPHY suspendm
10	RW	0x1	usbhost_data_bus16_8 Software configure for HOSTPHY data_bus16_8
9	RW	0x0	usbhost_pll_en Software configure for HOSTPHY pll_en
8	RW	0x1	usbhost_chg_rst Software configure for HOSTPHY chg_rst
7	RW	0x0	usbhost_chg_en Software configure for HOSTPHY chg_en

Bit	Attr	Reset Value	Description
6	RW	0x1	usbhost_termselect Software configure for HOSTPHY termselect
5:4	RW	0x1	usbhost_xcvrselect Software configure for HOSTPHY xcvrselect
3:2	RW	0x0	usbhost_opmode Software configure for HOSTPHY opmode
1	RW	0x0	usbhost_suspend_n Software configure for HOSTPHY suspend_n
0	RW	0x0	hostphy_input_sel HOSTPHY input selection. 1'b0: Hardware input 1'b1: Software input

GRF_USBPHY_STATUS

Address: Operational Base + offset (0x10248)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29	RW	0x1	usbhost_phy_connect HOSTPHY start handshake. 1'b0: PHY handshake is prohibited 1'b1: PHY handshake is allowed
28	RW	0x0	usbhost_chg_valid HOSTPHY charge valid. 1'b0: Not charged 1'b1: Charged
27	RW	0x0	usbhost_phy_ls_fs_rcv HOSTPHY ls_fs_rcv status
26	RW	0x0	usbhost_avalid HOSTPHY avalid status
25	RW	0x0	usbhost_bvalid HOSTPHY bvalid status
24	RO	0x0	reserved
23	RW	0x0	usbhost_hostdisconnect HOSTPHY hostdisconnect status
22	RW	0x0	usbhost_utmi_iddig HOSTPHY utmi_iddig status
21:20	RW	0x0	usbhost_linestate HOSTPHY linestate status
19	RW	0x1	usbhost_sessend HOSTPHY sessend status
18	RW	0x0	usbhost_vbusvalid HOSTPHY vbusvalid status
17	RW	0x0	usbhost_vmi HOSTPHY vmi status
16	RW	0x0	usbhost_vpi HOSTPHY vpi status
15:14	RO	0x0	reserved
13	RW	0x1	usbotg_phy_connect OTGPHY start handshake. 1'b0: PHY handshake is prohibited 1'b1: PHY handshake is allowed
12	RW	0x0	usbotg_chg_valid OTGPHY charge valid. 1'b0: Not charged 1'b1: Charged

Bit	Attr	Reset Value	Description
11	RW	0x0	usbotg_phy_ls_fs_rcv OTGPHY phy_ls_fs_rcv status
10	RW	0x0	usbotg_avalid OTGPHY avalid status
9	RW	0x0	usbotg_bvalid OTGPHY bvalid status
8	RW	0x0	usbotg_fs_xver_own OTGPHY fs_xver_own status
7	RW	0x0	usbotg_hostdisconnect OTGPHY hostdisconnect status
6	RW	0x1	usbotg_utmi_iddig OTGPHY utmi_iddig status
5:4	RW	0x0	usbotg_linestate OTGPHY linestate status
3	RW	0x1	usbotg_sessend OTGPHY sessend status
2	RW	0x0	usbotg_vbusvalid OTGPHY vbusvalid status
1	RW	0x0	usbotg_vmi OTGPHY vmi status
0	RW	0x0	usbotg_vpi OTGPHY vpi status

GRF_CIFIO_CON

Address: Operational Base + offset (0x10250)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:11	RO	0x00	reserved
10	RW	0x0	cif_clk_inv_sel cif_clk inverted enable. 1'b0: Not inverted 1'b1: Inverted
9	RW	0x0	cif_clk_delay_en cif_clk delay enable. 1'b0: cif_clk bypassed 1'b1: cif_clk delayed by delayine
8	RW	0x0	cif_datapath Pixel input data path selection for CIF. 1'b0: Single edge sampling 1'b1: Double edge sampling
7:0	RW	0x00	cif_clk_delay_num cif_clk delayline number

GRF_SD_JTAG_SWITCH_DLY_CNT

Address: Operational Base + offset (0x10254)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sd_jtag_switch_dly_counter JTAG switch delay counter. Count by 24MHz clock.

GRF_UART2RX_LOW_CON

Address: Operational Base + offset (0x10258)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	uart2rx_low_thres UART2 RX pin low filter threshold. Count by 24MHz clock.

GRF_IOFUNC_CON0

Address: Operational Base + offset (0x10260)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	i2s0_sdi3_oen I2S0 sdi3 output enable. 1'b0: Output enable 1'b1: Output disable
14	RW	0x0	i2s0_sdi2_oen I2S0 sdi2 output enable. 1'b0: Output enable 1'b1: Output disable
13	RW	0x0	i2s0_sdi1_oen I2S0 sdi1 output enable. 1'b0: Output enable 1'b1: Output disable
12	RW	0x0	pdm_iomux_sel PDM IOMUX selection. 1'b0: IOMUX group 0 (PDM*M0) selected 1'b1: IOMUX group 1 (PDM*M1) selected
11	RW	0x0	i2s2_lrck_sel I2S2 lrck selection. 1'b0: i2s2_lrck_rx selected 1'b1: i2s2_lrck_tx selected
10	RW	0x0	i2s1_lrck_sel I2S1 lrck selection. 1'b0: i2s1_lrck_rx selected 1'b1: i2s1_lrck_tx selected
9	RW	0x0	i2s0_mclk_sel I2S0 mclk selection. 1'b0: i2s0_mclk_tx selected 1'b1: i2s0_mclk_rx selected
8:5	RO	0x0	reserved
4	RW	0x0	i2s2_iomux_sel I2S2 IOMUX selection. 1'b0: IOMUX group 0 (I2S2*M0) selected 1'b1: IOMUX group 1 (I2S2*M1) selected
3:2	RW	0x0	i2s1_iomux_sel I2S1 IOMUX selection. 2'b00: IOMUX group 0 (I2S1*M0) selected 2'b01: IOMUX group 1 (I2S1*M1) selected 2'b10: IOMUX group 2 (I2S1*M2) selected 2'b11: Reserved
1	RO	0x0	reserved
0	RW	0x0	i2s0_iomux_sel I2S0 IOMUX selection. 1'b0: IOMUX group 0 (I2S0*M0) selected 1'b1: IOMUX group 1 (I2S0*M1) selected

GRF_IOFUNC_CON1

Address: Operational Base + offset (0x10264)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	jtag_iomux_sel A7 JTAG IOMUX selection. 1'b0: IOMUX group 0 (A7JTAG*M0) selected 1'b1: IOMUX group 1 (A7JTAG*M1) selected
14	RO	0x0	reserved
13	RW	0x0	can_iomux_sel CAN IOMUX selection. 1'b0: IOMUX group 0 (CAN*M0) selected 1'b1: IOMUX group 1 (CAN*M1) selected
12	RW	0x0	rgmii_iomux_sel RGMII IOMUX selection. 1'b0: IOMUX group 0 (RGMII*M0) selected 1'b1: IOMUX group 1 (RGMII*M1) selected
11:10	RW	0x0	spi1_iomux_sel SPI1 IOMUX selection. 2'b00: IOMUX group 0 (SPI1*M0) selected 2'b01: IOMUX group 1 (SPI1*M1) selected 2'b10: IOMUX group 2 (SPI1*M2) selected 2'b11: Reserved
9:8	RW	0x0	i2c5_iomux_sel I2C5 IOMUX selection. 2'b00: IOMUX group 0 (I2C5*M0) selected 2'b01: IOMUX group 1 (I2C5*M1) selected 2'b10: IOMUX group 2 (I2C5*M2) selected 2'b11: Reserved
7	RO	0x0	reserved
6	RW	0x0	i2c4_iomux_sel I2C4 IOMUX selection. 1'b0: IOMUX group 0 (I2C4*M0) selected 1'b1: IOMUX group 1 (I2C4*M1) selected
5:4	RW	0x0	i2c3_iomux_sel I2C3 IOMUX selection. 2'b00: IOMUX group 0 (I2C3*M0) selected 2'b01: IOMUX group 1 (I2C3*M1) selected 2'b10: IOMUX group 2 (I2C3*M2) selected 2'b11: Reserved
3:1	RO	0x0	reserved
0	RW	0x0	cif_iomux_sel CIF IOMUX selection. 1'b0: IOMUX group 0 (CIF*M0) selected 1'b1: IOMUX group 1 (CIF*M1) selected

GRF_IOFUNC_CON2

Address: Operational Base + offset (0x10268)

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

Bit	Attr	Reset Value	Description
15:14	RW	0x0	uart5_iomux_sel UART5 IOMUX selection. 2'b00: IOMUX group 0 (UART5*M0) selected 2'b01: IOMUX group 1 (UART5*M1) selected 2'b10: IOMUX group 2 (UART5*M2) selected 2'b11: Reserved
13:12	RW	0x0	uart4_iomux_sel UART4 IOMUX selection. 2'b00: IOMUX group 0 (UART4*M0) selected 2'b01: IOMUX group 1 (UART4*M1) selected 2'b10: IOMUX group 2 (UART4*M2) selected 2'b11: Reserved
11:10	RW	0x0	uart3_iomux_sel UART3 IOMUX selection. 2'b00: IOMUX group 0 (UART3*M0) selected 2'b01: IOMUX group 1 (UART3*M1) selected 2'b10: IOMUX group 2 (UART3*M2) selected 2'b11: Reserved
9	RO	0x0	reserved
8	RW	0x0	uart2_iomux_sel UART2 IOMUX selection. 1'b0: IOMUX group 0 (UART2*M0) selected 1'b1: IOMUX group 1 (UART2*M1) selected
7	RO	0x0	reserved
6	RW	0x0	pwm11_iomux_sel PWM11 IOMUX selection. 1'b0: IOMUX group 0 (PWM11*M0) selected 1'b1: IOMUX group 1 (PWM11*M1) selected
5	RO	0x0	reserved
4	RW	0x0	pwm10_iomux_sel PWM10 IOMUX selection. 1'b0: IOMUX group 0 (PWM10*M0) selected 1'b1: IOMUX group 1 (PWM10*M1) selected
3	RO	0x0	reserved
2	RW	0x0	pwm9_iomux_sel PWM9 IOMUX selection. 1'b0: IOMUX group 0 (PWM9*M0) selected 1'b1: IOMUX group 1 (PWM9*M1) selected
1	RO	0x0	reserved
0	RW	0x0	pwm8_iomux_sel PWM8 IOMUX selection. 1'b0: IOMUX group 0 (PWM8*M0) selected 1'b1: IOMUX group 1 (PWM8*M1) selected

GRF_IOFUNC_CON3

Address: Operational Base + offset (0x1026C)

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

Bit	Attr	Reset Value	Description
15	RW	0x0	i2s2_mclk_ioe_ i2s2_mclk_ioe_ I2S2 mclk_tx output enable. 1'b0: Disable 1'b1: Enable
14	RW	0x0	i2s1_mclk_ioe_ i2s1_mclk_ioe_ I2S1 mclk output enable. 1'b0: Disable 1'b1: Enable
13	RW	0x0	i2s0_mclk_rx_ioe_ i2s0_mclk_rx_ioe_ I2S0 mclk_rx output enable. 1'b0: Disable 1'b1: Enable
12	RW	0x0	i2s0_mclk_tx_ioe_ i2s0_mclk_tx_ioe_ I2S0 mclk_tx output enable. 1'b0: Disable 1'b1: Enable
11	RW	0x0	ciflite_csiphy_sel CSIPHY source selection for CIFLITE. 1'b0: From CSIPHY0 1'b1: From CSIPHY1
10	RW	0x0	isp_csiphy_sel CSIPHY source selection for ISP. 1'b0: From CSIPHY0 1'b1: From CSIPHY1
9	RW	0x0	cif_csiphy_sel CSIPHY source selection for CIF. 1'b0: From CSIPHY0 1'b1: From CSIPHY1
8:6	RO	0x0	reserved
5	RW	0x0	force_jtag_m1 Force IO to JTAG IOMUX group 1 enable. 1'b0: Disable 1'b1: Enable
4	RW	0x1	force_jtag Force IO to JTAG IOMUX group 0 enable. 1'b0: Disable 1'b1: Enable
3	RO	0x0	reserved
2	RW	0x0	lcdc_dclk_inv_sel lcdc_dclk output inverted enable. 1'b0: Disable 1'b1: Enable
1	RO	0x0	reserved
0	RW	0x0	lcdc_bypass LCDC IO output path selection. 1'b0: IO through flip-flop before output 1'b1: IO bypass flip-flop before output

GRF_USBPHY0_CFG_CON

Address: Operational Base + offset (0x10270)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:3	RO	0x0000	reserved
2	RW	0x1	sel_apb APB selection input
1	RW	0x0	cfg_rden APB read enable input
0	RW	0x0	cfg_wren APB write enable input

GRF_USBPHY0_CFG_ADDRIN

Address: Operational Base + offset (0x10274)

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

GRF_USBPHY0_CFG_ADDROUT

Address: Operational Base + offset (0x10278)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15:0	RW	0x0000	cfg_addr APB configuration address status

GRF_USBPHY0_CFG_DLY_CON

Address: Operational Base + offset (0x1027C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7:4	RW	0x0	prdy_cnt APB pready counter
3:0	RW	0x0	rden_cnt APB read enable counter

GRF_USBPHY1_CFG_CON

Address: Operational Base + offset (0x10280)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:3	RO	0x0000	reserved
2	RW	0x1	sel_apb APB selection input
1	RW	0x0	cfg_rden APB read enable input

Bit	Attr	Reset Value	Description
0	RW	0x0	cfg_wren APB write enable input

GRF_USBPHY1_CFG_ADDRIN

Address: Operational Base + offset (0x10284)

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

GRF_USBPHY1_CFG_ADDROUT

Address: Operational Base + offset (0x10288)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15:0	RW	0x0000	cfg_addr APB configuration address status

GRF_USBPHY1_CFG_DLY_CON

Address: Operational Base + offset (0x1028C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7:4	RW	0x0	prdy_cnt APB pready counter
3:0	RW	0x0	rden_cnt APB read enable counter

GRF_USB_SIG_DETECT_CON

Address: Operational Base + offset (0x10300)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:10	RO	0x00	reserved
9	RW	0x0	host_disconnect_fall_irq_en host_disconnect falling edge status irq enable. 1'b0: Disable 1'b1: Enable
8	RW	0x0	host_disconnect_rise_irq_en host_disconnect rising edge status irq enable. 1'b0: Disable 1'b1: Enable
7	RW	0x0	otg_disconnect_fall_irq_en otg_disconnect falling edge status irq enable. 1'b0: Disable 1'b1: Enable

Bit	Attr	Reset Value	Description
6	RW	0x0	otg_disconnect_rise_irq_en otg_disconnect rising edge status irq enable. 1'b0: Disable 1'b1: Enable
5	RW	0x0	otg_id_fall_irq_en otg_id falling edge status irq enable. 1'b0: Disable 1'b1: Enable
4	RW	0x0	otg_id_rise_irq_en otg_id rising edge status irq enable. 1'b0: Disable 1'b1: Enable
3	RW	0x0	otg_bvalid_fall_irq_en otg_bvalid falling edge status irq enable. 1'b0: Disable 1'b1: Enable
2	RW	0x0	otg_bvalid_rise_irq_en otg_bvalid rising edge status irq enable. 1'b0: Disable 1'b1: Enable
1	RW	0x0	host_linestate_irq_en host_linestate change status irq enable. 1'b0: Disable 1'b1: Enable
0	RW	0x0	otg_linestate_irq_en otg_linestate change status irq enable. 1'b0: Disable 1'b1: Enable

GRF_USB_SIG_DETECT_STATUS

Address: Operational Base + offset (0x10304)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:10	RO	0x00	reserved
9	RW	0x0	host_disconnect_fall_irq host_disconnect falling edge irq status
8	RW	0x0	host_disconnect_rise_irq host_disconnect rising edge irq status
7	RW	0x0	otg_disconnect_fall_irq otg_disconnect falling edge irq status
6	RW	0x0	otg_disconnect_rise_irq otg_disconnect rising edge irq status
5	RW	0x0	otg_id_fall_irq otg_id falling edge irq status
4	RW	0x0	otg_id_rise_irq otg_id rising edge irq status
3	RW	0x0	otg_bvalid_fall_irq otg_bvalid falling edge irq status
2	RW	0x0	otg_bvalid_rise_irq otg_bvalid rising edge irq status
1	RW	0x0	host_linestate_irq host_linestate change irq status

Bit	Attr	Reset Value	Description
0	RW	0x0	otg_linestate_irq otg_linestate change irq status

GRF_USB_SIG_DETECT_CLR

Address: Operational Base + offset (0x10308)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:10	RO	0x00	reserved
9	RW	0x0	host_disconnect_fall_irq_clr host_disconnect falling edge status irq enable. 1'b0: Disable 1'b1: Enable
8	RW	0x0	host_disconnect_rise_irq_clr host_disconnect rising edge status irq clear. Write 1 to clear irq status.
7	RW	0x0	otg_disconnect_fall_irq_clr otg_disconnect falling edge status irq clear. Write 1 to clear irq status.
6	RW	0x0	otg_disconnect_rise_irq_clr otg_disconnect rising edge status irq clear. Write 1 to clear irq status.
5	RW	0x0	otg_id_fall_irq_clr otg_id falling edge status irq clear. Write 1 to clear irq status.
4	RW	0x0	otg_id_rise_irq_clr otg_id rising edge status irq clear. Write 1 to clear irq status.
3	RW	0x0	otg_bvalid_fall_irq_clr otg_bvalid falling edge status irq clear. Write 1 to clear irq status.
2	RW	0x0	otg_bvalid_rise_irq_clr otg_bvalid rising edge status irq clear. Write 1 to clear irq status.
1	RW	0x0	host_linestate_irq_clr host_linestate change status irq clear. Write 1 to clear irq status.
0	RW	0x0	otg_linestate_irq_clr otg_linestate change status irq clear. Write 1 to clear irq status.

GRF_USB_LINESTATE_CON

Address: Operational Base + offset (0x10310)

Bit	Attr	Reset Value	Description
31:20	RO	0x000	reserved
19:0	RW	0x30100	linestate_detect_con HOST/OTG port linestate filter time control register

GRF_USB_DISCONNECT_CON

Address: Operational Base + offset (0x10314)

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

GRF_USB_BVALID_CON

Address: Operational Base + offset (0x10318)

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

GRF_USB_ID_CON

Address: Operational Base + offset (0x1031C)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:0	RW	0x0030100	grf_id_detect_con OTG ID port filter time control register, the unit is pclk with up to 100MHz.

6.4 DDRGRF Register Description

6.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
DDRGRF_DDR_CON0	0x0000	W	0x00000000	DDR control register 0
DDRGRF_DDR_CON1	0x0004	W	0x00000600	DDR control register 1
DDRGRF_DDR_CON2	0x0008	W	0x00000000	DDR control register 2
DDRGRF_DDR_SPLIT_CON	0x0010	W	0x00000110	DDR SPLIT control register
DDRGRF_DDR_LP_CON	0x0020	W	0x00001101	DDR lower power control register
DDRGRF_DDRPHY_CON0	0x0040	W	0x00000000	DDRPHY control register 0
DDRGRF_DDRPHY_CON1	0x0044	W	0x00000000	DDRPHY control register 1
DDRGRF_DDRPHY_CON2	0x0048	W	0x00000000	DDRPHY control register 2
DDRGRF_DDRPHY_CON3	0x004c	W	0x00000000	DDRPHY control register 3
DDRGRF_DDRPHY_CON4	0x0050	W	0x00000000	DDRPHY control register 4
DDRGRF_DDRPHY_CON5	0x0054	W	0x00000000	DDRPHY control register 5
DDRGRF_DDR_STATUS0	0x0100	W	0x00000000	DDR status register 0
DDRGRF_DDR_STATUS1	0x0104	W	0x00000000	DDR status register 1
DDRGRF_DDR_STATUS2	0x0108	W	0x00000000	DDR status register 2
DDRGRF_DDR_STATUS3	0x010c	W	0x00000000	DDR status register 3
DDRGRF_DDR_STATUS4	0x0110	W	0x00000000	DDR status register 4
DDRGRF_DDR_STATUS5	0x0114	W	0x00000000	DDR status register 5
DDRGRF_DDR_STATUS6	0x0118	W	0x00000000	DDR status register 6
DDRGRF_DDR_STATUS7	0x011c	W	0x00000000	DDR status register 7
DDRGRF_DDR_STATUS8	0x0120	W	0x00000017	DDR status register 8
DDRGRF_DDR_STATUS9	0x0124	W	0x80000000	DDR status register 9
DDRGRF_DDR_STATUS10	0x0128	W	0x00000000	DDR status register 10
DDRGRF_DDR_STATUS11	0x012c	W	0x00000000	DDR status register 11
DDRGRF_DDR_STATUS12	0x0130	W	0x00000000	DDR status register 12
DDRGRF_DDR_STATUS13	0x0134	W	0x00000000	DDR status register 13
DDRGRF_DDR_STATUS14	0x0138	W	0x00000000	DDR status register 14
DDRGRF_DDR_STATUS15	0x013c	W	0x00000000	DDR status register 15
DDRGRF_DDR_STATUS16	0x0140	W	0x00000000	DDR status register 16
DDRGRF_DDR_STATUS17	0x0144	W	0x00000000	DDR status register 17
DDRGRF_DDRHOLD_STATUS	0x0150	W	0x01010140	DDR hold status register
DDRGRF_DDRPHY_STATUS0	0x0160	W	0x3fffffff	DDRPHY status register 0
DDRGRF_DDRPHY_STATUS1	0x0164	W	0xffffffff	DDRPHY status register 1

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

6.4.2 Detail Register Description

DDRGRF_DDR_CON0

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	hold_ddr_trans DDR tansfer hold. 1'b0: Disable 1'b1: Enable
14	RW	0x0	awpoison AXI write poison control. 1'b0: Disable 1'b1: Enable
13	RW	0x0	awurgent AXI write urgent control. 1'b0: Disable 1'b1: Enable
12	RW	0x0	arpoison AXI read poison control. 1'b0: Disable 1'b1: Enable
11	RW	0x0	arurgent AXI bus read urgent control. 1'b0: Disable 1'b1: Enable
10	RW	0x0	pa_wmask When asserted(active high), it will prevent the corresponding write to PA. 1'b0: Disable 1'b1: Enable
9:8	RW	0x0	pa_rmask When asserted(active high), it will prevent the corresponding read to PA. 1'b0: Disable 1'b1: Enable
7:6	RO	0x0	reserved
5	RW	0x0	csysreq_upctl_ddrstdby STDBY controls UPCTL csysreq_ddrc. 1'b0: Disable 1'b1: Enable
4	RW	0x0	csysreq_upctl_pmu PMU controls UPCTL csysreq_ddrc. 1'b0: Disable 1'b1: Enable
3	RW	0x0	csysreq_aclk 1'b0: Request UPCTL aclk enter low power 1'b1: Request UPCTL aclk exit low power
2	RW	0x0	dfi_init_start Software for dfi_init_start. 1'b0: Disable 1'b1: Enable
1	RW	0x0	dfi_init_start_sel dfi_init_start selection. 1'b0: From UPCTL 1'b1: From GRF_DDR_CON0[2]

Bit	Attr	Reset Value	Description
0	RW	0x0	upctl_slverr_enable UPCTL APB slave error response enable. 1'b0: Disable 1'b1: Enable

DDRGRF_DDR_CON1

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RO	0x0	reserved
11:8	RW	0x6	auto_sr_dly The delay of auto gated UPCTL core_clk. It should be to be 0x6.
7:6	RO	0x0	reserved
5	RW	0x0	con_upctl2_pdsrlp_cg_en UPCTL clock gating after enter PD/SR state. 1'b0: Disable 1'b1: Enable
4	RW	0x0	upctl_syscreq_cg_en UPCTL core_clk gating when external ddrc_csysreq asserted. 1'b0: Disable 1'b1: Enable
3	RW	0x0	selfref_type2_en UPCTL core_clk auto gating in type2 self-refresh. 1'b0: Disable 1'b1: Enable
2	RW	0x0	upctl_core_cg_en UPCTL core_clk auto gating. 1'b0: Disable 1'b1: Enable
1	RW	0x0	upctl_apb_cg_en UPCTL pclk auto gating. 1'b0: Disable 1'b1: Enable
0	RW	0x0	upctl_axi_cg_en UPCTL aclk auto gating. 1'b0: Disable 1'b1: Enable

DDRGRF_DDR_CON2

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:11	RO	0x00	reserved
10	RW	0x0	dif_freq_change_ack DFI frequence change acknowledge. 1'b0: Disable 1'b1: Enable

Bit	Attr	Reset Value	Description
9:0	RW	0x3ff	grf_con_ddr_clk_gate Clock gating control for UPCTL, should be set to 0x3ff for DDR normal work.

DDRGRF_DDR_SPLIT_CON

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:11	RO	0x00	reserved
10:9	RW	0x0	spmode Split mode select. 2'b00: DDR controller and PHY works at 32 bits mode. Low 16 bits are valid if access address is above split address. 2'b01: DDR controller and PHY works at 32 bits mode. High 16 bits are valid if access address is above split address. 2'b10: DDR controller and PHY works at 16 bits mode. Low 8 bits are valid if access address is above split address. 2'b11: DDR controller and PHY works at 16 bits mode. High 8 bits are valid if access address is above split address.
8	RW	0x1	bypass AXI split module bypass. 1'b0: Enable 1'b1: Bypass
7:0	RW	0x0a	spaddr Split address high 8 bits of 32bit address. For example, if spaddr=0x10, then the split address is 0x10000000. The axi_split module will be bypassed if reading or writing DDR below split address, otherwise AXI burst will be split.

DDRGRF_DDR_LP_CON

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13	RW	0x0	sr_ctl_en sr exit/enter reload/inverse lpckdis_ini. 1'b0: Disable 1'b1: Enable
12	RW	0x1	pd_ctl_en pd exit/enter reload/inverse lpckdis_ini. 1'b0: Disable 1'b1: Enable
11:10	RO	0x0	reserved
9	RW	0x0	lpckdis_en dfi_lp_ck_disable enable. 1'b0: Disable 1'b1: Enable
8	RW	0x1	lpckdis_ini dfi_lp_ck_disable intial value

Bit	Attr	Reset Value	Description
7:3	RO	0x00	reserved
2	RW	0x0	lp23_mode LPDDR2/LPDDR3 mode enable. 1'b0: Disable 1'b1: Enable
1	RO	0x0	reserved
0	RW	0x1	ddr23_mode DDR2/DDR3 mode enable. 1'b0: Disable 1'b1: Enable

DDRGRF_DDRPHY_CON0

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31	RW	0x0	gpio_a_dq3_txen DDRPHY's gpio_a_dq3_txen source
30	RW	0x0	gpio_a_dq2_txen DDRPHY's gpio_a_dq2_txen source
29	RW	0x0	gpio_a_dq1_txen DDRPHY's gpio_a_dq1_txen source
28	RW	0x0	gpio_a_dq0_txen DDRPHY's gpio_a_dq0_txen source
27	RW	0x0	gpio_actn_txen DDRPHY's gpio_actn_txen source
26	RW	0x0	gpio_odt1_txen DDRPHY's gpio_odt1_txen source
25	RW	0x0	gpio_odt0_txen DDRPHY's gpio_odt0_txen source
24	RW	0x0	gpio_csb1_txen DDRPHY's gpio_csb1_txen source
23	RW	0x0	gpio_csb0_txen DDRPHY's gpio_csb0_txen source
22	RW	0x0	gpio_ckb_txen DDRPHY's gpio_ckb_txen source
21	RW	0x0	gpio_ck_txen DDRPHY's gpio_ck_txen source
20	RW	0x0	gpio_bg1_txen DDRPHY's gpio_bg1_txen source
19	RW	0x0	gpio_bg0_txen DDRPHY's gpio_bg0_txen source
18	RW	0x0	gpio_ba1_txen DDRPHY's gpio_ba1_txen source
17	RW	0x0	gpio_a17_txen DDRPHY's gpio_a17_txen source
16	RW	0x0	gpio_a16_txen DDRPHY's gpio_a16_txen source
15	RW	0x0	gpio_a15_txen DDRPHY's gpio_a15_txen source
14	RW	0x0	gpio_a14_txen DDRPHY's gpio_a14_txen source
13	RW	0x0	gpio_a13_txen DDRPHY's gpio_a13_txen source
12	RW	0x0	gpio_a12_txen DDRPHY's gpio_a12_txen source

Bit	Attr	Reset Value	Description
11	RW	0x0	gpio_a11_txen DDRPHY's gpio_a11_txen source
10	RW	0x0	gpio_a10_txen DDRPHY's gpio_a10_txen source
9	RW	0x0	gpio_a9_txen DDRPHY's gpio_a9_txen source
8	RW	0x0	gpio_a8_txen DDRPHY's gpio_a8_txen source
7	RW	0x0	gpio_a7_txen DDRPHY's gpio_a7_txen source
6	RW	0x0	gpio_a6_txen DDRPHY's gpio_a6_txen source
5	RW	0x0	gpio_a5_txen DDRPHY's gpio_a5_txen source
4	RW	0x0	gpio_a4_txen DDRPHY's gpio_a4_txen source
3	RW	0x0	gpio_a3_txen DDRPHY's gpio_a3_txen source
2	RW	0x0	gpio_a2_txen DDRPHY's gpio_a2_txen source
1	RW	0x0	gpio_a1_txen DDRPHY's gpio_a1_txen source
0	RW	0x0	gpio_a0_txen DDRPHY's gpio_a0_txen source

DDRGRF_DDRPHY_CON1

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29	RW	0x0	gpio_resetn_txen DDRPHY's gpio_resetn_txen source
28	RW	0x0	gpio_cke0_txen DDRPHY's gpio_cke0_txen source
27	RW	0x0	gpio_b_dq15_txen DDRPHY's gpio_b_dq15_txen source
26	RW	0x0	gpio_b_dq14_txen DDRPHY's gpio_b_dq14_txen source
25	RW	0x0	gpio_b_dq13_txen DDRPHY's gpio_b_dq13_txen source
24	RW	0x0	gpio_b_dq12_txen DDRPHY's gpio_b_dq12_txen source
23	RW	0x0	gpio_b_dq11_txen DDRPHY's gpio_b_dq11_txen source
22	RW	0x0	gpio_b_dq10_txen DDRPHY's gpio_b_dq10_txen source
21	RW	0x0	gpio_b_dq9_txen DDRPHY's gpio_b_dq9_txen source
20	RW	0x0	gpio_b_dq8_txen DDRPHY's gpio_b_dq8_txen source
19	RW	0x0	gpio_b_dq7_txen DDRPHY's gpio_b_dq7_txen source
18	RW	0x0	gpio_b_dq6_txen DDRPHY's gpio_b_dq6_txen source

Bit	Attr	Reset Value	Description
17	RW	0x0	gpio_b_dq5_txen DDRPHY's gpio_b_dq5_txen source
16	RW	0x0	gpio_b_dq4_txen DDRPHY's gpio_b_dq4_txen source
15	RW	0x0	gpio_b_dq3_txen DDRPHY's gpio_b_dq3_txen source
14	RW	0x0	gpio_b_dq2_txen DDRPHY's gpio_b_dq2_txen source
13	RW	0x0	gpio_b_dq1_txen DDRPHY's gpio_b_dq1_txen source
12	RW	0x0	gpio_b_dq0_txen DDRPHY's gpio_b_dq0_txen source
11	RW	0x0	gpio_a_dq15_txen DDRPHY's gpio_a_dq15_txen source
10	RW	0x0	gpio_a_dq14_txen DDRPHY's gpio_a_dq14_txen source
9	RW	0x0	gpio_a_dq13_txen DDRPHY's gpio_a_dq13_txen source
8	RW	0x0	gpio_a_dq12_txen DDRPHY's gpio_a_dq12_txen source
7	RW	0x0	gpio_a_dq11_txen DDRPHY's gpio_a_dq11_txen source
6	RW	0x0	gpio_a_dq10_txen DDRPHY's gpio_a_dq10_txen source
5	RW	0x0	gpio_a_dq9_txen DDRPHY's gpio_a_dq9_txen source
4	RW	0x0	gpio_a_dq8_txen DDRPHY's gpio_a_dq8_txen source
3	RW	0x0	gpio_a_dq7_txen DDRPHY's gpio_a_dq7_txen source
2	RW	0x0	gpio_a_dq6_txen DDRPHY's gpio_a_dq6_txen source
1	RW	0x0	gpio_a_dq5_txen DDRPHY's gpio_a_dq5_txen source
0	RW	0x0	gpio_a_dq4_txen DDRPHY's gpio_a_dq4_txen source

DDRGRF_DDRPHY_CON2

Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31	RW	0x0	gpio_a_dq3_txdata DDRPHY's gpio_a_dq3_txdata source
30	RW	0x0	gpio_a_dq2_txdata DDRPHY's gpio_a_dq2_txdata source
29	RW	0x0	gpio_a_dq1_txdata DDRPHY's gpio_a_dq1_txdata source
28	RW	0x0	gpio_a_dq0_txdata DDRPHY's gpio_a_dq0_txdata source
27	RW	0x0	gpio_actn_txdata DDRPHY's gpio_actn_txdata source
26	RW	0x0	gpio_odt1_txdata DDRPHY's gpio_odt1_txdata source
25	RW	0x0	gpio_odt0_txdata DDRPHY's gpio_odt0_txdata source

Bit	Attr	Reset Value	Description
24	RW	0x0	gpio_csb1_txdata DDRPHY's gpio_csb1_txdata source
23	RW	0x0	gpio_csb0_txdata DDRPHY's gpio_csb0_txdata source
22	RW	0x0	gpio_ckb_txdata DDRPHY's gpio_ckb_txdata source
21	RW	0x0	gpio_ck_txdata DDRPHY's gpio_ck_txdata source
20	RW	0x0	gpio_bg1_txdata DDRPHY's gpio_bg1_txdata source
19	RW	0x0	gpio_bg0_txdata DDRPHY's gpio_bg0_txdata source
18	RW	0x0	gpio_ba1_txdata DDRPHY's gpio_ba1_txdata source
17	RW	0x0	gpio_a17_txdata DDRPHY's gpio_a17_txdata source
16	RW	0x0	gpio_a16_txdata DDRPHY's gpio_a16_txdata source
15	RW	0x0	gpio_a15_txdata DDRPHY's gpio_a15_txdata source
14	RW	0x0	gpio_a14_txdata DDRPHY's gpio_a14_txdata source
13	RW	0x0	gpio_a13_txdata DDRPHY's gpio_a13_txdata source
12	RW	0x0	gpio_a12_txdata DDRPHY's gpio_a12_txdata source
11	RW	0x0	gpio_a11_txdata DDRPHY's gpio_a11_txdata source
10	RW	0x0	gpio_a10_txdata DDRPHY's gpio_a10_txdata source
9	RW	0x0	gpio_a9_txdata DDRPHY's gpio_a9_txdata source
8	RW	0x0	gpio_a8_txdata DDRPHY's gpio_a8_txdata source
7	RW	0x0	gpio_a7_txdata DDRPHY's gpio_a7_txdata source
6	RW	0x0	gpio_a6_txdata DDRPHY's gpio_a6_txdata source
5	RW	0x0	gpio_a5_txdata DDRPHY's gpio_a5_txdata source
4	RW	0x0	gpio_a4_txdata DDRPHY's gpio_a4_txdata source
3	RW	0x0	gpio_a3_txdata DDRPHY's gpio_a3_txdata source
2	RW	0x0	gpio_a2_txdata DDRPHY's gpio_a2_txdata source
1	RW	0x0	gpio_a1_txdata DDRPHY's gpio_a1_txdata source
0	RW	0x0	gpio_a0_txdata DDRPHY's gpio_a0_txdata source

DDRGRF_DDRPHY_CON3

Address: Operational Base + offset (0x004C)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29	RW	0x0	gpio_resetn_txdata DDRPHY's gpio_resetn_txdata source
28	RW	0x0	gpio_cke0_txdata DDRPHY's gpio_cke0_txdata source
27	RW	0x0	gpio_b_dq15_txdata DDRPHY's gpio_b_dq15_txdata source
26	RW	0x0	gpio_b_dq14_txdata DDRPHY's gpio_b_dq14_txdata source
25	RW	0x0	gpio_b_dq13_txdata DDRPHY's gpio_b_dq13_txdata source
24	RW	0x0	gpio_b_dq12_txdata DDRPHY's gpio_b_dq12_txdata source
23	RW	0x0	gpio_b_dq11_txdata DDRPHY's gpio_b_dq11_txdata source
22	RW	0x0	gpio_b_dq10_txdata DDRPHY's gpio_b_dq10_txdata source
21	RW	0x0	gpio_b_dq9_txdata DDRPHY's gpio_b_dq9_txdata source
20	RW	0x0	gpio_b_dq8_txdata DDRPHY's gpio_b_dq8_txdata source
19	RW	0x0	gpio_b_dq7_txdata DDRPHY's gpio_b_dq7_txdata source
18	RW	0x0	gpio_b_dq6_txdata DDRPHY's gpio_b_dq6_txdata source
17	RW	0x0	gpio_b_dq5_txdata DDRPHY's gpio_b_dq5_txdata source
16	RW	0x0	gpio_b_dq4_txdata DDRPHY's gpio_b_dq4_txdata source
15	RW	0x0	gpio_b_dq3_txdata DDRPHY's gpio_b_dq3_txdata source
14	RW	0x0	gpio_b_dq2_txdata DDRPHY's gpio_b_dq2_txdata source
13	RW	0x0	gpio_b_dq1_txdata DDRPHY's gpio_b_dq1_txdata source
12	RW	0x0	gpio_b_dq0_txdata DDRPHY's gpio_b_dq0_txdata source
11	RW	0x0	gpio_a_dq15_txdata DDRPHY's gpio_a_dq15_txdata source
10	RW	0x0	gpio_a_dq14_txdata DDRPHY's gpio_a_dq14_txdata source
9	RW	0x0	gpio_a_dq13_txdata DDRPHY's gpio_a_dq13_txdata source
8	RW	0x0	gpio_a_dq12_txdata DDRPHY's gpio_a_dq12_txdata source
7	RW	0x0	gpio_a_dq11_txdata DDRPHY's gpio_a_dq11_txdata source
6	RW	0x0	gpio_a_dq10_txdata DDRPHY's gpio_a_dq10_txdata source
5	RW	0x0	gpio_a_dq9_txdata DDRPHY's gpio_a_dq9_txdata source
4	RW	0x0	gpio_a_dq8_txdata DDRPHY's gpio_a_dq8_txdata source

Bit	Attr	Reset Value	Description
3	RW	0x0	gpio_a_dq7_txdata DDRPHY's gpio_a_dq7_txdata source
2	RW	0x0	gpio_a_dq6_txdata DDRPHY's gpio_a_dq6_txdata source
1	RW	0x0	gpio_a_dq5_txdata DDRPHY's gpio_a_dq5_txdata source
0	RW	0x0	gpio_a_dq4_txdata DDRPHY's gpio_a_dq4_txdata source

DDRGRF_DDRPHY_CON4

Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31	RW	0x0	gpio_a_dq3_rxen DDRPHY's gpio_a_dq3_rxen source
30	RW	0x0	gpio_a_dq2_rxen DDRPHY's gpio_a_dq2_rxen source
29	RW	0x0	gpio_a_dq1_rxen DDRPHY's gpio_a_dq1_rxen source
28	RW	0x0	gpio_a_dq0_rxen DDRPHY's gpio_a_dq0_rxen source
27	RW	0x0	gpio_actn_rxen DDRPHY's gpio_actn_rxen source
26	RW	0x0	gpio_odt1_rxen DDRPHY's gpio_odt1_rxen source
25	RW	0x0	gpio_odt0_rxen DDRPHY's gpio_odt0_rxen source
24	RW	0x0	gpio_csb1_rxen DDRPHY's gpio_csb1_rxen source
23	RW	0x0	gpio_csb0_rxen DDRPHY's gpio_csb0_rxen source
22	RW	0x0	gpio_ckb_rxen DDRPHY's gpio_ckb_rxen source
21	RW	0x0	gpio_ck_rxen DDRPHY's gpio_ck_rxen source
20	RW	0x0	gpio_bg1_rxen DDRPHY's gpio_bg1_rxen source
19	RW	0x0	gpio_bg0_rxen DDRPHY's gpio_bg0_rxen source
18	RW	0x0	gpio_ba1_rxen DDRPHY's gpio_ba1_rxen source
17	RW	0x0	gpio_a17_rxen DDRPHY's gpio_a17_rxen source
16	RW	0x0	gpio_a16_rxen DDRPHY's gpio_a16_rxen source
15	RW	0x0	gpio_a15_rxen DDRPHY's gpio_a15_rxen source
14	RW	0x0	gpio_a14_rxen DDRPHY's gpio_a14_rxen source
13	RW	0x0	gpio_a13_rxen DDRPHY's gpio_a13_rxen source
12	RW	0x0	gpio_a12_rxen DDRPHY's gpio_a12_rxen source
11	RW	0x0	gpio_a11_rxen DDRPHY's gpio_a11_rxen source

Bit	Attr	Reset Value	Description
10	RW	0x0	gpio_a10_rxen DDRPHY's gpio_a10_rxen source
9	RW	0x0	gpio_a9_rxen DDRPHY's gpio_a9_rxen source
8	RW	0x0	gpio_a8_rxen DDRPHY's gpio_a8_rxen source
7	RW	0x0	gpio_a7_rxen DDRPHY's gpio_a7_rxen source
6	RW	0x0	gpio_a6_rxen DDRPHY's gpio_a6_rxen source
5	RW	0x0	gpio_a5_rxen DDRPHY's gpio_a5_rxen source
4	RW	0x0	gpio_a4_rxen DDRPHY's gpio_a4_rxen source
3	RW	0x0	gpio_a3_rxen DDRPHY's gpio_a3_rxen source
2	RW	0x0	gpio_a2_rxen DDRPHY's gpio_a2_rxen source
1	RW	0x0	gpio_a1_rxen DDRPHY's gpio_a1_rxen source
0	RW	0x0	gpio_a0_rxen DDRPHY's gpio_a0_rxen source

DDRGRF_DDRPHY_CON5

Address: Operational Base + offset (0x0054)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29	RW	0x0	gpio_resetn_rxen DDRPHY's gpio_resetn_rxen source
28	RW	0x0	gpio_cke0_rxen DDRPHY's gpio_cke0_rxen source
27	RW	0x0	gpio_b_dq15_rxen DDRPHY's gpio_b_dq15_rxen source
26	RW	0x0	gpio_b_dq14_rxen DDRPHY's gpio_b_dq14_rxen source
25	RW	0x0	gpio_b_dq13_rxen DDRPHY's gpio_b_dq13_rxen source
24	RW	0x0	gpio_b_dq12_rxen DDRPHY's gpio_b_dq12_rxen source
23	RW	0x0	gpio_b_dq11_rxen DDRPHY's gpio_b_dq11_rxen source
22	RW	0x0	gpio_b_dq10_rxen DDRPHY's gpio_b_dq10_rxen source
21	RW	0x0	gpio_b_dq9_rxen DDRPHY's gpio_b_dq9_rxen source
20	RW	0x0	gpio_b_dq8_rxen DDRPHY's gpio_b_dq8_rxen source
19	RW	0x0	gpio_b_dq7_rxen DDRPHY's gpio_b_dq7_rxen source
18	RW	0x0	gpio_b_dq6_rxen DDRPHY's gpio_b_dq6_rxen source
17	RW	0x0	gpio_b_dq5_rxen DDRPHY's gpio_b_dq5_rxen source

Bit	Attr	Reset Value	Description
16	RW	0x0	gpio_b_dq4_rxen DDRPHY's gpio_b_dq4_rxen source
15	RW	0x0	gpio_b_dq3_rxen DDRPHY's gpio_b_dq3_rxen source
14	RW	0x0	gpio_b_dq2_rxen DDRPHY's gpio_b_dq2_rxen source
13	RW	0x0	gpio_b_dq1_rxen DDRPHY's gpio_b_dq1_rxen source
12	RW	0x0	gpio_b_dq0_rxen DDRPHY's gpio_b_dq0_rxen source
11	RW	0x0	gpio_a_dq15_rxen DDRPHY's gpio_a_dq15_rxen source
10	RW	0x0	gpio_a_dq14_rxen DDRPHY's gpio_a_dq14_rxen source
9	RW	0x0	gpio_a_dq13_rxen DDRPHY's gpio_a_dq13_rxen source
8	RW	0x0	gpio_a_dq12_rxen DDRPHY's gpio_a_dq12_rxen source
7	RW	0x0	gpio_a_dq11_rxen DDRPHY's gpio_a_dq11_rxen source
6	RW	0x0	gpio_a_dq10_rxen DDRPHY's gpio_a_dq10_rxen source
5	RW	0x0	gpio_a_dq9_rxen DDRPHY's gpio_a_dq9_rxen source
4	RW	0x0	gpio_a_dq8_rxen DDRPHY's gpio_a_dq8_rxen source
3	RW	0x0	gpio_a_dq7_rxen DDRPHY's gpio_a_dq7_rxen source
2	RW	0x0	gpio_a_dq6_rxen DDRPHY's gpio_a_dq6_rxen source
1	RW	0x0	gpio_a_dq5_rxen DDRPHY's gpio_a_dq5_rxen source
0	RW	0x0	gpio_a_dq4_rxen DDRPHY's gpio_a_dq4_rxen source

DDRGRF_DDR_STATUS0

Address: Operational Base + offset (0x0100)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	mrr_data0_0 Mode register read data 0 bit[31:0] status

DDRGRF_DDR_STATUS1

Address: Operational Base + offset (0x0104)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	mrr_data0_1 Mode register read data 0 bit[63:32] status

DDRGRF_DDR_STATUS2

Address: Operational Base + offset (0x0108)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	mrr_data0_2 Mode register read data 0 bit[95:64] status

DDRGRF_DDR_STATUS3

Address: Operational Base + offset (0x010C)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	mrr_data0_3 Mode register read data 0 bit[127:96] status

DDRGRF_DDR_STATUS4

Address: Operational Base + offset (0x0110)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	mrr_data1_0 Mode register read data 1 bit[31:0] status

DDRGRF_DDR_STATUS5

Address: Operational Base + offset (0x0114)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	mrr_data1_1 Mode register read data 1 bit[63:32] status

DDRGRF_DDR_STATUS6

Address: Operational Base + offset (0x0118)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	mrr_data1_2 Mode register read data 1 bit[95:64] status

DDRGRF_DDR_STATUS7

Address: Operational Base + offset (0x011C)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	mrr_data1_3 Mode register read data 1 bit[127:96] status

DDRGRF_DDR_STATUS8

Address: Operational Base + offset (0x0120)

Bit	Attr	Reset Value	Description
31:21	RO	0x000	reserved
20	RO	0x0	dfi_freq_change_type DFI frequency change type status
19	RO	0x0	dfi_freq_change_req DFI frequency change request state. 1'b0: Inactive 1'b1: Active
18	RO	0x0	cpu_port_probe_mainStatAlarm The status of cpu_port_probe_mainStatAlarm. 1'b0: Inactive 1'b1: Active
17	RO	0x0	cpu_port_probe_mainTraceAlarm The status of cpu_port_probe_mainTraceAlarm. 1'b0: Inactive 1'b1: Active
16	RO	0x0	gpu_port_probe_mainStatAlarm The status of gpu_port_probe_mainStatAlarm. 1'b0: Inactive 1'b1: Active
15	RO	0x0	gpu_port_probe_mainTraceAlarm The status of gpu_port_probe_mainTraceAlarm. 1'b0: Inactive 1'b1: Active

Bit	Attr	Reset Value	Description
14	RO	0x0	mmip_port_probe_mainStatAlarm The status of mmip_port_probe_mainStatAlarm. 1'b0: Inactive 1'b1: Active
13	RO	0x0	mmip_port_probe_mainTraceAlarm The status of mmip_port_probe_mainTraceAlarm. 1'b0: Inactive 1'b1: Active
12	RO	0x0	peri_port_probe_mainStatAlarm The status of peri_port_probe_mainStatAlarm. 1'b0: Inactive 1'b1: Active
11	RO	0x0	peri_port_probe_mainTraceAlarm The status of peri_port_probe_mainTraceAlarm. 1'b0: Inactive 1'b1: Active
10	RO	0x0	dfi_scramble_read_of The status of dfi_scramble_read_of. 1'b0: Inactive 1'b1: Active
9	RO	0x0	dfi_scramble_write_of The status of dfi_scramble_write_of. 1'b0: Inactive 1'b1: Active
8	RO	0x0	dfi_scramble_key_ready The status of dfi_scramble_key_ready. 1'b0: Not ready 1'b1: Ready
7:6	RO	0x0	ddrc_reg_selfref_type The status of ddrc_reg_selfref_type.
5	RO	0x0	cactive_aclk The status of cactive_aclk. 1'b0: Inactive 1'b1: Active
4	RO	0x1	csysaclk_aclk The status of csysaclk_aclk. 1'b0: Inactive 1'b1: Active
3	RO	0x0	csysreq_aclk The status of csysreq_aclk. 1'b0: Inactive 1'b1: Active
2	RO	0x1	cactive_ddrc The status of external cactive_ddrc. 1'b0: Inactive 1'b1: Active
1	RO	0x1	csysack_ddrc The status of external csysack_ddrc. 1'b0: Inactive 1'b1: Active
0	RO	0x1	csysreq_ddrc The status of external csysreq_ddrc. 1'b0: Inactive 1'b1: Active

DDRGRF_DDR_STATUS9

Address: Operational Base + offset (0x0124)

Bit	Attr	Reset Value	Description
31	RO	0x1	dfi_lp_ck_disable The status of low power of DDR PHY
30	RO	0x0	reserved
29:24	RO	0x00	hif_refresh_req_bank The status of hif_refresh_req_bank
23	RO	0x0	reserved
22:16	RO	0x00	wr_credit_cnt The status of wr_credit_cnt
15	RO	0x0	reserved
14:8	RO	0x00	hpr_credit_cnt The status of hpr_credit_cnt
7	RO	0x0	reserved
6:0	RO	0x00	lpr_credit_cnt The status of lpr_credit_cnt

DDRGRF_DDR_STATUS10

Address: Operational Base + offset (0x0128)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	mrr_data2_0 Mode register read data 2 bit[31:0] status

DDRGRF_DDR_STATUS11

Address: Operational Base + offset (0x012C)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	mrr_data2_1 Mode register read data 2 bit[63:32] status

DDRGRF_DDR_STATUS12

Address: Operational Base + offset (0x0130)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	mrr_data2_2 Mode register read data 2 bit[95:64] status

DDRGRF_DDR_STATUS13

Address: Operational Base + offset (0x0134)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	mrr_data2_3 Mode register read data 2 bit[127:96] status

DDRGRF_DDR_STATUS14

Address: Operational Base + offset (0x0138)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	mrr_data3_0 Mode register read data 3 bit[31:0] status

DDRGRF_DDR_STATUS15

Address: Operational Base + offset (0x013C)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	mrr_data3_1 Mode register read data 3 bit[63:32] status

DDRGRF_DDR_STATUS16

Address: Operational Base + offset (0x0140)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	mrr_data3_2 Mode register read data 3 bit[95:64] status

DDRGRF_DDR_STATUS17

Address: Operational Base + offset (0x0144)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	mrr_data3_3 Mode register read data 3 bit[127:96] status

DDRGRF_DDRHOLD_STATUS

Address: Operational Base + offset (0x0150)

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28	RO	0x0	arvalid_from_msch arvalid from msch status. 1'b0: Inactive 1'b1: Active
27	RO	0x0	arvalid_to_next arvalid to next status. 1'b0: Inactive 1'b1: Active
26	RO	0x0	arready_from_next arready from next status. 1'b0: Inactive 1'b1: Active
25	RO	0x0	arready_to_msch arready to msch status. 1'b0: Inactive 1'b1: Active
24	RO	0x1	arready_low arready low status. 1'b0: Inactive 1'b1: Active
23:21	RO	0x0	reserved
20	RO	0x0	wvalid_from_msch wvalid from msch status. 1'b0: Inactive 1'b1: Active
19	RO	0x0	wvalid_to_next wvalid to next status. 1'b0: Inactive 1'b1: Active
18	RO	0x0	wready_from_next wready from next status. 1'b0: Inactive 1'b1: Active
17	RO	0x0	wready_to_msch wready to msch status. 1'b0: Inactive 1'b1: Active
16	RO	0x1	wready_low wready low status. 1'b0: Inactive 1'b1: Active

Bit	Attr	Reset Value	Description
15:13	RO	0x0	reserved
12	RO	0x0	awvalid_from_msch awvalid from msch status. 1'b0: Inactive 1'b1: Active
11	RO	0x0	awvalid_to_next awvalid to next status. 1'b0: Inactive 1'b1: Active
10	RO	0x0	awready_from_next awready from next status. 1'b0: Inactive 1'b1: Active
9	RO	0x0	awready_to_msch awready to msch status. 1'b0: Inactive 1'b1: Active
8	RO	0x1	awready_low awready low status. 1'b0: Inactive 1'b1: Active
7:0	RO	0x40	write_burst_cnt Write burst count status

DDRGRF_DDRPHY_STATUS0

Address: Operational Base + offset (0x0160)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29	RO	0x1	gpio_resetn_rxdata DDRPHY's gpio_resetn_rxdata output
28	RO	0x1	gpio_cke0_rxdata DDRPHY's gpio_cke0_rxdata output
27	RO	0x1	gpio_actn_rxdata DDRPHY's gpio_actn_rxdata output
26	RO	0x1	gpio_odt1_rxdata DDRPHY's gpio_odt1_rxdata output
25	RO	0x1	gpio_odt0_rxdata DDRPHY's gpio_odt0_rxdata output
24	RO	0x1	gpio_csb1_rxdata DDRPHY's gpio_csb1_rxdata output
23	RO	0x1	gpio_csb0_rxdata DDRPHY's gpio_csb0_rxdata output
22	RO	0x1	gpio_ckb_rxdata DDRPHY's gpio_ckb_rxdata output
21	RO	0x1	gpio_ck_rxdata DDRPHY's gpio_ck_rxdata output
20	RO	0x1	gpio_bg1_rxdata DDRPHY's gpio_bg1_rxdata output
19	RO	0x1	gpio_bg0_rxdata DDRPHY's gpio_bg0_rxdata output
18	RO	0x1	gpio_ba1_rxdata DDRPHY's gpio_ba1_rxdata output
17	RO	0x1	gpio_a17_rxdata DDRPHY's gpio_a17_rxdata output

Bit	Attr	Reset Value	Description
16	RO	0x1	gpio_a16_rxdata DDRPHY's gpio_a16_rxdata output
15	RO	0x1	gpio_a15_rxdata DDRPHY's gpio_a15_rxdata output
14	RO	0x1	gpio_a14_rxdata DDRPHY's gpio_a14_rxdata output
13	RO	0x1	gpio_a13_rxdata DDRPHY's gpio_a13_rxdata output
12	RO	0x1	gpio_a12_rxdata DDRPHY's gpio_a12_rxdata output
11	RO	0x1	gpio_a11_rxdata DDRPHY's gpio_a11_rxdata output
10	RO	0x1	gpio_a10_rxdata DDRPHY's gpio_a10_rxdata output
9	RO	0x1	gpio_a9_rxdata DDRPHY's gpio_a9_rxdata output
8	RO	0x1	gpio_a8_rxdata DDRPHY's gpio_a8_rxdata output
7	RO	0x1	gpio_a7_rxdata DDRPHY's gpio_a7_rxdata output
6	RO	0x1	gpio_a6_rxdata DDRPHY's gpio_a6_rxdata output
5	RO	0x1	gpio_a5_rxdata DDRPHY's gpio_a5_rxdata output
4	RO	0x1	gpio_a4_rxdata DDRPHY's gpio_a4_rxdata output
3	RO	0x1	gpio_a3_rxdata DDRPHY's gpio_a3_rxdata output
2	RO	0x1	gpio_a2_rxdata DDRPHY's gpio_a2_rxdata output
1	RO	0x1	gpio_a1_rxdata DDRPHY's gpio_a1_rxdata output
0	RO	0x1	gpio_a0_rxdata DDRPHY's gpio_a0_rxdata output

DDRGRF_DDRPHY_STATUS1

Address: Operational Base + offset (0x0164)

Bit	Attr	Reset Value	Description
31	RO	0x1	gpio_b_dq15_rxdata DDRPHY's gpio_b_dq15_rxdata output
30	RO	0x1	gpio_b_dq14_rxdata DDRPHY's gpio_b_dq14_rxdata output
29	RO	0x1	gpio_b_dq13_rxdata DDRPHY's gpio_b_dq13_rxdata output
28	RO	0x1	gpio_b_dq12_rxdata DDRPHY's gpio_b_dq12_rxdata output
27	RO	0x1	gpio_b_dq11_rxdata DDRPHY's gpio_b_dq11_rxdata output
26	RO	0x1	gpio_b_dq10_rxdata DDRPHY's gpio_b_dq10_rxdata output
25	RO	0x1	gpio_b_dq9_rxdata DDRPHY's gpio_b_dq9_rxdata output
24	RO	0x1	gpio_b_dq8_rxdata DDRPHY's gpio_b_dq8_rxdata output

Bit	Attr	Reset Value	Description
23	RO	0x1	gpio_b_dq7_rxdata DDRPHY's gpio_b_dq7_rxdata output
22	RO	0x1	gpio_b_dq6_rxdata DDRPHY's gpio_b_dq6_rxdata output
21	RO	0x1	gpio_b_dq5_rxdata DDRPHY's gpio_b_dq5_rxdata output
20	RO	0x1	gpio_b_dq4_rxdata DDRPHY's gpio_b_dq4_rxdata output
19	RO	0x1	gpio_b_dq3_rxdata DDRPHY's gpio_b_dq3_rxdata output
18	RO	0x1	gpio_b_dq2_rxdata DDRPHY's gpio_b_dq2_rxdata output
17	RO	0x1	gpio_b_dq1_rxdata DDRPHY's gpio_b_dq1_rxdata output
16	RO	0x1	gpio_b_dq0_rxdata DDRPHY's gpio_b_dq0_rxdata output
15	RO	0x1	gpio_a_dq15_rxdata DDRPHY's gpio_a_dq15_rxdata output
14	RO	0x1	gpio_a_dq14_rxdata DDRPHY's gpio_a_dq14_rxdata output
13	RO	0x1	gpio_a_dq13_rxdata DDRPHY's gpio_a_dq13_rxdata output
12	RO	0x1	gpio_a_dq12_rxdata DDRPHY's gpio_a_dq12_rxdata output
11	RO	0x1	gpio_a_dq11_rxdata DDRPHY's gpio_a_dq11_rxdata output
10	RO	0x1	gpio_a_dq10_rxdata DDRPHY's gpio_a_dq10_rxdata output
9	RO	0x1	gpio_a_dq9_rxdata DDRPHY's gpio_a_dq9_rxdata output
8	RO	0x1	gpio_a_dq8_rxdata DDRPHY's gpio_a_dq8_rxdata output
7	RO	0x1	gpio_a_dq7_rxdata DDRPHY's gpio_a_dq7_rxdata output
6	RO	0x1	gpio_a_dq6_rxdata DDRPHY's gpio_a_dq6_rxdata output
5	RO	0x1	gpio_a_dq5_rxdata DDRPHY's gpio_a_dq5_rxdata output
4	RO	0x1	gpio_a_dq4_rxdata DDRPHY's gpio_a_dq4_rxdata output
3	RO	0x1	gpio_a_dq3_rxdata DDRPHY's gpio_a_dq3_rxdata output
2	RO	0x1	gpio_a_dq2_rxdata DDRPHY's gpio_a_dq2_rxdata output
1	RO	0x1	gpio_a_dq1_rxdata DDRPHY's gpio_a_dq1_rxdata output
0	RO	0x1	gpio_a_dq0_rxdata DDRPHY's gpio_a_dq0_rxdata output

6.5 PMUGRF Register Description

6.5.1 Registers Summary

Name	Offset	Size	Reset Value	Description
PMUGRF_GPIO0A_IOMUX_L	0x0000	W	0x00001010	GPIO0A low bits IOMUX control register
PMUGRF_GPIO0A_IOMUX_H	0x0004	W	0x00000000	GPIO0A high bits IOMUX control register
PMUGRF_GPIO0B_IOMUX_L	0x0008	W	0x00000000	GPIO0B low bits IOMUX control register
PMUGRF_GPIO0B_IOMUX_H	0x000C	W	0x00000000	GPIO0B high bits IOMUX control register
PMUGRF_GPIO0C_IOMUX_L	0x0010	W	0x00000000	GPIO0C low bits IOMUX control register
PMUGRF_GPIO0A_DS_L	0x0020	W	0x00001111	GPIO0A low bits driver strength control register
PMUGRF_GPIO0A_DS_H	0x0024	W	0x00001111	GPIO0A high bits driver strength control register
PMUGRF_GPIO0B_DS_L	0x0028	W	0x00001111	GPIO0B low bits driver strength control register
PMUGRF_GPIO0B_DS_H	0x002C	W	0x00001111	GPIO0B high bits driver strength control register
PMUGRF_GPIO0C_DS_L	0x0030	W	0x00001111	GPIO0C low bits driver strength control register
PMUGRF_OSC_DS	0x0034	W	0x00000001	OSC driver strength control register
PMUGRF_GPIO0A_P	0x0040	W	0x0000A542	GPIO0A PU/PD control register
PMUGRF_GPIO0B_P	0x0044	W	0x0000A5AA	GPIO0B PU/PD control register
PMUGRF_GPIO0C_P_L	0x0048	W	0x000000AA	GPIO0C low bits PU/PD control register
PMUGRF_GPIO0A_IE	0x0050	W	0x000000FF	GPIO0A IE control register
PMUGRF_GPIO0B_IE	0x0054	W	0x000000FF	GPIO0B IE control register
PMUGRF_GPIO0C_IE_L	0x0058	W	0x0000000F	GPIO0C low bits IE control register
PMUGRF_GPIO0A_SMT	0x0060	W	0x00000000	GPIO0A SMT control register
PMUGRF_GPIO0B_SMT	0x0064	W	0x00000000	GPIO0B SMT control register
PMUGRF_GPIO0C_SMT_L	0x0068	W	0x00000000	GPIO0C low bits SMT control register
PMUGRF_PMU_SOC_CON0	0x0100	W	0x00000000	PMU SOC control register 0
PMUGRF_PMU_SOC_CON1	0x0104	W	0x00000000	PMU SOC control register 1
PMUGRF_PMU_SOC_CON2	0x0108	W	0x00000000	PMU SOC control register 2
PMUGRF_PMU_SOC_CON3	0x010C	W	0x0000001C	PMU SOC control register 3
PMUGRF_PMU_SOC_CON4	0x0110	W	0x00002805	PMU SOC control register 4
PMUGRF_IOFUNC_CON4	0x0114	W	0x00000000	IO function control register 4
PMUGRF_IOFUNC_CON5	0x0118	W	0x00000000	IO function control register 5
PMUGRF_IO_VSEL	0x0140	W	0x00000000	VCCIO voltage control register
PMUGRF_IO_VRET	0x0144	W	0x000003FE	VCCIO retention control register
PMUGRF_PMUPVTM_CLKDIV	0x0180	W	0x00000000	PMU PVTM 32KHz clock divider register
PMUGRF_OS_REG0	0x0200	W	0x00000000	OS register 0
PMUGRF_OS_REG1	0x0204	W	0x00000000	OS register 1
PMUGRF_OS_REG2	0x0208	W	0x00000000	OS register 2
PMUGRF_OS_REG3	0x020C	W	0x00000000	OS register 3

Name	Offset	Size	Reset Value	Description
PMUGRF_OS_REG4	0x0210	W	0x00000000	OS register 4
PMUGRF_OS_REG5	0x0214	W	0x00000000	OS register 5
PMUGRF_OS_REG6	0x0218	W	0x00000000	OS register 6
PMUGRF_OS_REG7	0x021C	W	0x00000000	OS register 7
PMUGRF_OS_REG8	0x0220	W	0x00000000	OS register 9
PMUGRF_OS_REG9	0x0224	W	0x00000000	OS register 9
PMUGRF_OS_REG10	0x0228	W	0x00000000	OS register 10
PMUGRF_OS_REG11	0x022C	W	0x00000000	OS register 11
PMUGRF_RSTFUNC_STATUS	0x0230	W	0x00000000	System reset status register
PMUGRF_RSTFUNC_CLR	0x0234	W	0x00000000	System reset status clear register
PMUGRF_SD_DETECT_CON	0x0380	W	0x00000000	SDMMC detect irq enable register
PMUGRF_SD_DETECT_STATUS	0x0390	W	0x00000000	SDMMC detect irq status register
PMUGRF_SD_DETECT_CLR	0x03A0	W	0x00000000	SDMMC detect irq clear register
PMUGRF_SD_DETECT_COUNT	0x03B0	W	0x000003E8	SDMMC detect counter register

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

6.5.2 Detail Register Description

PMUGRF_GPIO0A_IOMUX_L

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14:12	RW	0x1	gpio0a3_sel 3'h0: GPIO0_A3_u 3'h1: SDMMC0_DET Others: Reserved
11	RO	0x0	reserved
10:8	RW	0x0	gpio0a2_sel 3'h0: GPIO0_A2_z 3'h1: CLK_32K Others: Reserved
7	RO	0x0	reserved
6:4	RW	0x1	gpio0a1_sel 3'h0: GPIO0_A1_z 3'h1: TSADC_SHUT_M0 3'h2: TSADC_SHUTORG Others: Reserved
3	RO	0x0	reserved
2:0	RW	0x0	gpio0a0_sel 3'h0: GPIO0_A0_d 3'h1: CLK_REF Others: Reserved

PMUGRF_GPIO0A_IOMUX_H

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14:12	RW	0x0	gpio0a7_sel 3'h0: GPIO0_A7_d 3'h1: SPI0_MISO_M0 Others: Reserved
11	RO	0x0	reserved
10:8	RW	0x0	gpio0a6_sel 3'h0: GPIO0_A6_d 3'h1: SPI0_MOSI_M0 Others: Reserved
7	RO	0x0	reserved
6:4	RW	0x0	gpio0a5_sel 3'h0: GPIO0_A5_u 3'h1: SPI0_CS0N_M0 Others: Reserved
3	RO	0x0	reserved
2:0	RW	0x0	gpio0a4_sel 3'h0: GPIO0_A4_u 3'h1: SPI0_CS1N_M0 Others: Reserved

PMUGRF_GPIO0B_IOMUX_L

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14:12	RW	0x0	gpio0b3_sel 3'h0: GPIO0_B3_d 3'h1: FLASH_VOL_SEL Others: Reserved
11	RO	0x0	reserved
10:8	RW	0x0	gpio0b2_sel 3'h0: GPIO0_B2_d 3'h1: PMIC_SLEEP 3'h2: TSADC_SHUT_M1 3'h3: PWM6_PIN_M0 Others: Reserved
7	RO	0x0	reserved
6:4	RW	0x0	gpio0b1_sel 3'h0: GPIO0_B1_d 3'h1: PMIC_INT 3'h3: PWM7_PIN_M0 Others: Reserved
3	RO	0x0	reserved
2:0	RW	0x0	gpio0b0_sel 3'h0: GPIO0_B0_d 3'h1: SPI0_CLK_M0 Others: Reserved

PMUGRF_GPIO0B_IOMUX_H

Address: Operational Base + offset (0x000C)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14:12	RW	0x0	gpio0b7_sel 3'h0: GPIO0_B7_d 3'h2: UART1_RX_M0 3'h3: PWM1_PIN_M0 Others: Reserved
11	RO	0x0	reserved
10:8	RW	0x0	gpio0b6_sel 3'h0: GPIO0_B6_d 3'h2: UART1_TX_M0 3'h3: PWM0_PIN_M0 Others: Reserved
7	RO	0x0	reserved
6:4	RW	0x0	gpio0b5_sel 3'h0: GPIO0_B5_u 3'h1: I2C0_SDA Others: Reserved
3	RO	0x0	reserved
2:0	RW	0x0	gpio0b4_sel 3'h0: GPIO0_B4_u 3'h1: I2C0_SCL Others: Reserved

PMUGRF_GPIO0C_IOMUX_L

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14:12	RW	0x0	gpio0c3_sel 3'h0: GPIO0_C3_d 3'h1: I2C2_SDA 3'h3: PWM5_PIN_M0 Others: Reserved
11	RO	0x0	reserved
10:8	RW	0x0	gpio0c2_sel 3'h0: GPIO0_C2_d 3'h1: I2C2_SCL 3'h3: PWM4_PIN_M0 Others: Reserved
7	RO	0x0	reserved

Bit	Attr	Reset Value	Description
6:4	RW	0x0	gpio0c1_sel 3'h0: GPIO0_C1_d 3'h1: PMU_DEBUG 3'h2: UART1_CTSN_M0 3'h3: PWM3_PIN_M0 Others: Reserved
3	RO	0x0	reserved
2:0	RW	0x0	gpio0c0_sel 3'h0: GPIO0_C0_d 3'h1: SDMMC0_PWR 3'h2: UART1_RTSN_M0 3'h3: PWM2_PIN_M0 Others: Reserved

PMUGRF_GPIO0A_DS_L

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RW	0x1	gpio0a3_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
11:8	RW	0x1	gpio0a2_ds 4'd0: Level 0 strength 4'd1: Level 1 strength 4'd2: Level 2 strength 4'd3: Level 3 strength 4'd4: Level 4 strength 4'd5: Level 5 strength 4'd6: Level 6 strength 4'd7: Level 7 strength 4'd8: Level 8 strength 4'd9: Level 9 strength 4'd10: Level 10 strength 4'd11: Level 11 strength 4'd12: Level 12 strength 4'd13: Level 13 strength 4'd14: Level 14 strength 4'd15: Level 15 strength
7:4	RW	0x1	gpio0a1_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved

Bit	Attr	Reset Value	Description
3:0	RW	0x1	gpio0a0_ds 4'd0: Level 0 strength 4'd1: Level 1 strength 4'd2: Level 2 strength 4'd3: Level 3 strength 4'd4: Level 4 strength 4'd5: Level 5 strength 4'd6: Level 6 strength 4'd7: Level 7 strength 4'd8: Level 8 strength 4'd9: Level 9 strength 4'd10: Level 10 strength 4'd11: Level 11 strength 4'd12: Level 12 strength 4'd13: Level 13 strength 4'd14: Level 14 strength 4'd15: Level 15 strength

PMUGRF_GPIO0A_DS_H

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RW	0x1	gpio0a7_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
11:8	RW	0x1	gpio0a6_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
7:4	RW	0x1	gpio0a5_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
3:0	RW	0x1	gpio0a4_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved

PMUGRF_GPIO0B_DS_L

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RW	0x1	gpio0b3_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
11:8	RW	0x1	gpio0b2_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
7:4	RW	0x1	gpio0b1_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
3:0	RW	0x1	gpio0b0_ds 4'd0: Level 0 strength 4'd1: Level 1 strength 4'd2: Level 2 strength 4'd3: Level 3 strength 4'd4: Level 4 strength 4'd5: Level 5 strength 4'd6: Level 6 strength 4'd7: Level 7 strength 4'd8: Level 8 strength 4'd9: Level 9 strength 4'd10: Level 10 strength 4'd11: Level 11 strength 4'd12: Level 12 strength 4'd13: Level 13 strength 4'd14: Level 14 strength 4'd15: Level 15 strength

PMUGRF_GPIO0B_DS_H

Address: Operational Base + offset (0x002C)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RW	0x1	gpio0b7_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved

Bit	Attr	Reset Value	Description
11:8	RW	0x1	gpio0b6_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
7:4	RW	0x1	gpio0b5_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
3:0	RW	0x1	gpio0b4_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved

PMUGRF_GPIO0C_DS_L

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RW	0x1	gpio0c3_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
11:8	RW	0x1	gpio0c2_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
7:4	RW	0x1	gpio0c1_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved
3:0	RW	0x1	gpio0c0_ds 4'd0: Level 0' strength 4'd1: Level 1' strength 4'd2: Level 2' strength 4'd3: Level 3' strength Others: Reserved

PMUGRF_OSC_DS

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:2	RO	0x0000	reserved
1:0	RW	0x1	osc_ds 2'd0: Level 0'' strength 2'd1: Level 1'' strength 2'd2: Level 2'' strength 2'd3: Level 3'' strength

PMUGRF_GPIO0A_P

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RW	0x2	gpio0a7_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
13:12	RW	0x2	gpio0a6_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
11:10	RW	0x1	gpio0a5_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
9:8	RW	0x1	gpio0a4_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
7:6	RW	0x1	gpio0a3_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
5:4	RW	0x0	gpio0a2_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
3:2	RW	0x0	gpio0a1_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved

Bit	Attr	Reset Value	Description
1:0	RW	0x2	gpio0a0_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved

PMUGRF_GPIO0B_P

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RW	0x2	gpio0b7_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
13:12	RW	0x2	gpio0b6_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
11:10	RW	0x1	gpio0b5_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
9:8	RW	0x1	gpio0b4_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
7:6	RW	0x2	gpio0b3_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
5:4	RW	0x2	gpio0b2_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
3:2	RW	0x2	gpio0b1_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
1:0	RW	0x2	gpio0b0_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved

PMUGRF_GPIO0C_P_L

Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7:6	RW	0x2	gpio0c3_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
5:4	RW	0x2	gpio0c2_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
3:2	RW	0x2	gpio0c1_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved
1:0	RW	0x2	gpio0c0_p 2'b00: Z (Normal operation) 2'b01: Weak 1 (pull-up) 2'b10: Weak 0 (pull-down) 2'b11: Reserved

PMUGRF_GPIO0A_IE

Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x1	gpio0a7_ie 1'b0: Input disable 1'b1: Input enable
6	RW	0x1	gpio0a6_ie 1'b0: Input disable 1'b1: Input enable
5	RW	0x1	gpio0a5_ie 1'b0: Input disable 1'b1: Input enable
4	RW	0x1	gpio0a4_ie 1'b0: Input disable 1'b1: Input enable
3	RW	0x1	gpio0a3_ie 1'b0: Input disable 1'b1: Input enable
2	RW	0x1	gpio0a2_ie 1'b0: Input disable 1'b1: Input enable

Bit	Attr	Reset Value	Description
1	RW	0x1	gpio0a1_ie 1'b0: Input disable 1'b1: Input enable
0	RW	0x1	gpio0a0_ie 1'b0: Input disable 1'b1: Input enable

PMUGRF_GPIO0B_IE

Address: Operational Base + offset (0x0054)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x1	gpio0b7_ie 1'b0: Input disable 1'b1: Input enable
6	RW	0x1	gpio0b6_ie 1'b0: Input disable 1'b1: Input enable
5	RW	0x1	gpio0b5_ie 1'b0: Input disable 1'b1: Input enable
4	RW	0x1	gpio0b4_ie 1'b0: Input disable 1'b1: Input enable
3	RW	0x1	gpio0b3_ie 1'b0: Input disable 1'b1: Input enable
2	RW	0x1	gpio0b2_ie 1'b0: Input disable 1'b1: Input enable
1	RW	0x1	gpio0b1_ie 1'b0: Input disable 1'b1: Input enable
0	RW	0x1	gpio0b0_ie 1'b0: Input disable 1'b1: Input enable

PMUGRF_GPIO0C_IE_L

Address: Operational Base + offset (0x0058)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:4	RO	0x000	reserved
3	RW	0x1	gpio0c3_ie 1'b0: Input disable 1'b1: Input enable
2	RW	0x1	gpio0c2_ie 1'b0: Input disable 1'b1: Input enable

Bit	Attr	Reset Value	Description
1	RW	0x1	gpio0c1_ie 1'b0: Input disable 1'b1: Input enable
0	RW	0x1	gpio0c0_ie 1'b0: Input disable 1'b1: Input enable

PMUGRF_GPIO0A_SMT

Address: Operational Base + offset (0x0060)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x0	gpio0a7_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
6	RW	0x0	gpio0a6_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
5	RW	0x0	gpio0a5_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
4	RW	0x0	gpio0a4_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
3	RW	0x0	gpio0a3_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
2	RW	0x0	gpio0a2_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
1	RW	0x0	gpio0a1_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
0	RW	0x0	gpio0a0_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled

PMUGRF_GPIO0B_SMT

Address: Operational Base + offset (0x0064)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x0	gpio0b7_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
6	RW	0x0	gpio0b6_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled

Bit	Attr	Reset Value	Description
5	RW	0x0	gpio0b5_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
4	RW	0x0	gpio0b4_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
3	RW	0x0	gpio0b3_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
2	RW	0x0	gpio0b2_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
1	RW	0x0	gpio0b1_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
0	RW	0x0	gpio0b0_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled

PMUGRF_GPIO0C_SMT_L

Address: Operational Base + offset (0x0068)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:4	RO	0x000	reserved
3	RW	0x0	gpio0c3_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
2	RW	0x0	gpio0c2_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
1	RW	0x0	gpio0c1_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
0	RW	0x0	gpio0c0_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled

PMUGRF_PMU_SOC_CON0

Address: Operational Base + offset (0x0100)

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

Bit	Attr	Reset Value	Description
13	RW	0x0	ddrphy_bufferen_core 1'b0: Enable DDRPHY io retention 1'b1: Disable DDRPHY io retention
12	RW	0x0	ddrphy_bufferen_sel DDRPHY bufferen selection. 1'b0: Select from pmu and ddr_fail_safe 1'b1: Select from ddrphy_bufferen_core
11:10	RO	0x0	reserved
9	RW	0x0	pmic_sleep_sel PMIC_SLEEP source selection. 1'b0: npor_out2chip_rst, from reset pulse generator, can reset external PMIC. 1'b1: pmu_sleep, from PMU block, only support sleep function for external PMIC.
8:7	RO	0x0	reserved
6	RW	0x0	uart1_cts_inv uart1_cts polarity selection. 1'b0: Low active 1'b1: High active
5	RW	0x0	uart1_rts_inv uart1_rts polarity selection. 1'b0: Low active 1'b1: High active
4:1	RO	0x0	reserved
0	RW	0x0	clk32k_ioe 32KHz clock output enable. 1'b0: Output disable 1'b1: Output enable

PMUGRF_PMU_SOC_CON1

Address: Operational Base + offset (0x0104)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	hold_preset_uart1 Hold preset_uart1. When enabled, the relative logic cannot be reset. 1'b0: Disable 1'b1: Enable
14	RW	0x0	hold_preset_spi0 Hold preset_spi0. When enabled, the relative logic cannot be reset. 1'b0: Disable 1'b1: Enable
13	RW	0x0	hold_preset_pwm1 Hold preset_pwm1. When enabled, the relative logic cannot be reset. 1'b0: Disable 1'b1: Enable

Bit	Attr	Reset Value	Description
12	RW	0x0	hold_preset_pwm0 Hold preset_pwm0. When enabled, the relative logic cannot be reset. 1'b0: Disable 1'b1: Enable
11	RW	0x0	hold_preset_pmusgrf Hold preset_pmusgrf. When enabled, the relative logic cannot be reset. 1'b0: Disable 1'b1: Enable
10	RW	0x0	hold_preset_pmusgrf_remap Hold preset_pmusgrf_remap. When enabled, the relative logic cannot be reset. 1'b0: Disable 1'b1: Enable
9	RW	0x0	hold_preset_pmupvtm Hold preset_pmupvtm. When enabled, the relative logic cannot be reset. 1'b0: Disable 1'b1: Enable
8	RW	0x0	hold_preset_pmumem Hold preset_pmumem. When enabled, the relative logic cannot be reset. 1'b0: Disable 1'b1: Enable
7	RW	0x0	hold_preset_pmugrf Hold preset_pmugrf. When enabled, the relative logic cannot be reset. 1'b0: Disable 1'b1: Enable
6	RW	0x0	hold_preset_pmucru Hold preset_pmucru. When enabled, the relative logic cannot be reset. 1'b0: Disable 1'b1: Enable
5	RW	0x0	hold_preset_pdpmu_biu Hold preset_pdpmu_biu. When enabled, the relative logic cannot be reset. 1'b0: Disable 1'b1: Enable
4	RW	0x0	hold_preset_i2c2 Hold preset_i2c2. When enabled, the relative logic cannot be reset. 1'b0: Disable 1'b1: Enable
3	RW	0x0	hold_preset_i2c0 Hold preset_i2c0. When enabled, the relative logic cannot be reset. 1'b0: Disable 1'b1: Enable
2	RW	0x0	hold_preset_gpio0 Hold preset_gpio0. When enabled, the relative logic cannot be reset. 1'b0: Disable 1'b1: Enable

Bit	Attr	Reset Value	Description
1	RW	0x0	hold_preset_chipverotp Hold preset_chipverotp. When enabled, the relative logic cannot be reset. 1'b0: Disable 1'b1: Enable
0	RW	0x0	hold_dbreset_gpio0 Hold dbreset_gpio0. When enabled, the relative logic cannot be reset. 1'b0: Disable 1'b1: Enable

PMUGRF_PMU_SOC_CON2

Address: Operational Base + offset (0x0108)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:13	RO	0x0	reserved
12:8	RW	0x00	reset_width The chip output reset width, equals to reset_width*256
7	RW	0x0	hold_sreset_uart1 Hold sreset_uart1. When enabled, the relative logic cannot be reset. 1'b0: Disable 1'b1: Enable
6	RW	0x0	hold_reset_spi0 Hold reset_spi0. When enabled, the relative logic cannot be reset. 1'b0: Disable 1'b1: Enable
5	RW	0x0	hold_reset_pwm1 Hold reset_pwm0. When enabled, the relative logic cannot be reset. 1'b0: Disable 1'b1: Enable
4	RW	0x0	hold_reset_pwm0 Hold reset_pwm0. When enabled, the relative logic cannot be reset. 1'b0: Disable 1'b1: Enable
3	RW	0x0	hold_reset_pmupvtm Hold reset_pmupvtm. When enabled, the relative logic cannot be reset. 1'b0: Disable 1'b1: Enable
2	RW	0x0	hold_reset_i2c2 Hold reset_i2c2. When enabled, the relative logic cannot be reset. 1'b0: Disable 1'b1: Enable
1	RW	0x0	hold_reset_i2c0 Hold reset_i2c0. When enabled, the relative logic cannot be reset. 1'b0: Disable 1'b1: Enable

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

PMUGRF_PMU_SOC_CON3

Address: Operational Base + offset (0x010C)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RO	0x0	reserved
11	RW	0x0	pmu_pwr_idlereq PMU pwr_idlereq software enable. 1'b0: Disable 1'b1: Enbale
10	RW	0x0	clk_gpll_ls_sel GPLL clock path selection. 1'b0: Normal 1'b1: Through level-shift cell
9	RW	0x0	clk_cpll_ls_sel CPLL clock path selection. 1'b0: Normal 1'b1: Through level-shift cell
8	RW	0x0	upctl_c_sysreq_cfg 1'b0: After DDR failsafe module enters self-refresh status, then request DDR controller to enter low power state. 1'b1: Always enable requesting DDR controller to enter low power state, when DDR failsafe module is working.
7	RW	0x0	ddr_io_ret_oen_cfg ddr_io_ret_oen_cfg bit control. 1'b0: ddr_io_ret output enable 1'b1: ddr_io_ret_output disable
6	RW	0x0	ddr_io_ret_cfg 1'b0: DDR IO retention managed by hardware automatically 1'b1: Enable DDR IO retention manually
5	RW	0x0	ddr_io_ret_de_req 1'b0: DDR IO retention managed by hardware automatically 1'b1: Enable IO retention manually
4	RW	0x1	ddrc_gating_en 1'b0: Disable DDR clock gating during system failure 1'b1: Enable DDR clock gating during system failure
3	RW	0x1	sref_enter_en 1'b0: Disable DDR self-refresh enter when system is failed 1'b1: Enable DDR self-refresh enter when system is failed
2	RW	0x1	ddrio_ret_en 1'b0: Remain DDR IO status when system is failed 1'b1: Enable DDR IO retention when system is failed
1	RW	0x0	wdt_reset_trigger_en Enable failsafe wdt_reset input. 1'b0: Disable 1'b1: Enable

Bit	Attr	Reset Value	Description
0	RW	0x0	tsadc_shut_reset_trigger_en Enable failsafe tsadc_shut input. 1'b0: Disable 1'b1: Enable

PMUGRF_PMU_SOC_CON4

Address: Operational Base + offset (0x0110)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13:0	RW	0x2805	pmusram_spracfg SPRF memory EMA configuration for PMU_SRAM. bit [0]: RME bit [3:1]: RM bit [4]: LS bit [5]: BC1 bit [6]: BC2 bit [7]: TEST1 bit [10:8]: WPULSE bit [13:11]: RM

PMUGRF_IOFUNC_CON4

Address: Operational Base + offset (0x0114)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14	RW	0x0	pwm7_iomux_sel PWM7 IOMUX selection. 1'b0: IOMUX group 0 (PWM7*M0) selected 1'b1: IOMUX group 1 (PWM7*M1) selected
13	RO	0x0	reserved
12	RW	0x0	pwm6_iomux_sel PWM6 IOMUX selection. 1'b0: IOMUX group 0 (PWM6*M0) selected 1'b1: IOMUX group 1 (PWM6*M1) selected
11	RO	0x0	reserved
10	RW	0x0	pwm5_iomux_sel PWM5 IOMUX selection. 1'b0: IOMUX group 0 (PWM5*M0) selected 1'b1: IOMUX group 1 (PWM5*M1) selected
9	RO	0x0	reserved
8	RW	0x0	pwm4_iomux_sel PWM4 IOMUX selection. 1'b0: IOMUX group 0 (PWM4*M0) selected 1'b1: IOMUX group 1 (PWM4*M1) selected
7	RO	0x0	reserved

Bit	Attr	Reset Value	Description
6	RW	0x0	pwm3_iomux_sel PWM3 IOMUX selection. 1'b0: IOMUX group 0 (PWM3*M0) selected 1'b1: IOMUX group 1 (PWM3*M1) selected
5	RO	0x0	reserved
4	RW	0x0	pwm2_iomux_sel PWM2 IOMUX selection. 1'b0: IOMUX group 0 (PWM2*M0) selected 1'b1: IOMUX group 1 (PWM2*M1) selected
3	RO	0x0	reserved
2	RW	0x0	pwm1_iomux_sel PWM1 IOMUX selection. 1'b0: IOMUX group 0 (PWM1*M0) selected 1'b1: IOMUX group 1 (PWM1*M1) selected
1	RO	0x0	reserved
0	RW	0x0	pwm0_iomux_sel PWM0 IOMUX selection. 1'b0: IOMUX group 0 (PWM0*M0) selected 1'b1: IOMUX group 1 (PWM0*M1) selected

PMUGRF_IOFUNC_CON5

Address: Operational Base + offset (0x0118)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	pwm7_switch_sel PWM7 output selection. 1'b0: From PWM module 1'b1: From PMU module
14	RW	0x0	pwm6_switch_sel PWM6 output selection. 1'b0: From PWM module 1'b1: From PMU module
13	RW	0x0	pwm5_switch_sel PWM5 output selection. 1'b0: From PWM module 1'b1: From PMU module
12	RW	0x0	pwm4_switch_sel PWM4 output selection. 1'b0: From PWM module 1'b1: From PMU module
11	RW	0x0	pwm3_switch_sel PWM3 output selection. 1'b0: From PWM module 1'b1: From PMU module
10	RW	0x0	pwm2_switch_sel PWM2 output selection. 1'b0: From PWM module 1'b1: From PMU module
9	RW	0x0	pwm1_switch_sel PWM1 output selection. 1'b0: From PWM module 1'b1: From PMU module

Bit	Attr	Reset Value	Description
8	RW	0x0	pwm0_switch_sel PWM0 output selection. 1'b0: From PWM module 1'b1: From PMU module
7:5	RO	0x0	reserved
4	RW	0x0	i2c2_iomux_sel I2C2 Controller selection. 1'b0: Reserved 1'b1: Controlled by I2C
3	RO	0x0	reserved
2	RW	0x0	uart1_iomux_sel UART1 IOMUX selection. 1'b0: IOMUX group 0 (UART1*M0) selected 1'b1: IOMUX group 1 (UART1*M0) selected
1:0	RW	0x0	spi0_iomux_sel SPI0 IOMUX selection. 2'b00: IOMUX group 0 (SPI0*M0) selected 2'b01: IOMUX group 1 (SPI0*M1) selected 2'b01: IOMUX group 2 (SPI0*M2) selected 2'b11: Reserved

PMUGRF_IO_VSEL

Address: Operational Base + offset (0x0140)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:10	RO	0x00	reserved
9	RW	0x0	pmuio1_vsel PMUIO1 voltage selection. 1'b0: 3.3V 1'b1: 1.8V
8	RW	0x0	pmuio0_vsel PMUIO0 voltage selection. 1'b0: 3.3V 1'b1: 1.8V
7	RW	0x0	vccio7_vsel VCCIO7 voltage selection. 1'b0: 3.3V 1'b1: 1.8V
6	RW	0x0	vccio6_vsel VCCIO6 voltage selection. 1'b0: 3.3V 1'b1: 1.8V
5	RW	0x0	vccio5_vsel VCCIO5 voltage selection. 1'b0: 3.3V 1'b1: 1.8V
4	RW	0x0	vccio4_vsel VCCIO4 voltage selection. 1'b0: 3.3V 1'b1: 1.8V

Bit	Attr	Reset Value	Description
3	RW	0x0	vccio3_vsel VCCIO3 voltage selection. 1'b0: 3.3V 1'b1: 1.8V
2	RW	0x0	vccio2_vsel VCCIO2 voltage selection. 1'b0: 3.3V 1'b1: 1.8V
1	RW	0x0	vccio1_vsel VCCIO1 voltage selection. 1'b0: 3.3V 1'b1: 1.8V
0	RW	0x0	flash_vosel VCCIO1 IO domain voltage source selection. 1'b0: Controlled by GPIO0B3 1'b1: Controlled by GRF_IO_VSEL[1]

PMUGRF_IO_VRET

Address: Operational Base + offset (0x0144)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x03	reserved
7	RW	0x1	vccio7_ret VCCIO7 retention control. 1'b0: Retention mode 1'b1: Normal mode
6	RW	0x1	vccio6_ret VCCIO6 retention control. 1'b0: Retention mode 1'b1: Normal mode
5	RW	0x1	vccio5_ret VCCIO5 retention control. 1'b0: Retention mode 1'b1: Normal mode
4	RW	0x1	vccio4_ret VCCIO4 retention control. 1'b0: Retention mode 1'b1: Normal mode
3	RW	0x1	vccio3_ret VCCIO3 retention control. 1'b0: Retention mode 1'b1: Normal mode
2	RW	0x1	vccio2_ret VCCIO2 retention control. 1'b0: Retention mode 1'b1: Normal mode
1	RW	0x1	vccio1_ret VCCIO1 retention control. 1'b0: Retention mode 1'b1: Normal mode
0	RO	0x0	reserved

PMUGRF_PMUPVTM_CLKDIV

Address: Operational Base + offset (0x0180)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7:2	RW	0x00	pmupvtm_clkout_div Divider for PVTM output clock to generate 32KHz clock. Frequency is divided by (pmupvtm_clkout_div*4+1).
1:0	RO	0x0	reserved

PMUGRF_OS_REG0

Address: Operational Base + offset (0x0200)

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

PMUGRF_OS_REG1

Address: Operational Base + offset (0x0204)

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

PMUGRF_OS_REG2

Address: Operational Base + offset (0x0208)

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

PMUGRF_OS_REG3

Address: Operational Base + offset (0x020C)

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

PMUGRF_OS_REG4

Address: Operational Base + offset (0x0210)

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

PMUGRF_OS_REG5

Address: Operational Base + offset (0x0214)

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

PMUGRF_OS_REG6

Address: Operational Base + offset (0x0218)

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

PMUGRF_OS_REG7

Address: Operational Base + offset (0x021C)

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

PMUGRF_OS_REG8

Address: Operational Base + offset (0x0220)

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

PMUGRF_OS_REG9

Address: Operational Base + offset (0x0224)

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

PMUGRF_OS_REG10

Address: Operational Base + offset (0x0228)

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

PMUGRF_OS_REG11

Address: Operational Base + offset (0x022C)

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

PMUGRF_RSTFUNC_STATUS

Address: Operational Base + offset (0x0230)

Bit	Attr	Reset Value	Description
31:4	RO	0x0000000	reserved
3	RO	0x0	ddr_fail_safe_src When high, ddr_fail_safe is active.
2	RO	0x0	tsadc_shut_reset_src When high, reset by TSADC shut trigger.
1	RO	0x0	wdt_reset_src When high, reset by WDT trigger.
0	RO	0x0	first_reset_src When high, reset by first reset trigger.

PMUGRF_RSTFUNC_CLR

Address: Operational Base + offset (0x0234)

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

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

PMUGRF_SD_DETECT_CON

Address: Operational Base + offset (0x0380)

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

PMUGRF_SD_DETECT_STATUS

Address: Operational Base + offset (0x0390)

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

PMUGRF_SD_DETECT_CLR

Address: Operational Base + offset (0x03A0)

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

PMUGRF_SD_DETECT_COUNT

Address: Operational Base + offset (0x03B0)

Bit	Attr	Reset Value	Description
31:20	RO	0x000	reserved
19:0	RW	0x003e8	sdmmc_det_count SDMMC detect pin filter counter, in the unit of pclk.

Chapter 7 Timer

7.1 Overview

Timer is a programmable timer peripheral. This component is an APB slave device. In *RV1109/RV1126* there are 6 timers (Timer0~5) and 2 Secure timers (STimer0~2). CNTP_TVAL is provided by STimer1.

Timer5 and STimer0~1 count up from zero to a programmed value and generate an interrupt when the counter reaches the programmed value.

Timer0~4 count down from a programmed value to zero and generate an interrupt when the counter reaches zero.

Timer supports the following features:

- Timer0~5 is used for no-secure, STimer0~1 is used for secure.
- Two operation modes: free-running and user-defined count.
- Interrupt can be enabled or disabled individual for each timer.

7.2 Block Diagram

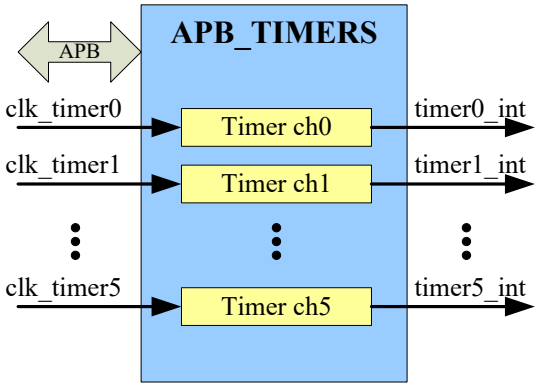


Fig. 7-1 Timer Block Diagram

The above figure shows the architecture of the APB timers (include six programmable timer channels). The Stimers that in the bus subsystem only include two programmable timer channels.

7.3 Function Description

7.3.1 Timer clock

TIMER0~TIMER5 and STIMER0~1 are in the pd_bus subsystem. The clock source is 24MHz.

7.3.2 Programming sequence

1. Initialize the timer by the TIMERn_CONTROLREG (0≤n≤5) register:

- Disable the timer by writing a "0" to the timer enable bit (bit 0). Accordingly, the timer_en output signal is de-asserted.
- Program the timer mode—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≤n≤5) and TIMERn_LOAD_COUNT0 (0≤n≤5) register.

3. Enable the timer by writing a "1" to bit 0 of TIMERn_CONTROLREG (0≤n≤5).

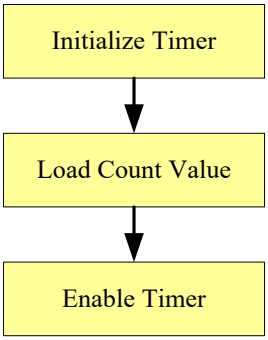


Fig. 7-2 Timer Usage Flow

7.3.3 Loading a timer count value

For the descending Timers(Timer0~4).The initial value for each timer—that is, the value from which it counts down—is loaded into the timer using the load count register (TIMERn_LOAD_COUNT1 and TIMERn_ LOAD_COUNT0). Two events can cause a timer to load the initial value from its load count register:

- Timer is enabled after reset or disabled.
- Timer counts down to 0, when timer is configured into free-running mode.

For the incremental Timers(Timer5 and STimer0~1).The initial value for each timer is zero. The count register will count up to the value loaded in the register TIMERn_LOAD_COUNT1 and TIMERn_ LOAD_COUNT0. Two events can cause a timer to load zero:

- Timer is enabled after reset or disabled.
- Timer counts up to the value stored in TIMERn_LOAD_COUNT1 and TIMERn_ LOAD_COUNT0, when timer is configured into free-running mode.

7.3.4 Timer mode selection

- User-defined count mode – Timer loads TIMERn_LOAD_COUNT1 and TIMERn_ LOAD_ COUNT0 registers (for descending timers) or zero (for incremental timers) as initial value. When the timer counts down to 0 (for descending timers) or counts up to the value in TIMERn_LOAD_COUNT1 and TIMERn_ LOAD_ COUNT0 (for incremental timers), it will not automatically reload the COUNT register. User need to disable timer firstly and follow the programming sequence to make timer work again.
- Free-running mode – Timer loads the TIMERn_LOAD_COUNT1 and TIMERn_LOAD_COUNT0(for descending timers) or zero (for incremental timers) register as initial value. Timer will automatically reload the COUNT register, when timer counts down to 0 (for descending timers) or counts up to the value in TIMERn_LOAD_COUNT1 and TIMERn_LOAD_COUNT0 (for incremental timers).

7.4 Register Description

7.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
TIMER_TIMERn_LOAD_COUNT0	0x0000	W	0x00000000	Low 32 bits value to be loaded into Timer0
TIMER_TIMERn_LOAD_COUNT1	0x0004	W	0x00000000	High 32 bits value to be loaded into Timer0.
TIMER_TIMERn_CURR_VALUE0	0x0008	W	0x00000000	Low 32 bits of current value of Timer0.
TIMER_TIMERn_CURR_VALUE1	0x000c	W	0x00000000	High 32 bits of current value of Timer0.
TIMER_TIMERn_CONTROL	0x0010	W	0x00000000	Timer Control Register
TIMER_TIMERn_INTSTATUS	0x0018	W	0x00000000	Timer Interrupt Stauts Register
TIMER_REVISION_6CH	0x00f0	W	0x11970006	timer version for 6 channel timer

Name	Offset	Size	Reset Value	Description
TIMER_REVISION_2CH	0x00f0	W	0x11970002	timer version for 2 channel stimer

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

7.4.2 Detail Register Description

TIMER_TIMERn_LOAD_COUNT0

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	count0 Load count low bits

TIMER_TIMERn_LOAD_COUNT1

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	count1 Load count high bits

TIMER_TIMERn_CURR_VALUE0

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	current_value0 current_cnt_low bits

TIMER_TIMERn_CURR_VALUE1

Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	current_value1 current_cnt_high bits

TIMER_TIMERn_CONTROL

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:3	RO	0x00000000	reserved
2	RW	0x0	int_en Timer interrupt enable 1'b0: Disable 1'b1: Enable
1	RW	0x0	timer_mode Timer mode 1'b0: Free-running mode 1'b1: User-defined mode
0	RW	0x0	timer_en Timer enable 1i®b0: Disable 1'b1: Enable

TIMER_TIMERn_INTSTATUS

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	WO	0x0	int_pd This register contains the interrupt status for Timer. Write 1 to this register will clear the interrupt.

TIMER_REVISION_6CH

Address: Operational Base + offset (0x00f0)

Bit	Attr	Reset Value	Description
31:16	RO	0x1197	svn_revision svn revision 16'd4503
15:0	RO	0x0006	ip_function Channel number 16'd6: Six channel

TIMER_REVISION_2CH

Address: Operational Base + offset (0x00f0)

Bit	Attr	Reset Value	Description
31:16	RO	0x1197	svn_revision SVN revision 16'd4503
15:0	RO	0x0002	ip_function Channel number 16'd2: Two channel

7.5 Application Notes

7.5.1 Register Base Address

Table 7-1 Register Base Address

Module	CNT	Base Addr
Timer	TIMER0_BASE	0xFF660000
	TIMER1_BASE	0xFF660020
	TIMER2_BASE	0xFF660040
	TIMER3_BASE	0xFF660060
	TIMER4_BASE	0xFF660080
	TIMER5_BASE	0xFF6600A0
	TIMER_REVISION	0xff6600F0
Stimer	STIMER0_BASE	0xFF670000
	STIMER1_BASE	0xFF670020
	STIMER_REVISION	0xFF6700F0

7.5.2 Clock and Enable

In the chip, the timer_clk is from 24MHz XIN_OSC, asynchronous to the pclk. When user disables the timer enables bit (bit 0 of TIMERn_CONTROLREG (0⩽n⩽5)), the timer en output signal is de-asserted, and timer_clk will stop. When user enables the timer, the timer_en signal is asserted and timer_clk will start running.

The application is only allowed to re-config registers when timer_en is low.

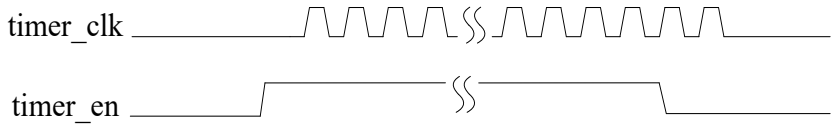


Fig. 7-3 Timing between timer_en and timer_clk

Please refer to function description section for the timer usage flow.

Chapter 8 Watchdog Timer(WDT)

8.1 Overview

Watchdog Timer (WDT) is an APB slave peripheral that can be used to prevent system lockup that caused by conflicting parts or programs in a SOC. The WDT would generate interrupt or reset signal when its counter reaches zero, then a reset controller would reset the system.

WDT supports the following features:

- 32 bits APB bus width
- WDT counter's clock 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 one watchdog in LOGIC named WDT0, and one watchdog in DSP named WDT1
- Both WDT0 and WDT1 can drive CRU to generate global software reset

8.2 Block Diagram

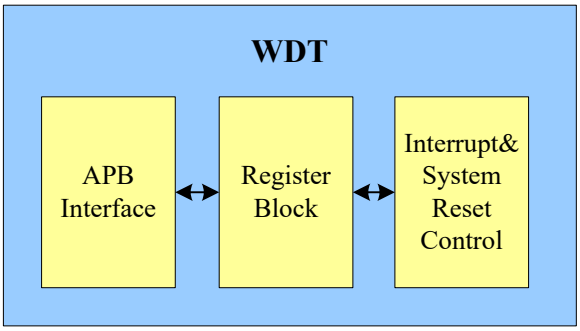


Fig. 8-1 WDT Block Diagram

WDT comprises with:

- APB Interface

The APB Interface implements the APB slave operation. Its data bus width is 32 bits.

- Register Block

A register block that read coherence for the current count register.

- Interrupt & System Reset Control

An interrupt/system reset generation block is comprised of a decrementing counter and control logic.

8.3 Function Description

8.3.1 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.

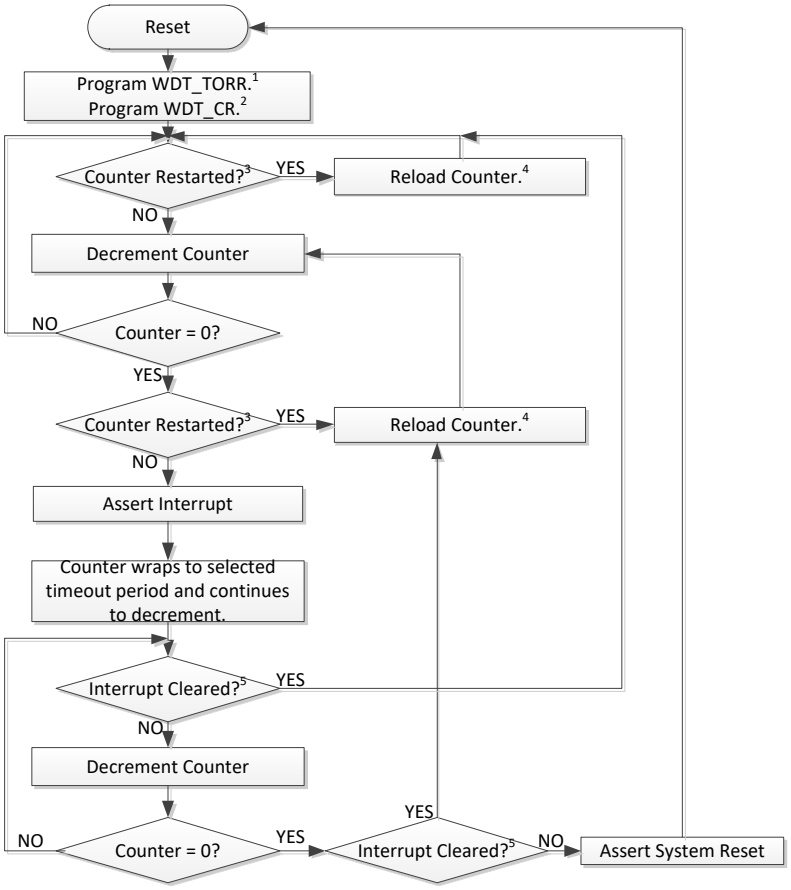
System Resets

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.

8.3.2 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. 8-2 WDTOperation Flow (RMOD=1)

8.4 Register Description

This section describes the control/status registers of the design. Software should read and write these registers using 32-bits accesses. There are two WDTs (WDT0 ~ WDT1), and each of them has same register group. Therefore, two WDTs' register groups have two different base addresses.

8.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
WDT_CR	0x0000	W	0x0000000a	Control Register
WDT_TORR	0x0004	W	0x00000000	Timeout Range Register
WDT_CCVR	0x0008	W	0x00000000	Current Counter Value Register
WDT_CRR	0x000c	W	0x00000000	Counter Restart Register
WDT_STAT	0x0010	W	0x00000000	Interrupt Status Register
WDT_EOI	0x0014	W	0x00000000	Interrupt Clear Register

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

8.4.2 Detail Register Description

WDT_CR

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:5	RO	0x0000000	reserved
4:2	RW	0x2	rst_pluse_length This is used to select the number of pclk cycles for which the system reset stays asserted. 3'b000: 2 pclk cycles 3'b001: 4 pclk cycles 3'b010: 8 pclk cycles 3'b011: 16 pclk cycles 3'b100: 32 pclk cycles 3'b101: 64 pclk cycles 3'b110: 128 pclk cycles 3'b111: 256 pclk cycles
1	RW	0x1	resp_mode Selects the output response generated to a timeout. 1'b0: Generate a system reset. 1'b1: First generate an interrupt and if it is not cleared by the time a second timeout occurs then generate a system reset.
0	RW	0x0	wdt_en Writable when the configuration parameter WDT_ALWAYS_EN=0, otherwise,it is readable. This bit is used to enable and disable the DW_apb_wdt.When disabled, the counter dose not decrement .Thus, no interrupt or system reset are generated. Once this bit has been enabled, it can be cleared only by a system reset. 1'b0: WDT disabled. 1'b1: WDT enabled.

WDT_TORR

Address: Operational Base + offset (0x0004)

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

Bit	Attr	Reset Value	Description
3:0	RW	0x0	timeout_period This field is used to select the timeout period from which the watchdog counter restarts. A change of the timeout period takes effect only after the next counter restart (kick). The range of values available for a 32-bit watchdog counter are: 4'b0000: 0x0000ffff 4'b0001: 0x0001ffff 4'b0010: 0x0003ffff 4'b0011: 0x0007ffff 4'b0100: 0x000fffff 4'b0101: 0x001fffff 4'b0110: 0x003fffff 4'b0111: 0x007fffff 4'b1000: 0x00ffffff 4'b1001: 0x01ffffff 4'b1010: 0x03ffffff 4'b1011: 0x07ffffff 4'b1100: 0x0fffffff 4'b1101: 0x1fffffff 4'b1110: 0x3fffffff 4'b1111: 0x7fffffff

WDT_CCVR

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	cur_cnt This register, when read, is the current value of the internal counter. This value is read coherently whenever it is read

WDT_CRR

Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	WO	0x00	cnt_restart This register is used to restart the WDT counter. As a safety feature to prevent accidental restarts, the value 0x76 must be written. A restart also clears the WDT interrupt. Reading this register returns zero.

WDT_STAT

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved

Bit	Attr	Reset Value	Description
0	RO	0x0	wdt_status This register shows the interrupt status of the WDT. 1'b1: Interrupt is active regardless of polarity. 1'b0: Interrupt is inactive.

WDT_EOI

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RO	0x0	wdt_int_clr This can be used to clear the interrupt without restarting the watchdog counter.

8.5 Application Notes

- Configure CRU_GLB_RST_CON to enable WDT triggered global software reset. Please refer to Chapter 3 for more information.

Chapter 9 Power Management Unit (PMU)

9.1 Overview

In order to meet low power requirements, a power management unit (PMU) is designed for controlling power resources in RV1109/RV1126. The RV1109/RV1126 PMU is dedicated for managing the power of the whole chip.
PMU supports the following features:

- Support multi voltage domains: VD_CORE, VD_LOGIC, VD_PMU
- Support multi power domains in VD_CORE: PD_CPU_0, PD_CPU_1, PD_CPU_2, PD_CPU_3
- Support multi power domains in VD_LOGIC: PD_VDPU, PD_VI, PD_VO, PD_ISPP, PD_NVM, PD_SDIO, PD_USB, PD_DDR, PD_USB, PD_CRYPT
- Support PD_NPU and PD_VEPU act as power domain or voltage domain
- Support BIU idle operations: BIU_DDR, BIU_BUSHOLD, BIU_BUSNOC1, BIU_BUSNOC2, BIU_BUSNOC3, BIU_AUDIO, BIU_VI, BIU_VO, BIU_ISPP, BIU_VEPU, BIU_VDPU, BIU_NVM, BIU_SDCARD, BIU_SDIO, BIU_GMAC, BIU_USB, BIU_PHP, BIU_PMU, BIU_NPU, BIU_TOPAPB, BIU_CRYPT
- Support CPU auto power down and SCU auto power down
- Support power down/up all power domains by software or hardware
- Support power down/up all voltage domains by software or hardware
- Support to send idle request to BIU
- Support global interrupt disable in low power mode
- Support low frequency clock source from PVTM
- Support PMU clock switch to low frequency clock in low power mode
- Support PLLs power down/up by hardware in low power mode
- Support OSC enable/disable request in low power mode
- Support to clamp all VD_PMU input before power off VD_LOGIC in low power mode
- Support wakeup reset control in power off mode
- Support DDR self-refresh in low power mode
- Support DDR controller clock auto gating in low power mode
- Support varies configurable wakeup source for low power mode

9.2 Block Diagram

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

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

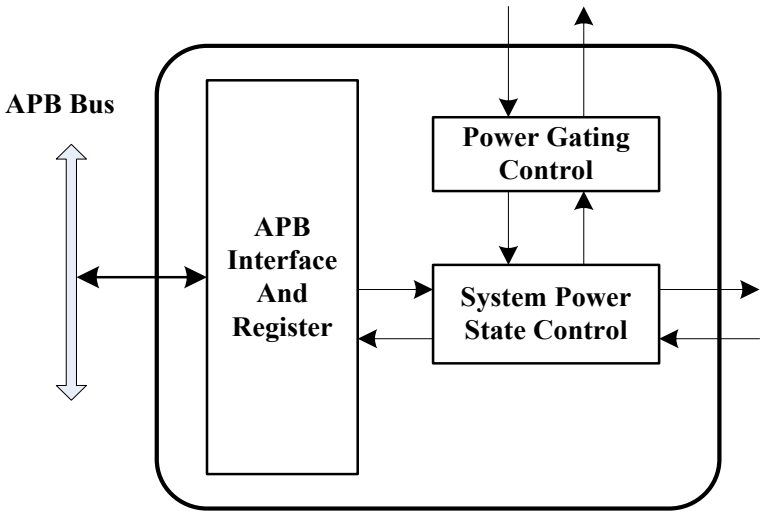


Fig. 9-1 PMU Bock Diagram

9.3 Function Description

9.3.1 Domain Partition

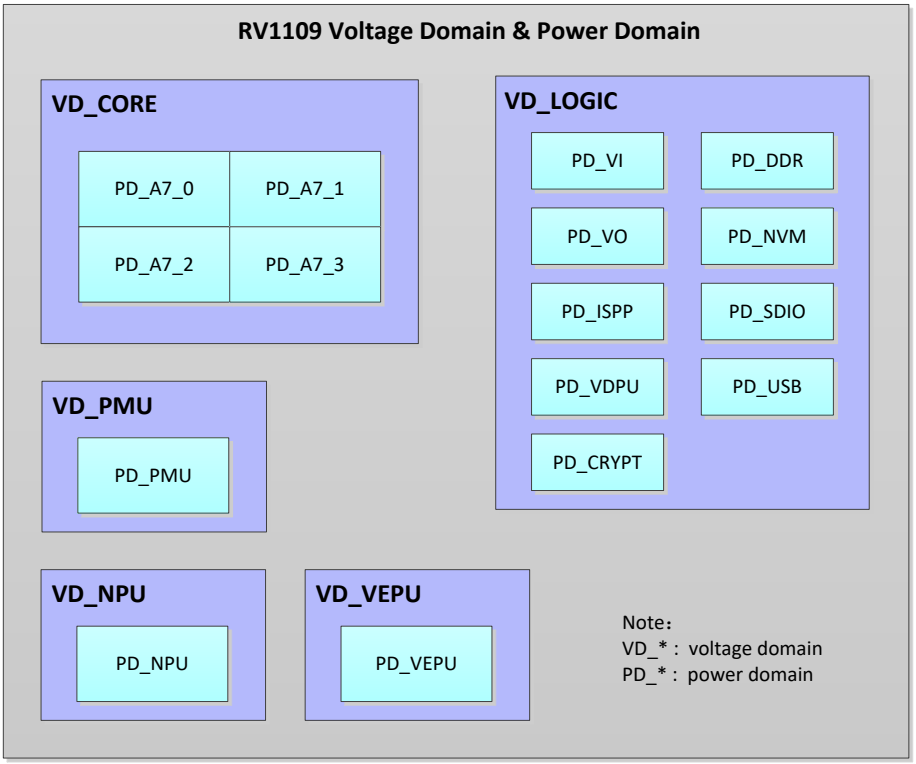


Fig. 9-2 RV1109/RV1126 Voltage Domain and Power Domain Partition

The above diagram describes the power domain and voltage domain partition, and the following table lists IPs in every power domain.

Table 9-1 RV1109/RV1126 Voltage Domain and Power Domain Summary

Voltage Domain	Power Domain	Description
VD_CORE	PD_CPU_0	CPU CORE 0
	PD_CPU_1	CPU CORE 1
	PD_CPU_2	CPU CORE 2
	PD_CPU_3	CPU CORE 3
	ALIVE	BIU_CPU SCU,L2,GIC400 DAP PVTM
VP_PMU	PD_PMU	BIU_PMU GPLL PMU_SRAM SGRF PMU PMUCRU PMUGRF I2C0, I2C2 GPIO0 PVTM UART1 PWM0, PWM1 SPI0
VD_NPU	PD_NPU	BIU_NPU VIPNN PVTM

Voltage Domain	Power Domain	Description
VD_VEPU	PD_VEPU	BIU_VEPU RKVENC
VD_LOGIC	PD_VI	BIU_VI ISP CIF CIFLITE CSIHOST
	PD_VO	BIU_VO VOP RGA IEP DSIHOST
	PD_ISPP	BIU_ISPP ISPP
	PD_VDPU	BIU_VDPU VDEC JPEG
	PD_CRYPT	BIU_CRYPT CRYPTO
	PD_DDR	BIU_DDR DDR_UMCTL2 DDR_DFICTL DDR_MONITOR DDR_STANDBY DDR_SCRAMBLE DDR_HOLD AXI_SPLIT DDR_GRF DDR
	PD_NVM	BIU_NVM EMMC NANDC FSPI
	PD_SDIO	BIU_SDIO SDIO
	PD_USB	BIU_USB USBHOST USBOTG
	ALIVE	BIU_SDCARD SDCARD
		BIU_AUDIO I2S0~2 PDM audioPWM ACDC_DIG
		BIU_GMAC GMAC

Voltage Domain	Power Domain	Description
		BIU_BUS MCU DCF SPINLOCK DECOM DMAC SYSTEM_SRAM BootRom MAILBOX GRF, SGRF OTPC I2C1/3/4/5 TIMER_6CH, STIMER_2CH WDT, WDT_S GPIO1/2/3/4 SPI1 UART0/2/3/4/5 PWM2 CAN CPU_TSADC, NPU_TSADC
		DDRPHY
		CSIPHY0, CSIPHY1
		DSIPHY
		HOSTPHY, OTGPHY
		CPUPVT, NPUPVT
		SARADC PHY
		OPTPHY
		APLL, CPLL, DPLL, HPLL
		BIU_TOP
		CRU

9.3.2 Operation Mode

First of all, we define two operation modes of PMU, normal mode and low power mode.
When operating at normal mode, that means software can manage power sources directly by accessing PMU registers. For example, CPU can write PMU_PWR_GATE_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 be delayed to execute after FSM running in particular phase.
To enter low power mode, after setting some power configurations, the PMU_PWR_CON[0] bit must be set 1 to enable PMU FSM. Then CPU needs to execute a WFI command to perform ready signal. After PMU detects all CPUs in WFI status, the FSM will be fetched. And the specific power sources will be controlled during specific status in FSM. So the low power mode is a "delay affect" way to handle power sources inside the RV1109/RV1126 chip.

9.4 Register Description

9.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
PMU_VERSION	0x0000	W	0x030A0000	PMU version register
PMU_PWR_CON	0x0004	W	0x0000000C	PMU power control register
PMU_MAIN_PWR_STATE	0x0008	W	0x00000000	PMU power status register

Name	Offset	Size	Reset Value	Description
PMU_INT_MASK_CON	0x000C	W	0x00000000	Interrupt mask control register
PMU_WAKEUP_INT_CON	0x0010	W	0x00000000	Wakeup interrupt control register
PMU_WAKEUP_INT_ST	0x0014	W	0x00000000	Wakeup interrupt status register
PMU_WAKEUP_EDGE_CON	0x0018	W	0x00000000	Edge wakeup source control register
PMU_WAKEUP_EDGE_ST	0x001C	W	0x00000000	Edge wakeup source status register
PMU_SCU_STABLE_CNT	0x0020	W	0x000FFFFF	SCU power stable counter register
PMU_PMIC_STABLE_CNT	0x0024	W	0x000FFFFF	PMIC stable counter register
PMU_OSC_STABLE_CNT	0x0028	W	0x000FFFFF	OSC stable counter register
PMU_WAKEUP_RSTCLR_CNT	0x002C	W	0x000FFFFF	Wakeup reset clear counter register
PMU_PLL_LOCK_CNT	0x0030	W	0x000FFFFF	PLL lock counter register
PMU_SCU_PWRUP_CNT	0x0038	W	0x00005DC0	SCU power up stable counter register
PMU_SCU_PWRDN_CNT	0x003C	W	0x00005DC0	SCU power down stable counter register
PMU_SCU_VOLUP_CNT	0x0040	W	0x00005DC0	SCU voltage up stable counter register
PMU_SCU_VOLDN_CNT	0x0044	W	0x00005DC0	SCU voltage down stable counter register
PMU_WAKEUP_TIMEOUT_CNT	0x0048	W	0x00005DC0	WAKEUP timeout counter register
PMU_PWM_SWITCH_CNT	0x004C	W	0x000FFFFF	PWM switch stable counter register
PMU_SCU_PWR_CON	0x0080	W	0x00000000	SCU power hardware control register
PMU_SCU_PWR_SFTCON	0x0084	W	0x00000000	SCU power software control register
PMU_SCU_PWR_STATE	0x008C	W	0x00000000	SCU power status register
PMU_CPU_PWR_SFTCON	0x0094	W	0x00000000	SCU software power control register
PMU_CLUSTER_PWR_ST	0x009C	W	0x00000000	Cluster power status register
PMU_CLUSTER_IDLE_CON	0x00A0	W	0x00000000	Cluster idle hardware control register
PMU_CLUSTER_IDLE_SFTCON	0x00A4	W	0x00000000	Cluster idle software control register
PMU_BUS_IDLE_CON0	0x00B0	W	0x00000000	Bus idle request hardware control register 0
PMU_BUS_IDLE_CON1	0x00B4	W	0x00000000	Bus idle request hardware control register 1
PMU_BUS_IDLE_SFTCON0	0x00C0	W	0x00000000	Bus idle request software control register 0
PMU_BUS_IDLE_SFTCON1	0x00C4	W	0x00000000	Bus idle request software control register 1
PMU_BUS_IDLE_ACK	0x00D0	W	0x00000000	Bus idle acknowledge status register
PMU_BUS_IDLE_ST	0x00D8	W	0x00000000	Bus idle status register

Name	Offset	Size	Reset Value	Description
PMU_NOC_AUTO_CON0	0x00E0	W	0x00000000	NOC automatic power control register 0
PMU_NOC_AUTO_CON1	0x00E4	W	0x00000000	NOC automatic power control register 1
PMU_DDR_PWR_CON	0x00F0	W	0x00000000	DDR power hardware control register
PMU_DDR_PWR_SFTCON	0x00F4	W	0x00000000	DDR power software control register
PMU_DDR_PWR_STATE	0x00F8	W	0x00000000	DDR power state machine register
PMU_DDR_PWR_ST	0x00FC	W	0x00000000	DDR power status register
PMU_PWR_GATE_CON	0x0100	W	0x00000000	Power domain hardware control register
PMU_PWR_GATE_STATE	0x0104	W	0x00000000	Power domain state machine register
PMU_PWR_DWN_ST	0x0108	W	0x00000000	Power domain status register
PMU_PWR_GATE_SFTCON	0x0110	W	0x00000000	Power domain software control register
PMU_VOL_GATE_SFTCON	0x0118	W	0x00000000	Voltage domain software control register
PMU_CRU_PWR_CON	0x0120	W	0x00000000	CRU power hardware control register
PMU_CRU_PWR_SFTCON	0x0124	W	0x00000000	CRU power software control register
PMU_CRU_PWR_STATE	0x0128	W	0x00000000	CRU power state machine register
PMU_PLLPD_CON	0x0130	W	0x00000000	PLL power hardware control register
PMU_PLLPD_SFTCON	0x0134	W	0x00000000	PLL power software control register
PMU_INFO_TX_CON	0x0150	W	0x00000000	PMU information transmit control register
PMU_SYS_REG0	0x01C0	W	0x00000000	PMU system register 0
PMU_SYS_REG1	0x01C4	W	0x00000000	PMU system register 1
PMU_SYS_REG2	0x01C8	W	0x00000000	PMU system register 2
PMU_SYS_REG3	0x01CC	W	0x00000000	PMU system register 3
PMU_SYS_REG4	0x01D0	W	0x00000000	PMU system register 4
PMU_SYS_REG5	0x01D4	W	0x00000000	PMU system register 5
PMU_SYS_REG6	0x01D8	W	0x00000000	PMU system register 6
PMU_SYS_REG7	0x01DC	W	0x00000000	PMU system register 7

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

9.4.2 Detail Register Description

PMU_VERSION

Address: Operational Base + offset (0x0000)

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

PMU_PWR_CON

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	pmu_sleep_pol pmu_sleep polarity selection. 1'b0: High active 1'b1: Low active
14:8	RO	0x00	reserved
7	RW	0x0	cru_bypass Bypass CRU in low power procedure. 1'b0: Disable 1'b1: Enable
6	RW	0x0	pwrdn_bypass Bypass power domain in low power procedure. 1'b0: Disable 1'b1: Enable
5	RW	0x0	ddr_bypass Bypass DDR in low power procedure. 1'b0: Disable 1'b1: Enable
4	RW	0x0	bus_bypass Bypass bus idle in low power procedure. 1'b0: Disable 1'b1: Enable
3:2	RO	0x3	reserved
1	RW	0x0	scu_bypass Bypass SCU in low power procedure. 1'b0: Disable 1'b1: Enable
0	R/WSC	0x0	powermode_en Low power mode enable. When controller enters low power flow, this bit is automatically cleared. 1'b0: Disable 1'b1: Enable

PMU_MAIN_PWR_STATE

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:4	RO	0x0000000	reserved
3:0	RO	0x0	pmu_power_state PMU main power state. 4'h0: Normal state 4'h1: SCU low power state 4'h3: Bus low power state 4'h4: DDR low power state 4'h5: Power gating low power state 4'h6: Clock and reset low power state 4'h7: sleep state 4'h8: Clock and reset active state 4'h9: Power gating active state 4'ha: DDR active state 4'hb: Bus active state 4'hd: SCU active state Others: Reserved

PMU_INT_MASK_CON

Address: Operational Base + offset (0x000C)

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

PMU_WAKEUP_INT_CON

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:11	RO	0x000000	reserved
10	RW	0x0	wakeup_timerout_en Enable PMU timeout interrupt as wakeup source. 1'b0: Disable 1'b1: Enable
9	RW	0x0	wakeup_sys_int_en Enable system interrupt source as wakeup source. 1'b0: Disable 1'b1: Enable
8	RW	0x0	wakeup_uart1_en Enable UART1 interrupt as wakeup source. 1'b0: Disable 1'b1: Enable
7	RW	0x0	wakeup_usb_en Enable USB detect interrupt as wakeup source. 1'b0: Disable 1'b1: Enable
6	RW	0x0	wakeup_sdio_en Enable SDIO interrupt as wakeup source. 1'b0: Disable 1'b1: Enable
5	RW	0x0	wakeup_sdmmc_en Enable SDMMC detect interrupt as wakeup source. 1'b0: Disable 1'b1: Enable
4	RW	0x0	wakeup_gpio0_int_en Enable GPIO0 interrupt as wakeup source. 1'b0: Disable 1'b1: Enable
3	RW	0x0	wakeup_cpu3_int_en Enable CPU3 interrupt as wakeup source. 1'b0: Disable 1'b1: Enable
2	RW	0x0	wakeup_cpu2_int_en Enable CPU2 interrupt as wakeup source. 1'b0: Disable 1'b1: Enable

Bit	Attr	Reset Value	Description
1	RW	0x0	wakeup_cpu1_int_en Enable CPU1 interrupt as wakeup source. 1'b0: Disable 1'b1: Enable
0	RW	0x0	wakeup_cpu0_int_en Enable CPU0 interrupt as wakeup source. 1'b0: Disable 1'b1: Enable

PMU_WAKEUP_INT_ST

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:11	RO	0x000000	reserved
10	RO	0x0	wakeup_timerout_int_st PMU timeout interrupt as wakeup source status. 1'b0: Inactive 1'b1: Active
9	RO	0x0	wakeup_sys_int_st System interrupt as wakeup source status. 1'b0: Inactive 1'b1: Active
8	RO	0x0	wakeup_uart1_int_st UART1 interrupt as wakeup source status. 1'b0: Inactive 1'b1: Active
7	RO	0x0	wakeup_usb_int_st USB detect interrupt as wakeup source status. 1'b0: Inactive 1'b1: Active
6	RO	0x0	wakeup_sdio_int_st SDIO interrupt as wakeup source status. 1'b0: Inactive 1'b1: Active
5	RO	0x0	wakeup_sdmmc_int_st SDMMC detect interrupt as wakeup source status. 1'b0: Inactive 1'b1: Active
4	RO	0x0	wakeup_gpio0_int_st GPIO0 interrupt as wakeup source status. 1'b0: Inactive 1'b1: Active
3	RO	0x0	wakeup_cpu3_int_st CPU3 interrupt as wakeup source status. 1'b0: Inactive 1'b1: Active
2	RO	0x0	wakeup_cpu2_int_st CPU2 interrupt as wakeup source status. 1'b0: Inactive 1'b1: Active
1	RO	0x0	wakeup_cpu1_int_st CPU1 interrupt wakeup status. 1'b0: Inactive 1'b1: Active

Bit	Attr	Reset Value	Description
0	RO	0x0	wakeup_cpu0_int_st CPU0 interrupt as wakeup source status. 1'b0: Inactive 1'b1: Active

PMU_WAKEUP_EDGE_CON

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:20	RO	0x000	reserved
19:0	RW	0x00000	edge_wakeup_en Enable both posedge and negedge of GPIO0 pins as wakeup source. Each pin has independent control bit. 1'b0: Disable 1'b1: Enable

PMU_WAKEUP_EDGE_ST

Address: Operational Base + offset (0x001C)

Bit	Attr	Reset Value	Description
31:20	RO	0x000	reserved
19:0	W1C	0x00000	edge_status Edge interrupt wakeup status. Each pin has independent control bit. 1'b0: Inactive 1'b1: Active

PMU_SCU_STABLE_CNT

Address: Operational Base + offset (0x0020)

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

PMU_PMIC_STABLE_CNT

Address: Operational Base + offset (0x0024)

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

PMU_OSC_STABLE_CNT

Address: Operational Base + offset (0x0028)

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

PMU_WAKEUP_RSTCLR_CNT

Address: Operational Base + offset (0x002C)

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

PMU_PLL_LOCK_CNT

Address: Operational Base + offset (0x0030)

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

PMU_SCU_PWRUP_CNT

Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:20	RO	0x000	reserved
19:0	RW	0x05dc0	stable_cnt SCU power up stable counter, reflect on the falling time of pd_scu_dwn_ack.

PMU_SCU_PWRDN_CNT

Address: Operational Base + offset (0x003C)

Bit	Attr	Reset Value	Description
31:20	RO	0x000	reserved
19:0	RW	0x05dc0	stable_cnt SCU power down stable counter, reflect on the rising time of pd_scu_dwn_ack.

PMU_SCU_VOLUP_CNT

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:20	RO	0x000	reserved
19:0	RW	0x05dc0	stable_cnt SCU voltage up stable counter, reflect on the falling time of pd_scu_dwn_en.

PMU_SCU_VOLDN_CNT

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:20	RO	0x000	reserved
19:0	RW	0x05dc0	stable_cnt SCU voltage up stable counter, reflect on the rising time of pd_scu_dwn_en.

PMU_WAKEUP_TIMEOUT_CNT

Address: Operational Base + offset (0x0048)

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

PMU_PWM_SWITCH_CNT

Address: Operational Base + offset (0x004C)

Bit	Attr	Reset Value	Description
31:0	RW	0x000fffff	stable_cnt PWM switch stable counter

PMU_SCU_PWR_CON

Address: Operational Base + offset (0x0080)

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
15:8	RO	0x00	reserved
7	RW	0x0	cluster_clk_src_gate_ena Cluster clock gate enable. 1'b0: Disable 1'b1: Enable
6	RW	0x0	scu_vol_gate_ena SCU voltage gate enable. 1'b0: Disable 1'b1: Enable
5	RW	0x0	cluster_cpu_pwrdn_ena CPU hardware power down enable. 1'b0: Disable 1'b1: Enable
4	RO	0x0	reserved
3	RW	0x0	scu_pwroff_ena SCU hardware power off enable. 1'b0: Disable 1'b1: Enable
2	RW	0x0	scu_pwrdn_ena SCU hardware power down enable. 1'b0: Disable 1'b1: Enable
1:0	RO	0x0	reserved

PMU_SCU_PWR_SFTCON

Address: Operational Base + offset (0x0084)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:4	RO	0x000	reserved
3	RW	0x0	scu_pwroff_ena SCU software power off enable. 1'b0: Disable 1'b1: Enable
2	RW	0x0	scu_pwrdn_ena SCU software power down enable. 1'b0: Disable 1'b1: Enable
1:0	RO	0x0	reserved

PMU_SCU_PWR_STATE

Address: Operational Base + offset (0x008C)

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

Bit	Attr	Reset Value	Description
3:0	RO	0x0	scu_power_state SCU power state. 4'h0: SCU normal state 4'h1: CPU power down state 4'h4: Cluster transfer idle state 4'h5: SCU power down state 4'h6: SCU sleep state 4'h7: SCU wakeup state 4'h8: SCU power up state 4'h9: Cluster transfer resume state 4'ha: CPU power up state Others: Reserved

PMU_CPU_PWR_SFTCON

Address: Operational Base + offset (0x0094)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:4	RO	0x000	reserved
3	RW	0x0	cpu3_sft_pwrdn_ena CPU3 software power down enable. 1'b0: Disable 1'b1: Enable
2	RW	0x0	cpu2_sft_pwrdn_ena CPU2 software power down enable. 1'b0: Disable 1'b1: Enable
1	RW	0x0	cpu1_sft_pwrdn_ena CPU1 software power down enable. 1'b0: Disable 1'b1: Enable
0	RW	0x0	cpu0_sft_pwrdn_ena CPU0 software power down enable. 1'b0: Disable 1'b1: Enable

PMU_CLUSTER_PWR_ST

Address: Operational Base + offset (0x009C)

Bit	Attr	Reset Value	Description
31:12	RO	0x00000	reserved
11	RO	0x0	cpu3_standbywfi CPU3 STANDBYWFI state. 1'b0: Inactive 1'b1: Active
10	RO	0x0	cpu2_standbywfi CPU2 STANDBYWFI state. 1'b0: Inactive 1'b1: Active
9	RO	0x0	cpu1_standbywfi CPU1 STANDBYWFI state. 1'b0: Inactive 1'b1: Active

Bit	Attr	Reset Value	Description
8	RO	0x0	cpu0_standbywfi CPU0 STANDBYWFI state. 1'b0: Inactive 1'b1: Active
7	RO	0x0	cpu3_dwn_state CPU3 power down state. 1'b0: Inactive 1'b1: Active
6	RO	0x0	cpu2_dwn_state CPU2 power down state. 1'b0: Inactive 1'b1: Active
5	RO	0x0	cpu1_dwn_state CPU1 power down state. 1'b0: Inactive 1'b1: Active
4	RO	0x0	cpu0_dwn_state CPU0 power down state. 1'b0: Inactive 1'b1: Active
3	RO	0x0	reserved
2	RO	0x0	scu_dwn_state SCU power down state. 1'b0: Inactive 1'b1: Active
1	RO	0x0	standbywfil2 L2 STANDBYWFI state. 1'b0: Inactive 1'b1: Active
0	RO	0x0	reserved

PMU_CLUSTER_IDLE_CON

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
15:5	RO	0x000	reserved
4	RW	0x0	idle_req_cpu Enable sending bus idle request to BIU_CPU by PMU. 1'b0: Disable 1'b1: Enable
3:0	RO	0x0	reserved

PMU_CLUSTER_IDLE_SFTCON

Address: Operational Base + offset (0x00A4)

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	0x000	reserved

Bit	Attr	Reset Value	Description
4	RW	0x0	idle_req_cpu Enable sending bus idle request to BIU_CPU by software. 1'b0: Disable 1'b1: Enable
3:0	RO	0x0	reserved

PMU_BUS_IDLE_CON0

Address: Operational Base + offset (0x00B0)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	idle_req_usb Enable sending bus idle request to BIU_USB by hardware. 1'b0: Disable 1'b1: Enable
14	RW	0x0	idle_req_gmac Enable sending bus idle request to BIU_GMAC by hardware. 1'b0: Disable 1'b1: Enable
13	RW	0x0	idle_req_sdio Enable sending bus idle request to BIU_SDIO by hardware. 1'b0: Disable 1'b1: Enable
12	RW	0x0	idle_req_sdcard Enable sending bus idle request to BIU_SDCARD by hardware. 1'b0: Disable 1'b1: Enable
11	RW	0x0	idle_req_nvm Enable sending bus idle request to BIU_NVM by hardware. 1'b0: Disable 1'b1: Enable
10	RW	0x0	idle_req_vdpu Enable sending bus idle request to BIU_VDPU by hardware. 1'b0: Disable 1'b1: Enable
9	RW	0x0	idle_req_vepu Enable sending bus idle request to BIU_VEPU by hardware. 1'b0: Disable 1'b1: Enable
8	RW	0x0	idle_req_ispp Enable sending bus idle request to BIU_ISPP by hardware. 1'b0: Disable 1'b1: Enable
7	RW	0x0	idle_req_vo Enable sending bus idle request to BIU_VO by hardware. 1'b0: Disable 1'b1: Enable
6	RW	0x0	idle_req_vi Enable sending bus idle request to BIU_VI by hardware. 1'b0: Disable 1'b1: Enable

Bit	Attr	Reset Value	Description
5	RW	0x0	idle_req_audio Enable sending bus idle request to BIU_AUDIO by hardware. 1'b0: Disable 1'b1: Enable
4	RW	0x0	idle_req_busnoc3 Enable sending bus idle request to BIU_BUSNOC3 by hardware. 1'b0: Disable 1'b1: Enable
3	RW	0x0	idle_req_busnoc2 Enable sending bus idle request to BIU_BUSNOC2 by hardware. 1'b0: Disable 1'b1: Enable
2	RW	0x0	idle_req_busnoc1 Enable sending idle request to BIU_BUSNOC1 by hardware. 1'b0: Disable 1'b1: Enable
1	RW	0x0	idle_req_bushold Enable sending bus idle request to BIU_BUSHOLD by hardware. 1'b0: Disable 1'b1: Enable
0	RW	0x0	idle_req_ddr Enable sending bus idle request to BIU_DDR by hardware. 1'b0: Disable 1'b1: Enable

PMU_BUS_IDLE_CON1

Address: Operational Base + offset (0x00B4)

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	0x000	reserved
4	RW	0x0	idle_req_crypt Enable sending bus idle request to BIU_CRYPT by hardware. 1'b0: Disable 1'b1: Enable
3	RW	0x0	idle_req_topapb Enablesending bus idle request to BIU_TOPAPB by hardware. 1'b0: Disable 1'b1: Enable
2	RW	0x0	idle_req_npu Enable sending bus idle request to BIU_NPU by hardware. 1'b0: Disable 1'b1: Enable
1	RW	0x0	idle_req_pmu Enable sending bus idle request to BIU_PMU by hardware. 1'b0: Disable 1'b1: Enable
0	RW	0x0	idle_req_php Enable sending bus idle request to BIU_PHP by hardware. 1'b0: Disable 1'b1: Enable

PMU_BUS_IDLE_SFTCON0

Address: Operational Base + offset (0x00C0)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	idle_req_usb Enable sending bus idle request to BIU_USB by software. 1'b0: Disable 1'b1: Enable
14	RW	0x0	idle_req_gmac Enable sending bus idle request to BIU_GMAC by software. 1'b0: Disable 1'b1: Enable
13	RW	0x0	idle_req_sdio Enable sending bus idle request to BIU_SDIO by software. 1'b0: Disable 1'b1: Enable
12	RW	0x0	idle_req_sdcard Enable sending bus idle request to BIU_SDCARD by software. 1'b0: Disable 1'b1: Enable
11	RW	0x0	idle_req_nvm Enable sending bus idle request to BIU_NVM by software. 1'b0: Disable 1'b1: Enable
10	RW	0x0	idle_req_vdpu Enable sending bus idle request to BIU_VDPU by software. 1'b0: Disable 1'b1: Enable
9	RW	0x0	idle_req_vepu Enable sending bus idle request to BIU_VEPU by software. 1'b0: Disable 1'b1: Enable
8	RW	0x0	idle_req_ispp Enable sending bus idle request to BIU_ISPP by software. 1'b0: Disable 1'b1: Enable
7	RW	0x0	idle_req_vo Enable sending bus idle request to BIU_VO by software. 1'b0: Disable 1'b1: Enable
6	RW	0x0	idle_req_vi Enable sending bus idle request to BIU_VI by software. 1'b0: Disable 1'b1: Enable
5	RW	0x0	idle_req_audio Enable sending bus idle request to BIU_AUDIO by software. 1'b0: Disable 1'b1: Enable
4	RW	0x0	idle_req_busnoc3 Enable sending bus idle request to BIU_BUSNOC3 by software. 1'b0: Disable 1'b1: Enable

Bit	Attr	Reset Value	Description
3	RW	0x0	idle_req_busnoc2 Enable sending bus idle request to BIU_BUSNOC2 by software. 1'b0: Disable 1'b1: Enable
2	RW	0x0	idle_req_busnoc1 Enable sending idle request to BIU_BUSNOC1 by software. 1'b0: Disable 1'b1: Enable
1	RW	0x0	idle_req_bushold Enable sending bus idle request to BIU_BUSHOLD by software. 1'b0: Disable 1'b1: Enable
0	RW	0x0	idle_req_ddr Enable sending bus idle request to BIU_DDR by software. 1'b0: Disable 1'b1: Enable

PMU_BUS_IDLE_SFTCON1

Address: Operational Base + offset (0x00C4)

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	0x000	reserved
4	RW	0x0	idle_req_crypt Enable sending bus idle request to BIU_CRYPT by software. 1'b0: Disable 1'b1: Enable
3	RW	0x0	idle_req_topapb Enablesending bus idle request to BIU_TOPAPB by software. 1'b0: Disable 1'b1: Enable
2	RW	0x0	idle_req_npu Enable sending bus idle request to BIU_NPU by software. 1'b0: Disable 1'b1: Enable
1	RW	0x0	idle_req_pmu Enable sending bus idle request to BIU_PMU by software. 1'b0: Disable 1'b1: Enable
0	RW	0x0	idle_req_php Enable sending bus idle request to BIU_PHP by software. 1'b0: Disable 1'b1: Enable

PMU_BUS_IDLE_ACK

Address: Operational Base + offset (0x00D0)

Bit	Attr	Reset Value	Description
31:21	RO	0x000	reserved
20	RO	0x0	idle_ack_crypt BIU_CRYPT bus idle acknowledge state. 1'b0: Not acknowledge 1'b1: Acknowledge

Bit	Attr	Reset Value	Description
19	RO	0x0	idle_ack_topapb BIU_TOPAPB bus idle acknowledge state. 1'b0: Not acknowledge 1'b1: Acknowledge
18	RO	0x0	idle_ack_npu BIU_NPU bus idle acknowledge state. 1'b0: Not acknowledge 1'b1: Acknowledge
17	RO	0x0	idle_ack_pmu BIU_PMU bus idle acknowledge state. 1'b0: Not acknowledge 1'b1: Acknowledge
16	RO	0x0	idle_ack_php BIU_PHP bus idle acknowledge state. 1'b0: Not acknowledge 1'b1: Acknowledge
15	RO	0x0	idle_ack_usb BIU_USB bus idle acknowledge state. 1'b0: Not acknowledge 1'b1: Acknowledge
14	RO	0x0	idle_ack_gmac BIU_GMAC bus idle acknowledge state. 1'b0: Not acknowledge 1'b1: Acknowledge
13	RO	0x0	idle_ack_sdio BIU_SDIO bus idle acknowledge state. 1'b0: Not acknowledge 1'b1: Acknowledge
12	RO	0x0	idle_ack_sdcard BIU_SDCARD bus idle acknowledge state. 1'b0: Not acknowledge 1'b1: Acknowledge
11	RO	0x0	idle_ack_nvm BIU_NVM bus idle acknowledge state. 1'b0: Not acknowledge 1'b1: Acknowledge
10	RO	0x0	idle_ack_vdpu BIU_VDPU bus idle acknowledge state. 1'b0: Not acknowledge 1'b1: Acknowledge
9	RO	0x0	idle_ack_vepu BIU_VEPU bus idle acknowledge state. 1'b0: Not acknowledge 1'b1: Acknowledge
8	RO	0x0	idle_ack_ispp BIU_ISPP bus idle acknowledge state. 1'b0: Not acknowledge 1'b1: Acknowledge
7	RO	0x0	idle_ack_vo BIU_VO bus idle acknowledge state. 1'b0: Not acknowledge 1'b1: Acknowledge

Bit	Attr	Reset Value	Description
6	RO	0x0	idle_ack_vi BIU_VI bus idle acknowledge state. 1'b0: Not acknowledge 1'b1: Acknowledge
5	RO	0x0	idle_ack_audio BIU_AUDIO bus idle acknowledge state. 1'b0: Not acknowledge 1'b1: Acknowledge
4	RO	0x0	idle_ack_busnoc3 BIU_BUSNOC3 bus idle acknowledge state. 1'b0: Not acknowledge 1'b1: Acknowledge
3	RO	0x0	idle_ack_busnoc2 BIU_BUSNOC2 bus idle acknowledge state. 1'b0: Not acknowledge 1'b1: Acknowledge
2	RO	0x0	idle_ack_busnoc1 BIU_BUSNOC1 bus idle acknowledge state. 1'b0: Not acknowledge 1'b1: Acknowledge
1	RO	0x0	idle_ack_bushold BIU_BUSHOLD bus idle acknowledge state. 1'b0: Not acknowledge 1'b1: Acknowledge
0	RO	0x0	idle_ack_ddr BIU_DDR bus idle acknowledge state. 1'b0: Not acknowledge 1'b1: Acknowledge

PMU_BUS_IDLE_ST

Address: Operational Base + offset (0x00D8)

Bit	Attr	Reset Value	Description
31:21	RO	0x000	reserved
20	RO	0x0	idle_crypt BIU_CRYPT idle state. 1'b0: Not idle 1'b1: Idle
19	RO	0x0	idle_topapb BIU_TOPAPB idle state. 1'b0: Not idle 1'b1: Idle
18	RO	0x0	idle_npu BIU_NPU idle state. 1'b0: Not idle 1'b1: Idle
17	RO	0x0	idle_pmu BIU_PMU idle state. 1'b0: Not idle 1'b1: Idle
16	RO	0x0	idle_php BIU_PHP idle state. 1'b0: Not idle 1'b1: Idle

Bit	Attr	Reset Value	Description
15	RO	0x0	idle_usb BIU_USB idle state. 1'b0: Not idle 1'b1: Idle
14	RO	0x0	idle_gmac BIU_GMAC idle state. 1'b0: Not idle 1'b1: Idle
13	RO	0x0	idle_sdio BIU_SDIO idle state. 1'b0: Not idle 1'b1: Idle
12	RO	0x0	idle_sdcard BIU_SDCARD idle state. 1'b0: Not idle 1'b1: Idle
11	RO	0x0	idle_nvm BIU_NVM idle state. 1'b0: Not idle 1'b1: Idle
10	RO	0x0	idle_vdpu BIU_VDPU idle state. 1'b0: Not idle 1'b1: Idle
9	RO	0x0	idle_vepu BIU_VEPU idle state. 1'b0: Not idle 1'b1: Idle
8	RO	0x0	idle_ispp BIU_ISPP idle state. 1'b0: Not idle 1'b1: Idle
7	RO	0x0	idle_vo BIU_VO idle state. 1'b0: Not idle 1'b1: Idle
6	RO	0x0	idle_vi BIU_VI idle state. 1'b0: Not idle 1'b1: Idle
5	RO	0x0	idle_audio BIU_AUDIO idle state. 1'b0: Not idle 1'b1: Idle
4	RO	0x0	idle_busnoc3 BIU_BUSNOC3 idle state. 1'b0: Not idle 1'b1: Idle
3	RO	0x0	idle_busnoc2 BIU_BUSNOC2 idle state. 1'b0: Not idle 1'b1: Idle

Bit	Attr	Reset Value	Description
2	RO	0x0	idle_busnoc1 BIU_BUSNOC1 idle state. 1'b0: Not idle 1'b1: Idle
1	RO	0x0	idle_bushold BIU_BUSHOLD idle state. 1'b0: Not idle 1'b1: Idle
0	RO	0x0	idle_ddr BIU_DDR idle state. 1'b0: Not idle 1'b1: Idle

PMU_NOC_AUTO_CON0

Address: Operational Base + offset (0x00E0)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	auto_idle_usb When perform idle operation, BIU_USB corresponding clock can be opened or gated automatically. 1'b0: Disable 1'b1: Enable
14	RW	0x0	auto_idle_gmac When perform idle operation, BIU_GMAC corresponding clock can be opened or gated automatically. 1'b0: Disable 1'b1: Enable
13	RW	0x0	auto_idle_sdio When perform idle operation, BIU_SDIO corresponding clock can be opened or gated automatically. 1'b0: Disable 1'b1: Enable
12	RW	0x0	auto_idle_sdcard When perform idle operation, BIU_SDCARD corresponding clock can be opened or gated automatically. 1'b0: Disable 1'b1: Enable
11	RW	0x0	auto_idle_nvm When perform idle operation, BIU_NVM corresponding clock can be opened or gated automatically. 1'b0: Disable 1'b1: Enable
10	RW	0x0	auto_idle_vdpu When perform idle operation, BIU_VDPU corresponding clock can be opened or gated automatically. 1'b0: Disable 1'b1: Enable
9	RW	0x0	auto_idle_vepu When perform idle operation, BIU_VEPU corresponding clock can be opened or gated automatically. 1'b0: Disable 1'b1: Enable

Bit	Attr	Reset Value	Description
8	RW	0x0	auto_idle_ispp When perform idle operation, BIU_ISPP corresponding clock can be opened or gated automatically. 1'b0: Disable 1'b1: Enable
7	RW	0x0	auto_idle_vo When perform idle operation, BIU_VO corresponding clock can be opened or gated automatically. 1'b0: Disable 1'b1: Enable
6	RW	0x0	auto_idle_vi When perform idle operation, BIU_VI corresponding clock can be opened or gated automatically. 1'b0: Disable 1'b1: Enable
5	RW	0x0	auto_idle_audio When perform idle operation, BIU_AUDIO corresponding clock can be opened or gated automatically. 1'b0: Disable 1'b1: Enable
4:1	RW	0x0	auto_idle_bus When perform idle operation, BIU_BUS corresponding clock can be opened or gated automatically. 3'b111: Enable Others: Disable
0	RW	0x0	auto_idle_ddr When perform idle operation, BIU_DDR corresponding clock can be opened or gated automatically. 1'b0: Disable 1'b1: Enable

PMU_NOC_AUTO_CON1

Address: Operational Base + offset (0x00E4)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:6	RO	0x000	reserved
5	RW	0x0	auto_idle_cpu When perform idle operation, BIU_CPU corresponding clock can be opened or gated automatically. 1'b0: Disable 1'b1: Enable
4	RW	0x0	auto_idle_crypt When perform idle operation, BIU_CRYPT corresponding clock can be opened or gated automatically. 1'b0: Disable 1'b1: Enable
3	RW	0x0	auto_idle_topapb When perform idle operation, BIU_TOPAPB corresponding clock can be opened or gated automatically. 1'b0: Disable 1'b1: Enable

Bit	Attr	Reset Value	Description
2	RW	0x0	auto_idle_npu When perform idle operation, BIU_NPU corresponding clock can be opened or gated automatically. 1'b0: Disable 1'b1: Enable
1	RW	0x0	auto_idle_pmu When perform idle operation, BIU_PMU corresponding clock can be opened or gated automatically. 1'b0: Disable 1'b1: Enable
0	RW	0x0	auto_idle_php When perform idle operation, BIU_PHP corresponding clock can be opened or gated automatically. 1'b0: Disable 1'b1: Enable

PMU_DDR_PWR_CON

Address: Operational Base + offset (0x00F0)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:5	RO	0x000	reserved
4	RW	0x0	ddrphy_auto_gating_ena Enable DDR phy auto clock gating function performed by PMU, when DDR enter self-refresh state. 1'b0: Disable 1'b1: Enable
3	RO	0x0	reserved
2	RW	0x0	ddrio_ret_exit_ena Enable DDR IO retention de-asserted performed by PMU. 1'b0: Disable 1'b1: Enable
1	RW	0x0	ddrio_ret_ena Enable DDR IO retention function performed by PMU. 1'b0: Disable 1'b1: Enable
0	RW	0x0	ddr_sref_ena Enable DDR self-refresh by PMU. 1'b0: Disable 1'b1: Enable

PMU_DDR_PWR_SFTCON

Address: Operational Base + offset (0x00F4)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:4	RO	0x000	reserved
3	RW	0x0	ddrctl_active_wait DDR controller waits for c_active high after c_sysack high. 1'b0: Disable 1'b1: Enable

Bit	Attr	Reset Value	Description
2	RW	0x0	sw_ddrio_ret_exit DDR IO retention exit request by software. 1'b0: Disable 1'b1: Enable
1	RW	0x0	sw_ddrio_ret_req DDR IO retention enter request by software. 1'b0: Disable 1'b1: Enable
0	RW	0x0	sw_ddr_sref_req DDR self-refresh request by software. 1'b0: Disable 1'b1: Enable

PMU_DDR_PWR_STATE

Address: Operational Base + offset (0x00F8)

Bit	Attr	Reset Value	Description
31:3	RO	0x00000000	reserved
2:0	RO	0x0	ddr_power_state DDR power state. 3'h0: Normal state 3'h1: Self-refresh enter state 3'h2: IO retention state 3'h3: Sleep state 3'h4: IO retention exit state 3'h5: Self-refresh exit state Others: Reserved

PMU_DDR_PWR_ST

Address: Operational Base + offset (0x00FC)

Bit	Attr	Reset Value	Description
31:3	RO	0x00000000	reserved
2	RO	0x0	ddrio_ret DDR IO retention state. 1'b0: Inactive 1'b1: Active
1	RO	0x0	ddrctl_c_active ddrctl c_active state. 1'b0: Inactive 1'b1: Active
0	RO	0x0	ddrctl_c_sysack ddrctl c_sysack state. 1'b0: Inactive 1'b1: Active

PMU_PWR_GATE_CON

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
15:11	RO	0x00	reserved

Bit	Attr	Reset Value	Description
10	RW	0x0	pd_crypt_dwn_ena Enable power down PD_CRYPT by PMU automatically. 1'b0: Disable 1'b1: Enable
9	RW	0x0	pd_usb_dwn_ena Enable power down PD_USB by PMU automatically. 1'b0: Disable 1'b1: Enable
8	RW	0x0	pd_sdio_dwn_ena Enable power down PD_SDIO by PMU automatically. 1'b0: Disable 1'b1: Enable
7	RW	0x0	pd_nvm_dwn_ena Enable power down PD_NVM by PMU automatically. 1'b0: Disable 1'b1: Enable
6	RW	0x0	pd_ddr_dwn_ena Enable power down PD_DDR by PMU automatically. 1'b0: Disable 1'b1: Enable
5	RW	0x0	pd_vo_dwn_ena Enable power down PD_VO by PMU automatically. 1'b0: Disable 1'b1: Enable
4	RW	0x0	pd_vi_dwn_ena Enable power down PD_VI by PMU automatically. 1'b0: Disable 1'b1: Enable
3	RW	0x0	pd_vdpu_dwn_ena Enable power down PD_VDPU by PMU automatically. 1'b0: Disable 1'b1: Enable
2	RW	0x0	pd_vepu_dwn_ena Enable power down PD_VEPU by PMU automatically. 1'b0: Disable 1'b1: Enable
1	RW	0x0	pd_ispp_dwn_ena Enable power down PD_ISPP by PMU automatically. 1'b0: Disable 1'b1: Enable
0	RW	0x0	pd_npu_dwn_ena Enable power down PD_NPU by PMU automatically. 1'b0: Disable 1'b1: Enable

PMU_PWR_GATE_STATE

Address: Operational Base + offset (0x0104)

Bit	Attr	Reset Value	Description
31:3	RO	0x00000000	reserved

Bit	Attr	Reset Value	Description
2:0	RO	0x0	power_gate_state Power domain state machine status. 3'h0: Normal state 3'h1: Power down start 3'h2: Power down selected domain 3'h3: Wait state 3'h4: Power up start 3'h5: Power up running selected domain Others: Reserved

PMU_PWR_DWN_ST

Address: Operational Base + offset (0x0108)

Bit	Attr	Reset Value	Description
31:11	RO	0x000000	reserved
10	RO	0x0	pd_crypt_dwn_stat The power status of PD_CRYPT. 1'b0: Power up 1'b1: Power down
9	RO	0x0	pd_usb_dwn_stat The power status of PD_USB. 1'b0: Power up 1'b1: Power down
8	RO	0x0	pd_sdio_dwn_stat The power status of PD_SDIO. 1'b0: Power up 1'b1: Power down
7	RO	0x0	pd_nvm_dwn_stat The power status of PD_NVM. 1'b0: Power up 1'b1: Power down
6	RO	0x0	pd_ddr_dwn_stat The power status of PD_DDR. 1'b0: Power up 1'b1: Power down
5	RO	0x0	pd_vo_dwn_stat The power status of PD_VO. 1'b0: Power up 1'b1: Power down
4	RO	0x0	pd_vi_dwn_stat The power status of PD_VI. 1'b0: Power up 1'b1: Power down
3	RO	0x0	pd_vdpu_dwn_stat The power status of PD_VDPU. 1'b0: Power up 1'b1: Power down
2	RO	0x0	pd_vepu_dwn_stat The power status of PD_VEPU. 1'b0: Power up 1'b1: Power down
1	RO	0x0	pd_ispp_dwn_stat The power status of PD_ISPP. 1'b0: Power up 1'b1: Power down

Bit	Attr	Reset Value	Description
0	RO	0x0	pd_npu_dwn_stat The power status of PD_NPU. 1'b0: Power up 1'b1: Power down

PMU_PWR_GATE_SFTCON

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
15:11	RO	0x00	reserved
10	RW	0x0	pd_crypt_dwn_ena Enable power down PD_CRYPT by software. 1'b0: Disable 1'b1: Enable
9	RW	0x0	pd_usb_dwn_ena Enable power down PD_USB by software. 1'b0: Disable 1'b1: Enable
8	RW	0x0	pd_sdio_dwn_ena Enable power down PD_SDIO by software. 1'b0: Disable 1'b1: Enable
7	RW	0x0	pd_nvm_dwn_ena Enable power down PD_NVM by software. 1'b0: Disable 1'b1: Enable
6	RW	0x0	pd_ddr_dwn_ena Enable power down PD_DDR by software. 1'b0: Disable 1'b1: Enable
5	RW	0x0	pd_vo_dwn_ena Enable power down PD_VO by software. 1'b0: Disable 1'b1: Enable
4	RW	0x0	pd_vi_dwn_ena Enable power down PD_VI by software. 1'b0: Disable 1'b1: Enable
3	RW	0x0	pd_vdpu_dwn_ena Enable power down PD_VDPU by software. 1'b0: Disable 1'b1: Enable
2	RW	0x0	pd_vepu_dwn_ena Enable power down PD_VEPU by software. 1'b0: Disable 1'b1: Enable
1	RW	0x0	pd_ispp_dwn_ena Enable power down PD_ISPP by software. 1'b0: Disable 1'b1: Enable

Bit	Attr	Reset Value	Description
0	RW	0x0	pd_npu_dwn_ena Enable power down PD_NPU by software. 1'b0: Disable 1'b1: Enable

PMU_VOL_GATE_SFTCON

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
15:3	RO	0x0000	reserved
2	RW	0x0	vd_vepu_ena Enable PD_VEPU as a voltage domain. 1'b0: Disable 1'b1: Enable
1	RO	0x0	reserved
0	RW	0x0	vd_npu_ena Enable PD_NPU as a voltage domain. 1'b0: Disable 1'b1: Enable

PMU_CRU_PWR_CON

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
15:12	RO	0x0	reserved
11	RW	0x0	pd_audio_clk_src_gate_ena Gating BIU_AUDIO's bus clock source. 1'b0: Disable 1'b1: Enable
10	RW	0x0	pd_peri_clk_src_gate_ena Gating BIU_PERI's bus clock source. 1'b0: Disable 1'b1: Enable
9	RW	0x0	pd_bus_clk_src_gate_ena Gating BIU_BUS's bus clock source. 1'b0: Disable 1'b1: Enable
8	RW	0x0	pwm_switch_iout PWM output. 1'b0: Disable 1'b1: Enable
7	RW	0x0	pwm_gpio_ioe_ena PWM output enable. 1'b0: Disable 1'b1: Enable
6	RW	0x0	pwm_switch_ena PWM switch. 1'b0: Disable 1'b1: Enable

Bit	Attr	Reset Value	Description
5	RW	0x0	power_off_ena Chip power off enable by hardware. 1'b0: Disable 1'b1: Enable
4	RW	0x0	alive_osc_ena pclk_pmu switch oscillator enable. When alive_32k_ena is asserted, this bit is ignored. 1'b0: Disable 1'b1: Enable
3	RW	0x0	input_clamp_ena VD_PMU input clamp enable by hardware. 1'b0: Disable 1'b1: Enable
2	RW	0x0	wakeup_rst_ena Wakeup reset enable. If asserted, the whole chip except IPs supporting reset hold function will be reset. 1'b0: Disable 1'b1: Enable
1	RW	0x0	osc_dis_ena Disable oscillator by hardware. 1'b0: Enable 1'b1: Disable
0	RW	0x0	alive_32k_ena Enable pclk_pmu and clk_pmu switch to 32KHz clock by hardware. 1'b0: Disable 1'b1: Enable

PMU_CRU_PWR_SFTCON

Address: Operational Base + offset (0x0124)

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

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

PMU_CRU_PWR_STATE

Address: Operational Base + offset (0x0128)

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

PMU_PLLPD_CON

Address: Operational Base + offset (0x0130)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:5	RO	0x000	reserved
4	RW	0x0	gpll_pd_ena GPLL power down by PMU. 1'b0: Disable 1'b1: Enable
3	RW	0x0	hpll_pd_ena HPLL power down by PMU. 1'b0: Disable 1'b1: Enable
2	RW	0x0	cpll_pd_ena CPLL power down by PMU. 1'b0: Disable 1'b1: Enable
1	RW	0x0	dpll_pd_ena DPLL power down by PMU. 1'b0: Disable 1'b1: Enable

Bit	Attr	Reset Value	Description
0	RW	0x0	apll_pd_ena APLL power down by PMU. 1'b0: Disable 1'b1: Enable

PMU_PLLPD_SFTCON

Address: Operational Base + offset (0x0134)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:5	RO	0x000	reserved
4	RW	0x0	gpll_pd_ena GPLL power down by software. 1'b0: Disable 1'b1: Enable
3	RW	0x0	hpll_pd_ena HPLL power down by software. 1'b0: Disable 1'b1: Enable
2	RW	0x0	cpll_pd_ena CPLL power down by software. 1'b0: Disable 1'b1: Enable
1	RW	0x0	dpll_pd_ena DPLL power down by software. 1'b0: Disable 1'b1: Enable
0	RW	0x0	apll_pd_ena APLL power down by software. 1'b0: Disable 1'b1: Enable

PMU_INFO_TX_CON

Address: Operational Base + offset (0x0150)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RW	0x00	info_tx_intv_time The interval time from 1 byte sends over to a new byte sends start. The value is the cycle number counted in clk_pmu
7	RO	0x0	reserved
6:4	RW	0x0	info_tx_con Power state output selection. 3'h0: PMU_MAIN_PWR_STATE 3'h1: PMU_SCU_PWR_STATE 3'h4: PMU_DDR_PWR_STATE 3'h5: PMU_PWR_GATE_STATE 3'h6: PMU_CRU_PWR_STATE Others: Reserved
3:1	RO	0x0	reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	info_tx_en Debug information transmit enable. 1'b0: Disable 1'b1: Enable

PMU_SYS_REG0

Address: Operational Base + offset (0x01C0)

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

PMU_SYS_REG1

Address: Operational Base + offset (0x01C4)

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

PMU_SYS_REG2

Address: Operational Base + offset (0x01C8)

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

PMU_SYS_REG3

Address: Operational Base + offset (0x01CC)

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

PMU_SYS_REG4

Address: Operational Base + offset (0x01D0)

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

PMU_SYS_REG5

Address: Operational Base + offset (0x01D4)

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

PMU_SYS_REG6

Address: Operational Base + offset (0x01D8)

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

PMU_SYS_REG7

Address: Operational Base + offset (0x01DC)

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

9.5 Timing Diagram

9.5.1 Each Domain Power Switch Timing

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

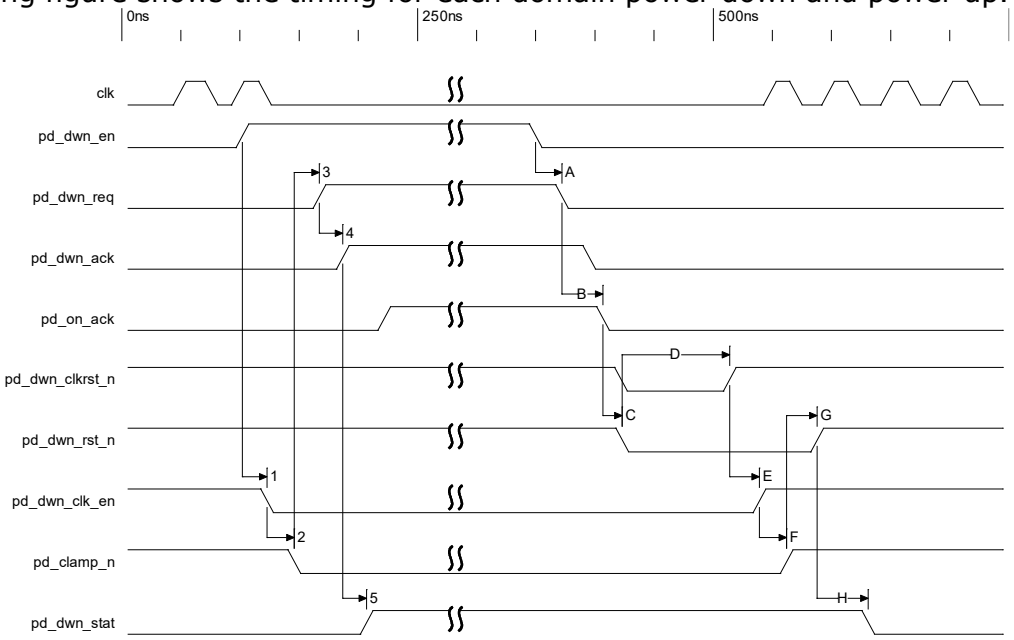


Fig. 9-3 Each Domain Power Switch Timing

9.5.2 External Wakeup PAD Timing

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

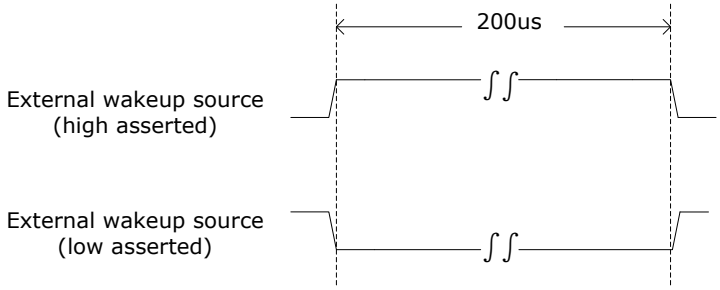


Fig. 9-4 External Wakeup Source PAD Timing

9.6 Application Notes

9.6.1 Low Power Mode

PMU can work in the Low power mode by setting bit[0] of PMU_PWR_CON register. After setting this bit and all CPUs enter standby states, PMU low power FSM will start to run. In the low power mode, PMU will manage power resources by hardware or software, 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.

9.6.2 Debug IO

ALL FSM power states could be monitored through IO. RV1109/RV1126 provide PMU Debug IO for FSM observation in UART signal mode.

Table 9-2 Debug IO for PMU FSM state

Module Pin	Direction	Pin Name	IOMUX Setting
pmu_debug	O	PMU_DEBUG/UART1_CTSN_M0/PWM3_IR_M0/GPIO0_C1_d	PMUGRF_GPIO0C_IOMUX[7:4]=4'h1

9.6.3 System Register

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

9.6.4 Configuration Constraint

In order to shut down the power domains which are managed by software correctly, the software must obey the rules bellow:

- Send BIU request to the BIU in power domain that you want to shut down by configure PMU_BUS_IDLE_SFTCON0 or PMU_BUS_IDLE_SFTCON1 register.
- Querying PMU_ BUS_IDLE_ST register to get the information until the pacific BIU is in idle state.
- Send power request to the power domain through PMU_PWR_GATE_CON register.
- Querying PMU_PWR_DWN_ST register to make sure the pacific power domain is power down.

Chapter 10 System Debug

10.1 Overview

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

10.1.1 Features

- Invasive debug with core halted
- SW-DP

10.1.2 Debug Components Address Map

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

Module	Base Address
DAP_ROM	0xff000000

10.2 Block Diagram

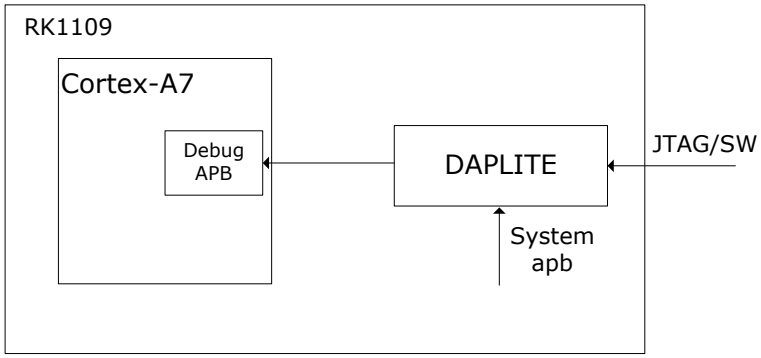


Fig. 10-1Debug system structure

10.3 Function Description

10.3.1 DAP

The DAP has following components:

- Serial Wire JTAG Debug Port(SWJ-DP)
- APB Access Port(APB-AP)
- ROM table

The debug port is the host tools interface to access the DAP-Lite. This interface controls any access ports provided within the DAP-Lite. The DAP-Lite supports a combined debug port which includes both JTAG and Serial Wire Debug(SWD), with a mechanism that supports switching between them.

The APB-AP acts as a bridge between SWJ-DP and APB bus which translate the Debug request to APB bus.

The DAP provides an internal ROM table connected to the master Debug APB port of the APB-Mux. The Debug ROM table is loaded at address 0x00000000 and 0x80000000 of this bus and is accessible from both APB-AP and the system APB input. Bit[31] of the address bus is not connected to the ROM Table, ensuring that both views read the same value. The ROM table stores the locations of the components on the Debug APB.

More information please refer to the document CoreSight_DAPLite_TRM.pdf for the debug detail description.

10.4 Register Description

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

10.5 Interface Description

10.5.1 DAP SWJ-DP Interface

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

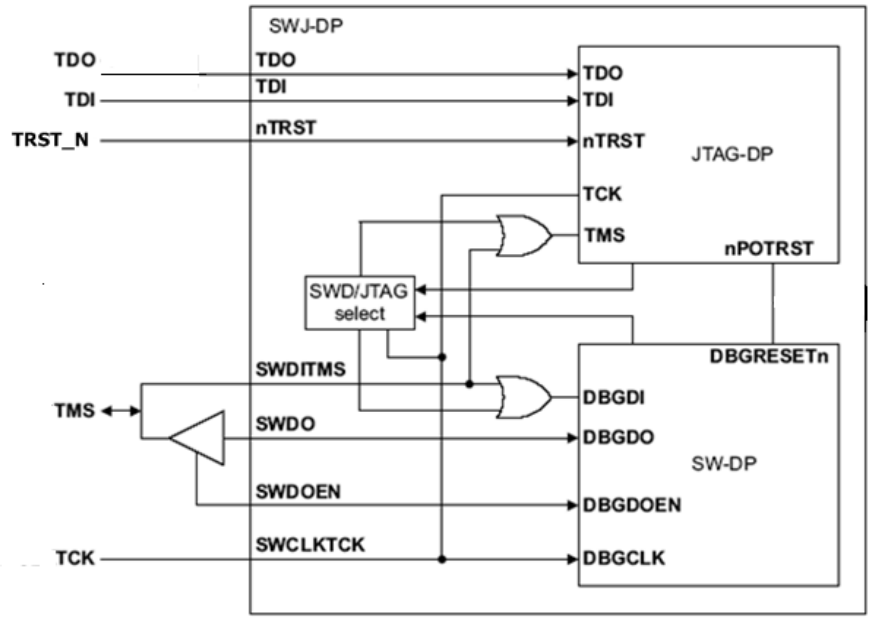


Fig. 10-2 DAP SWJ interface

10.5.2 DAP SW-DP Interface

This implementation is taken from ADIv5.1 and operates with a synchronous serial interface. This uses a single bidirectional data signal, and a clock signal.

The figure below describes the interaction between the timing of transactions on the serial wire interface, and the DAP internal bus transfers. It shows when the target responds with a WAIT acknowledgement.

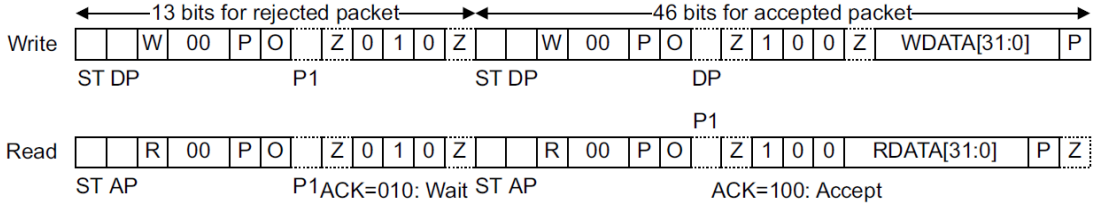


Fig. 10-3 SW-DP acknowledgement timing

Table 10-1 DAP-Lite Interface Description

Module pin	Direction	Pad name	IOMUX
jtag_tckm0	I	SDMMC0_D2/UART3_RX_M1/A7_JTAG_TCK_M0/RISC-V_JTAG_TCK/GPIO1_A6_u	GRF_GPIO1A_IOMUX_SEL_H[11:8]=4'b0011
jtag_tmsm0	I/O	SDMMC0_D3/UART3_TX_M1/A7_JTAG_TMS_M0/RISC-V_JTAG_TMS/GPIO1_A7_u	GRF_GPIO1A_IOMUX_SEL_H[11:8]=4'b0100
jtag_tckm1	I	UART2_TX_M1/A7_JTAG_TCK_M1/GPIO3_A2_u	GRF_GPIO3A_IOMUX_SEL_L[11:8]=4'b0010
jtag_tmsm1	I/O	UART2_RX_M1/A7_JTAG_TMS_M1/GPIO3_A3_u	GRF_GPIO3A_IOMUX_SEL_L[15:12]=4'b0010

By default, jtag_tckm0 and jtag_tmsm0 are forced to connect to corresponding io pad, since the bit[4] of grf_iofunc_sel:grf_force_jtag = 1'b1.

Chapter 11 SPINLOCK

11.1 Overview

The hardware spinlock used for spinlock status storage. All the CPU can access spinlock to get the lock status.

11.2 Block Diagram

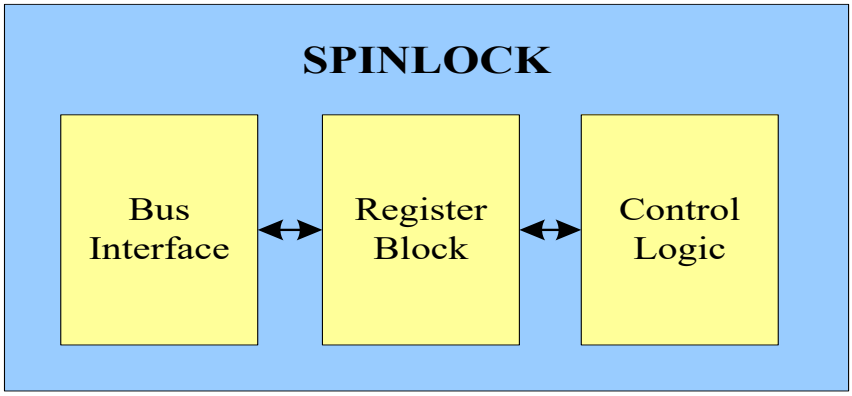


Fig. 11-1 Spinlock Block Diagram

- Bus Interface

The Bus Interface implements the bus slave operation.

- Register Block

The Register block includes the lock status registers.

- Control Logic

The control logic implemented the spinlock function.

11.3 Function Description

The SPINLOCK includes 64 lock-status register groups. Each lock-status register has 4bits, which corresponding to the lock status.
When 4bits spinlock_status is not zero, this lock-status register cannot be written.
When write data is 4'b0, 4bits spinlock_status registers clean to all zero.
If write data is 4'b0, that means unlock process.
If write data is 4bits ID, that means lock process. If read back value is equal to write data ID, that means lock successfully.

11.4 Register Description

11.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
SPINLOCK_status_n (range of n is 0~63)	0x0+4*n	W	0x00000000	spinlock status controller register_n

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

11.4.2 Detail Registers Description

SPINLOCK_status_nnpu_LINK_1_FROM

Address: Operational Base + offset (0x0000+4*n)

Bit	Attr	Reset Value	Description
31:4	RO	0x00	reserved

Bit	Attr	Reset Value	Description
3:0	RW	0x0	spinlock_status When 4bits is 0, 4bits can be written with new value. When 4bits is not 0, 4bits cannot be written. When write data is 0x0000, 4bits clean to 0.

11.5 Application Notes

Please refer to the "Function Description" section.

Chapter 12 Mailbox

12.1 Overview

The Mailbox module is a simple APB peripheral that allows Cortex-A7, RISC-V core to communicate with each other by writing operation to generate interrupt. The registers are accessible via APB interface.
The Mailbox has the following main features:

- Support APB interface
- Support four mailbox elements, each element includes one data word, one command word register and one flag bit that can represent one interrupt
- Support interrupts to Cortex-A7, and RISC-V core
- Provide 32 lock registers for software to use to indicate whether mailbox is occupied

12.2 Block Diagram

The figure below shows Mailbox block diagram:

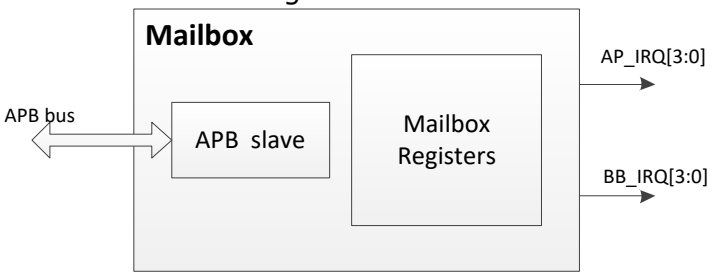


Fig. 12-1 Mailbox Block Diagram

12.3 Function Description

12.3.1 Mailbox

- Regard Cortex-A7 as the "AP" side of the Mailbox. The four elements combined interrupt (|mailbox_irq_ap[3:0]) to Cortex-A7 is:
  - Enabled when IRQs[111] is enabled in the GIC and MAILBOX_B2A_INTEN[i] is set to 1. (i=0~3)
  - Generated when there are writing operation to corresponding MAILBOX_B2A_CMD_i and MAILBOX_B2A_DAT_i orderly.
  - Cleared when writing 1 to corresponding MAILBOX_B2A_STATUS[i].
- Regard RISC-V core as the "BB" side of the Mailbox. The four elements combined interrupt (|mailbox_irq_bb[3:0]) to RISC-V core is:
  - Enabled when IRQs[112]/IRQs[13] is enabled in the INTC/IPIC of RISC-V core and MAILBOX_A2B_INTEN[i] is set to 1. (i=0~3)
  - Generated when there are writing operation to corresponding MAILBOX_A2B_CMD_i and MAILBOX_A2B_DAT_i orderly.
  - Cleared when writing 1 to corresponding MAILBOX_A2B_STATUS[i].
- You can also regard Cortex-A7 as the "BB" side of the Mailbox and regard RISC-V core as the "AP" side of the Mailbox. The configuration flow is similar.

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.

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

Name	Offset	Size	Reset Value	Description
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
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

Name	Offset	Size	Reset Value	Description
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
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:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access, **DW**-Double WORD (64 bits) access*

12.4.2 Detail Registers Description

MAILBOX_A2B_INTEN

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:4	RO	0x0000000	reserved
3	RW	0x0	int3 Interrupt enable for int3. 1'b0: Disable 1'b1: Enable
2	RW	0x0	int2 Interrupt enable for int2. 1'b0: Disable 1'b1: Enable
1	RW	0x0	int1 Interrupt enable for int1. 1'b0: Disable 1'b1: Enable
0	RW	0x0	int0 Interrupt enable for int0. 1'b0: Disable 1'b1: Enable

MAILBOX_A2B_STATUS

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:4	RO	0x0000000	reserved
3	W1C	0x0	int3 Interrupt status for int3. Clear the interrupt by writing 1 to this bit. 1'b0: Interrupt is inactive 1'b1: Interrupt is active
2	W1C	0x0	int2 Interrupt status for int2. Clear the interrupt by writing 1 to this bit. 1'b0: Interrupt is inactive 1'b1: Interrupt is active
1	W1C	0x0	int1 Interrupt status for int1. Clear the interrupt by writing 1 to this bit. 1'b0: Interrupt is inactive 1'b1: Interrupt is active
0	W1C	0x0	int0 Interrupt status for int0. Clear the interrupt by writing 1 to this bit. 1'b0: Interrupt is inactive 1'b1: Interrupt is active

MAILBOX_A2B_CMD_0

Address: Operational Base + offset (0x0008)

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

MAILBOX_A2B_DAT_0

Address: Operational Base + offset (0x000C)

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

MAILBOX_A2B_CMD_1

Address: Operational Base + offset (0x0010)

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

MAILBOX_A2B_DAT_1

Address: Operational Base + offset (0x0014)

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

MAILBOX_A2B_CMD_2

Address: Operational Base + offset (0x0018)

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

MAILBOX_A2B_DAT_2

Address: Operational Base + offset (0x001C)

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

MAILBOX_A2B_CMD_3

Address: Operational Base + offset (0x0020)

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

MAILBOX_A2B_DAT_3

Address: Operational Base + offset (0x0024)

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

MAILBOX_B2A_INTEN

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:4	RO	0x0000000	reserved
3	RW	0x0	int3 Interrupt enable for int3. 1'b0: Disable 1'b1: Enable
2	RW	0x0	int2 Interrupt enable for int2. 1'b0: Disable 1'b1: Enable
1	RW	0x0	int1 Interrupt enable for int1. 1'b0: Disable 1'b1: Enable
0	RW	0x0	int0 Interrupt enable for int0. 1'b0: Disable 1'b1: Enable

MAILBOX_B2A_STATUS

Address: Operational Base + offset (0x002C)

Bit	Attr	Reset Value	Description
31:4	RO	0x0000000	reserved
3	W1C	0x0	int3 Interrupt status for int3. Clear the interrupt by writing 1 to this bit. 1'b0: Interrupt is inactive 1'b1: Interrupt is active
2	W1C	0x0	int2 Interrupt status for int2. Clear the interrupt by writing 1 to this bit. 1'b0: Interrupt is inactive 1'b1: Interrupt is active

Bit	Attr	Reset Value	Description
1	W1C	0x0	int1 Interrupt status for int1. Clear the interrupt by writing 1 to this bit. 1'b0: Interrupt is inactive 1'b1: Interrupt is active
0	W1C	0x0	int0 Interrupt status for int0. Clear the interrupt by writing 1 to this bit. 1'b0: Interrupt is inactive 1'b1: Interrupt is active

MAILBOX_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	0x00000000	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	0x00000000	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	0x00000000	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	0x00000000	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	0x00000000	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	0x00000000	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	0x00000000	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	0x00000000	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	0x00000000	reserved
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	0x00000000	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	0x00000000	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	0x00000000	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	0x00000000	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	0x00000000	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	0x00000000	reserved
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	0x00000000	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	0x00000000	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	0x00000000	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	0x00000000	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	0x00000000	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	0x00000000	reserved
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	0x00000000	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	0x00000000	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	0x00000000	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	0x00000000	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	0x00000000	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	0x00000000	reserved
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	0x00000000	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	0x00000000	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	0x00000000	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	0x00000000	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	0x00000000	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 read value is 0 means it is available, and 1 means it has been automatically locked. Writing to the ATOMIC_LOCK register will clear this bit.
- Write to the CMD register before writing to the DAT register. If wrong order is used, then the interrupt cannot be generated successfully.
- If you want to clear the interrupt, you can read out the STATUS register and writing 1 to corresponding bit.

Chapter 13 DECOM

13.1 Overview

DECOM can decompress compressed files in GZIP, LZ4, Deflate and Zlib formats. DECOM controller supports the following features:

- Support for decompressing GZIP files
- Support for decompressing LZ4 files, including the General Structure of LZ4 Frame format and the Legacy Frame format
- Support for decompressing data in Deflate format
- Support for decompressing data in ZLIB format
- There is a 32bit APB slave interface for configuring decompression parameters and querying register status
- There is a 128-bit AXI Master interface for reading compressed data and outputting decompressed data. The AXI Master interface supports Burst 4/8/16 and Single transmissions
- Support one internal 128-bit wide and 64-location deep FIFOs(RX_FIFO) for caching source compressed data
- Support one internal 24-bit wide and 64-location deep FIFOs(HF_FIFO) for caching intermediate data during decompression
- Built-in a 32K-128bit two-port RAM for storing decompressed data
- Support complete interrupt and error interrupt output
- Support Hash32 check in LZ4 decompression process
- Support the limit_size function of the decompressed data to prevent the memory from being maliciously destroyed during the decompression process
- Support software to stop the decompression process

13.2 Block Diagram

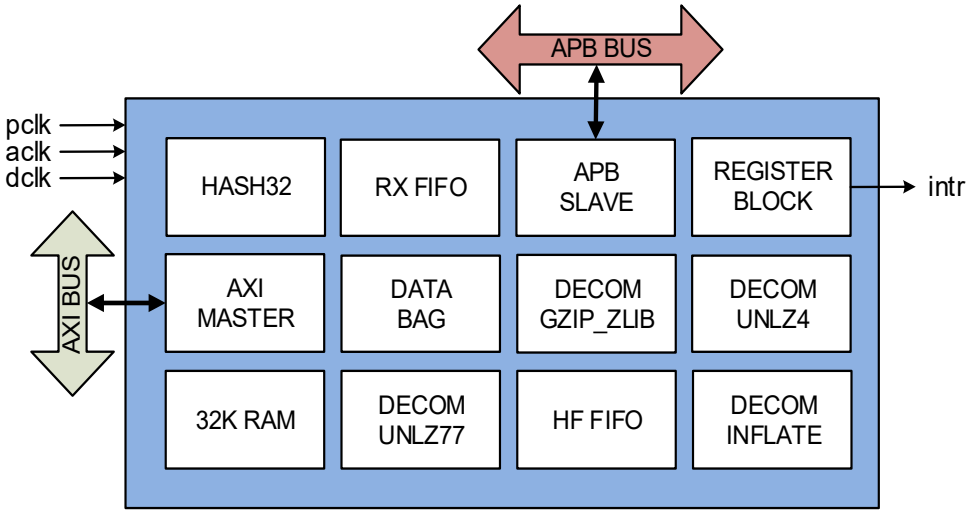


Fig. 13-1 DECOM Block Diagram

13.3 Function Description

13.3.1 Module Introduction

13.3.1.1 APB INTERFACE

The APB slave interface is used for registers configuration, decompression file parameter reading and interrupt status query, etc.

13.3.1.2 AXI MASTER INTERFACE

The AXI Master interface is used to read compressed file data from external storage into the

RX_FIFO and output the decompressed data to external storage. When decompressing the LZ4 file, it is also used to read back data with a matching distance of more than 32 KB from the external storage.

13.3.1.3 RX_FIFO

The RX_FIFO is used to buffer the source compressed file data read back through the AXI Master interface. The RX_FIFO is an asynchronous FIFO, the write interface uses aclk, and the read interface uses dclk.

13.3.1.4 DATA BAG

The Data Bag reads and bites the data in the RX_FIFO, and extracts the corresponding number of bits according to the needs of the decompression module.

13.3.1.5 UNLZ4 Module

The UNLZ4 module performs file header parsing on the compressed file of LZ4 format, and decompresses the data into LZ77 format (match length-matching distance) data pair or character data, and stores it in HF_FIFO.

13.3.1.6 GZIP_ZLIB

The GZIP_ZLIB module is used to perform file header and file end parsing for compressed files in GZIP format and compressed data in ZLIB format, and the compressed data portion is decompressed by the INFLATE module. The GZIP or ZLIB decompression mode can be configured by configuring the CTRL register.

13.3.1.7 INFLATE Module

INFLATE Module is used to decompress compressed files in deflate format, implement static/dynamic Huffman decoding, decompress compressed data in deflate format into LZ77 format (match length-match distance) data pairs or character data, and store them in HF_FIFO.

13.3.1.8 HF_FIFO

The HF_FIFO is used to buffer the LZ77 format (match length-match distance) data pair or character data decompressed by the UNLZ4 module and the INFLATE module, and wait for the UNLZ77 module to further decompress the intermediate data. The HF_FIFO is an asynchronous FIFO, the write interface uses dclk, and the read interface uses aclk.

13.3.1.9 UNLZ77 Module

UNLZ77 is used to implement LZ77 decoding and store the decompressed data in 32K RAM. When the matching distance is less than 32k byte, matching replication is performed in 32K RAM. When the matching distance is greater than 32k byte, the AXI Master interface is controlled to perform matching replication in the external storage.

13.3.1.10 32K-RAM

The 32K RAM is used to buffer the decompressed data. During the decompression process, the UNLZ77 module also performs read access to the RAM to complete the data matching.

13.3.1.11 HASH32 Module

In the process of decompressing the LZ4 compressed file, the HASH32 module is used to calculate the hash32 check value of the file header, data block and decompressed data.

13.3.2 Interface and Clock

The 32-bit APB Slave interface in DECOM is used for register configuration and interrupt status query interface, using pclk as the interface clock.

DECOM uses 128bit AXI Master interface to realize the functions of reading the data to be compressed, decompressing the data output and decompressing LZ4 external matching data; it should be noted that the AXI Master interface supports Single and Burst transmission, which only supports incr4, incr8 and Incr16 three transmission types, does not support Wrap transmission. DECOM automatically configures the burst type according to the length of the transmitted data and the starting address. The software configuration is not supported to select the burst type.

DECOM has three sets of clock and reset inputs, pclk and presetn, dclk and drstn, aclk and aresetn. Among them pclk and aclk are APB and AXI bus clocks respectively, and dclk is decompression clock.

13.3.3 Data Flow

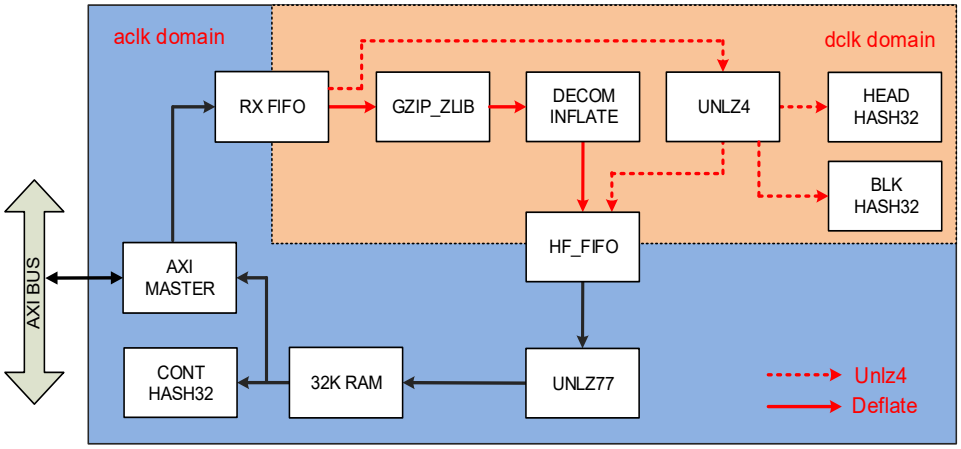


Fig. 13-2 DECOM Data Flow

13.4 Register Description

13.4.1 Internal Address Mapping

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

13.4.2 Registers Summary

Name	Offset	Size	Reset Value	Description
DECOM_CTRL	0x0000	W	0x00000000	Control Register.
DECOM_ENR	0x0004	W	0x00000000	Enable Register.
DECOM_RADDR	0x0008	W	0x00000000	Read Address.
DECOM_WADDR	0x000c	W	0x00000000	Write Address.
DECOM_UDDSL	0x0010	W	0xffffffff	Undecompressed Data Size
DECOM_UDDSH	0x0014	W	0xffffffff	Undecompressed Data Size
DECOM_TXTHR	0x0018	W	0x00000100	Transmit Threshold Level.
DECOM_SLEN	0x0020	W	0x00000000	Inflate Store Length.
DECOM_STAT	0x0024	W	0x00000000	DECOM Status Register.
DECOM_ISR	0x0028	W	0x00000000	Interrupt Status Register
DECOM_IEN	0x002c	W	0x00000000	Interrupt Enable Register
DECOM_AXI_STAT	0x0030	W	0x00000010	AXI Master Interface State.
DECOM_TSIZEL	0x0034	W	0x00000000	Decompressed Data Total Size Lower 32bit.
DECOM_TSIZEH	0x0038	W	0x00000000	Decompressed Data Total Size Upper 32bit.
DECOM_MGNUM	0x003c	W	0x00000000	LZ4 File Magic Number.
DECOM_FRAME	0x0040	W	0x00000000	LZ4 File Frame Descriptor.
DECOM_DICTID	0x0044	W	0x00000000	Dictionary ID.
DECOM_CSL	0x0048	W	0x00000000	LZ4 Content Size(CS) Lower 32-bits.
DECOM_CSH	0x004c	W	0x00000000	LZ4 Content Size(CS) Upper 32-bits.
DECOM_LMTSL	0x0050	W	0xffffffff	Limit Size of Decompressed Data.

Name	Offset	Size	Reset Value	Description
DECOM_LMTSH	0x0054	W	0xffffffff	Limit Size of Decompressed Data.
DECOM_GZFHD	0x0058	W	0x00000000	GZIP/ZLIB File Header Information.
DECOM_VERSION	0x00f0	W	0x00000926	DECOM Version Number

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

13.4.3 Detail Registers Description

DECOM_CTRL

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:6	RO	0x0000000	reserved
5	RW	0x0	ZLIBM ZLIB Mode 1'b0: Disable. 1'b1: Enable. When decompressing ZLIB compressed data, ZLIBM must be enabled and DEM should be set to 1.
4	RW	0x0	GZIPM GZIP Mode 1'b0: Disable. 1'b1: Enable. When decompressing GZIP files, GZIPM must be enabled and DEM should be set to 1.
3	RW	0x0	CCEN LZ4 Content Checksum Check Enable. In LZ4 files, a 32-bits content checksum will be appended at the end of the file. When CCEN is enabled, DECOM will verify the Content Checksum to check if the decompressed data is correct. Only valid when decompressing LZ4 files. 1'b0: Disable. 1'b1: Enable.
2	RW	0x0	BCEN LZ4 Block Checksum Check Enable. In LZ4 files, each data block will be followed by a 4-bytes checksum, calculated by using the xxHash-32 algorithm on the compressed data block. The intention is to detect data corruption (storage or transmission errors) immediately, before decoding. When BCEN is enabled, DECOM will check the Block Checksum to determine if the block to be decompressed is correct. Only valid when decompressing LZ4 files. 1'b0: Disable. 1'b1: Enable.

Bit	Attr	Reset Value	Description
1	RW	0x0	HCEN LZ4 Header Checksum Check Enable. 1'b0: Disable. 1'b1: Enable. The LZ4 header contains a 1-byte xxh32 Checksum value (HC). When HCEN is enabled, DECOM will check the HC to check if the header is correct. Only valid when decompressing LZ4 files.
0	RW	0x0	DEM DECOM Mode Select. 1'b0: Unlz4 mode. Decompress the file in LZ4 format. 1'b1: Inflate mode. Decompress the file in Inflate format. If GZIPM=1'b1or GZLIBM=1'b1, DEM must be set 1.

DECOM_ENR

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	WO	0x0	ENR Enables and disables all decompress operations. All FIFO buffers and aclk/dclk domain work registers are cleared when the device is disabled. Self Clear when decompression is complete. 1'b0: Disable decompressor. 1'b1: Enable decompressor to work. The ENR should be enabled after the other registers are configured.

DECOM_RADDR

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	RADDR Read Address. This register is used to configure the starting address of the file to be decompressed.

DECOM_WADDR

Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	WADDR Write Address. This register is used to configure the starting address of the location where the decompressed data is written. Note: WADDR[31:0] must be configured with a 128bit aligned address (WADDR[3:0]==4'b0).

DECOM_UDDSL

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:0	RW	0xffffffff	UDDSL Undecompressed Data Size Lower 32bits. UDDS[63:0] is the total size of undecompressed file. DECOM will read the compressed file data according to the value of UDDS[63:0], but the configuration of this register is optional. If this register is not configured, the default value is 64'hffff_ffff_ffff_ffff, DECOM will always read the data until the current The decompression process ends. (Unit:byte)

DECOM_UDDSH

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:0	RW	0xffffffff	UDDSH Undecompressed Data Size Higher 32bits.

DECOM_TXTHR

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000100	TXTHR When the number of 32K_MEM entries is greater than or equal to this value, the DECOM will automatically transfer the data in the 32K_MEM to the external via the AXI Master interface. The value ranges is 0~32256.(Unit:byte)

DECOM_SLEN

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	SLEN Inflate Store Block Size. Indicate the block size when decompressing the store block of Infalte file. For debugging SLC_ERR. (Unit:byte)

DECOM_STAT

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved
1	RO	0x0	LAST Used to indicate whether the currently processed compressed data block is the last one. 1'b0: Not Last Block. 1'b1: Last Block.

Bit	Attr	Reset Value	Description
0	RO	0x0	COMPLETE Decom Complete Flag. Only decompress total complete with no error, this flag is set to 1. 1'b0: Not complete. 1'b1: Decompress is complete with no error.

DECOM_ISR

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:20	RO	0x000	reserved
19	WO	0x0	DSOLI Decompressed Data Size Over Limit_size Interrupt. 1'b0: interrupt is not active 1'b1: interrupt is active
18	WO	0x0	ZDICTEI ZLIB Dictionary Error Interrupt. 1'b0: Interrupt is not active 1'b1: Interrupt is active
17	WO	0x0	GCMEI Gzip/Zlib Compression Method Check Error Interrupt. 1'b0: interrupt is not active 1'b1: interrupt is active
16	WO	0x0	GIDEI Gzip ID Error Interrupt. 1'b0: interrupt is not active 1'b1: interrupt is active
15	WO	0x0	CCCEI Unlz4 Content Checksum Check Error Interrupt. 1'b0: ccc_err interrupt is not active 1'b1: ccc_err interrupt is active
14	WO	0x0	BCCEI Unlz4 Block Checksum Check Error Interrupt. 1'b0: bcc_err interrupt is not active 1'b1: bcc_err interrupt is active
13	WO	0x0	HCCEI Unlz4 Header Checksum Check Error Interrupt. 1'b0: hcc_err interrupt is not active 1'b1: hcc_err interrupt is active
12	WO	0x0	CSEI Unlz4 Content Size Error Interrupt. 1'b0: cs_err interrupt is not active 1'b1: cs_err interrupt is active

Bit	Attr	Reset Value	Description
11	WO	0x0	DICTEI Unlz4 Dictionary Error Interrupt. 1'b0: Interrupt is not active 1'b1: Interrupt is active
10	WO	0x0	VNEI Unlz4 Version Number Error Interrupt. 1'b0: vn_err interrupt is not active 1'b1: vn_err interrupt is active
9	WO	0x0	MNEI Unlz4 Magic Number Error Interrupt 1'b0: mn_err interrupt is not active 1'b1: mn_err interrupt is active
8	WO	0x0	RDCEI AXI Read Channel Error Interrupt. Write 1 then Clear this interrupt. 1'b0: Inflate AXI RDC_ERR interrupt is not active. 1'b1: Inflate AXI RDC_ERR interrupt is active.
7	WO	0x0	WRCEI AXI Write Channel Error Interrupt. Write 1 then Clear this interrupt. 1'b0: Inflate AXI WRC_ERR interrupt is not active. 1'b1: Inflate AXI WRC_ERR interrupt is active.
6	WO	0x0	DISEI Inflate Huffman Distance Error Interrupt. Write 1 then Clear this interrupt. 1'b0: Inflate HFDIS_ERR interrupt is not active. 1'b1: Inflate HFDIS_ERR interrupt is active.
5	WO	0x0	LENEI Inflate Huffman Length Error Interrupt. Write 1 then Clear this interrupt. 1'b0: Inflate HFLEN_ERR interrupt is not active. 1'b1: Inflate HFLEN_ERR interrupt is active.
4	WO	0x0	LITEI Inflate Huffman Literal Error Interrupt. Write 1 then Clear this interrupt. 1'b0: Inflate HFLIT_ERR interrupt is not active. 1'b1: Inflate HFLIT_ERR interrupt is active.
3	WO	0x0	SQMEI Inflate SQ Match Error Interrupt. Write 1 then Clear this interrupt. 1'b0: Inflate SQM_ERR interrupt is not active. 1'b1: Inflate SQM_ERR interrupt is active.

Bit	Attr	Reset Value	Description
2	WO	0x0	SLCEI Inflate Store Block Len Check Error Interrupt. Write 1 then Clear this interrupt. 1'b0: Inflate SLC_ERR interrupt is not active. 1'b1: Inflate SLC_ERR interrupt is active.
1	WO	0x0	HDEI Inflate File Header Error Interrupt. Write 1 then Clear this interrupt. 1'b0: Inflate header error interrupt is not active. 1'b1: Inflate header error interrupt is active.
0	WO	0x0	DSI Decom Stop Interrupt. This interrupt indicates that DECOM has stopped working. Including decompression completion or decompression encountered an error. Write 1 then Clear this interrupt. 1'b0: decompression stop interrupt is not active. 1'b1: decompression stop interrupt is active.

DECOM_IEN

Address: Operational Base + offset (0x002c)

Bit	Attr	Reset Value	Description
31:20	RO	0x000	reserved
19	RW	0x0	DSOLIEN Decompressed Data Size Over Limit_size Interrupt Enable. 1'b0: Disable. 1'b1: Enable.
18	RW	0x0	ZDICTEIEN ZLIB Dictionary Error Interrupt Enable. 1'b0: Disable. 1'b1: Enable.
17	RW	0x0	GCMEIEN GZIP/ZLIB Compression Method Error Interrupt Enable. 1'b0: Disable. 1'b1: Enable.
16	RW	0x0	GIDEIEN GZIP ID Error Interrupt Enable. 1'b0: Disable. 1'b1: Enable.
15	RW	0x0	CCCEIEN Unlz4 Content Checksum Check Error Interrupt Enable. 1'b0: Disable. 1'b1: Enable.

Bit	Attr	Reset Value	Description
14	RW	0x0	BCCEIEN Unlz4 Block Checksum Check Error Interrupt Enable. 1'b0: Disable. 1'b1: Enable.
13	RW	0x0	HCCEIEN Unlz4 Header Checksum Check Error Interrupt Enable. 1'b0: Disable. 1'b1: Enable.
12	RW	0x0	CSEIEN Unlz4 Content Size Error Interrupt Enable. 1'b0: Disable. 1'b1: Enable.
11	RW	0x0	DICTEIEN Unlz4 Dictionary Error Interrupt Enable. 1'b0: Disable. 1'b1: Enable.
10	RW	0x0	VNEIEN Unlz4 Version Number Error Interrupt Enable. 1'b0: Disable. 1'b1: Enable.
9	RW	0x0	WNEIEN Unlz4 Magic Number Error Interrupt Enable. 1'b0: Disable. 1'b1: Enable.
8	RW	0x0	RDCEIEN AXI Read Channel Error Interrupt Enable. 1'b0: Disable. 1'b1: Enable.
7	RW	0x0	WRCEIEN AXI Write Channel Error Interrupt Enable. 1'b0: Disable. 1'b1: Enable.
6	RW	0x0	DISEIEN Inflate Huffman Distance Error Interrupt Enable. 1'b0: Disable. 1'b1: Enable.
5	RW	0x0	LENEIEN Inflate Huffman Length Error Interrupt Enable. 1'b0: Disable. 1'b1: Enable.
4	RW	0x0	LITEIEN Inflate Huffman Literal Error Interrupt Enable. 1'b0: Disable. 1'b1: Enable.

Bit	Attr	Reset Value	Description
3	RW	0x0	SQMEIEN Inflate SQ Match Error Interrupt Enable. 1'b0: Disable. 1'b1: Enable.
2	RW	0x0	SLCIEN Inflate Storte Block Len Check Error Interrupt Enable. 1'b0: Disable. 1'b1: Enable.
1	RW	0x0	HDEIEN Inflate File Header Error Interrupt Enable. 1'b0: Disable. 1'b1: Enable.
0	RW	0x0	DSIEN Decom Stop Interrupt Enable. 1'b0: Disable. 1'b1: Enable.

DECOM_AXI_STAT

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:5	RO	0x0000000	reserved
4	RO	0x1	AXI_IDLE AXI Master Idle State register. 1'b0: AXI Master is busy. 1'b1: AXI Master is idle. When the decompression is aborted, DECOM will wait for AXI Master's current read/write transfer to complete, that is, wait for AXI_IDLE=1, then reset DECOM, otherwise the uncompleted AXI transmission will cause AXI Bus exception. It should be noted that after the decompression is abnormal, the AXI Master's unfinished write operation will continue, but the output value of wstrb[15:0] will become 16'b0.
3:2	RO	0x0	RRESP AXI Read Channel Response State.
1:0	RO	0x0	BRESP AXI Write Channel Response State.

DECOM_TSIZEL

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	TSIZEL The Total Size of the data after decompression. TSIZE[63:0]. (Unit:byte)

DECOM_TSIZEH

Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	TSIZEH The Total Size of the data after decompression.

DECOM_MGNUM

Address: Operational Base + offset (0x003c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	MGNUM Magic Number is 0x184D2204. If the MN is not 0x184D2204, a MN Error will be generated. And DECOM will stop decompressing. For debug.

DECOM_FRAME

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:24	RO	0x00	reserved
23:16	RO	0x00	HCC Header Checksum Byte. One-byte checksum of combined descriptor fields, including optional ones. The value is the second byte of xxh32() using zero as a see, and the full Frame Descriptor as an input (including optional fields when they are present). A wrong checksum indicates an error in the descriptor. Header checksum is informational and can be skipped. The Header Checksum check function can be enabled by controlling the HCEN enable signal in the CTRL register. After being turned on, DECOM calculates the hash32 check value based on the data of the Frame Descriptor part of the LZ4 compressed file, and performs the Header Checksum Byte in the file. In contrast, if they are not equal, an HCC Error will be generated and an interrupt will be generated.
15:8	RO	0x00	BD BD Byte. Including Block Maximum Size information. This information is useful to help the decoder allocate memory. Size here refers to the original(uncompressed) data size.
7:6	RO	0x0	VN_NUM Version Number. 2-bits field, must be set to 2'b01. Any other value cannot be decoded by this version of the specification. If the version number is error, Version Number Error(VNE) will be generated.

Bit	Attr	Reset Value	Description
5	RO	0x0	BCC_FLG Block Checksum Flag. If this flag is set, each data block will be followed by a 4-bytes checksum, calculated by using the xxHash-32 algorithm on the raw (compressed) data block. The intention is to detect data corruption (storage or transmission errors) immediately, before decoding. Block checksum usage is optional.
4	RO	0x0	BIND_FLG Block Independence Flag. If this flag is set to "1", blocks are independent. If this flag is set to "0", each block depends on previous ones (up to LZ4 window size, which is 64 KB). In such case, it's necessary to decode all blocks in sequence. Block dependency improves compression ratio, especially for small blocks. On the other hand, it makes random access or multi-threaded decoding impossible. For debugging.
3	RO	0x0	CS_FLG Content Size Flag. If this flag is set, the uncompressed size of data included within the frame will be present as an 8 bytes unsigned little endian value, after the flags. Content Size usage is optional.
2	RO	0x0	CCC_FLG Content Checksum Flag. If this flag is set, a 32-bits content checksum will be appended after the EndMark.
1	RO	0x0	reserved
0	RO	0x0	DICT_FLG Dictionary ID Flag. If this flag is set, a 4-bytes Dict-ID field will be present, after the descriptor flags and the Content Size. If this flag is set, Dictionary ID Error will be generated. DECOM will stop decompressing.

DECOM_DICTID

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	DICTID Dictionary ID in the compressed file header. Dictionary ID is only present if the DID_FLG is set. It works as a kind of "known prefix" which is used by both the compressor and the decompressor to "warm-up" reference tables.

DECOM_CSL

Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	CSL Content Size(CS) Lower 32-bits. CS[63:0] is the original(uncompressed) size. This information is optional in LZ4 file header, and only present if CS_FLG is set. (Unit:byte) CS[63:0]=CSH, CSL;

DECOM_CSH

Address: Operational Base + offset (0x004c)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RO	0x0	CSH Content Size(CS) Upper 32-bits.

DECOM_LMTSL

Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:0	RW	0xffffffff	LMTSL Limit Size of Decompressed Data Lower 32bits. When the amount of Decompressed data is greater than the LMTS, DSOLI interrupt is generated and the decompression process is stopped.

DECOM_LMTSH

Address: Operational Base + offset (0x0054)

Bit	Attr	Reset Value	Description
31:0	RW	0xffffffff	LMTSH Limit Size of Decompressed Data Higher 32bits.

DECOM_GZFHD

Address: Operational Base + offset (0x0058)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RO	0x0	GZFHD GZIP/ZLIB File Header Information. When GZIPM = 1, GZFHD is GZIP file header information. When ZLIBM = 1, GZFHD is ZLIB file header information.

DECOM_VERSION

Address: Operational Base + offset (0x00f0)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000926	VERSION Vesion Number = 32'h0000_0926.

13.5 Application Notes

- After configuring the decompression mode and other decompression parameters, configure the decompression enable register ENR
- When the decompression is not completed, setting ENR from 1 to 0 will force the decompression process to stop and DECOM will return to the IDLE state. Need to reconfigure and enable DECOM to start the next decompression
- When an error is encountered in the decompression, DECOM will immediately stop the decompression process, set the relevant status register, and generate the corresponding interrupt
- After the decompression process is stopped, ENR will automatically change to 0
- After the decompression process is stopped, DSI will be set to 1. If DSIEN is 1, an interrupt will be generated; if DSIEN is 0, no interrupt will be generated
- After the decompression process is stopped, COMPLETE (ie STAT[0]) is not set to 1 if there is an error in the decompression process; COMPLETE is set to 1 only when the decompression is complete and there are no errors

Chapter 14 Temperature-Sensor ADC (TS-ADC)

14.1 Overview

TS-ADC Controller module supports user-defined mode and automatic mode.User-defined mode refers, TSADC all the control signals entirely by software writing to register for direct control.Automatic mode refers to the module automatically poll TSADC output, and the results were checked. If you find that the temperature High in a period of time, an interrupt is generated to the processor down-measures taken; if the temperature over a period of time High, the resulting TSHUT gave CRU module, let it reset the entire chip, or via GPIO give PMIC.

TS-ADC Controller supports the following features:

- Support User-Defined Mode and Automatic Mode
- In User-Defined Mode, start-of-conversion can be controlled completely by software, and also can be generated by hardware
- In Automatic Mode, the temperature of alarm(high/low temperature) interrupt can be configurable
- In Automatic Mode, the temperature of system reset can be configurable
- Support to 2 channel TS-ADC(CPU and NPU), the temperature criteria of each channel can be configurable
- In Automatic Mode, the time interval of temperature detection can be configurable
- In Automatic Mode, when detecting a high temperature, the time interval of temperature detection can be configurable
- High temperature debounce can be configurable
- -40~125°C temperature range and 5°C temperature resolution
- 12-bit SARADC up to 732 S/s sampling rate

14.2 Block Diagram

TS-ADC controller comprises with:

- APB Interface
- TS-ADC control logic

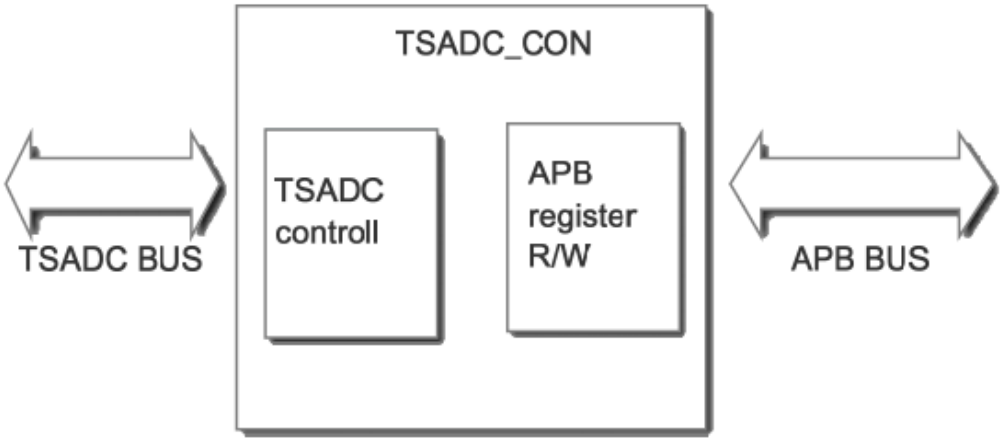


Fig. 14-1 TS-ADC Controller Block Diagram

14.3 Function Description

14.3.1 AHB Interface

There is an APB Slave interface in TS-ADC Controller, which is used to configure the TS-ADC Controller registers and look up the temperature from the temperature sensor.

14.3.2 TS-ADC Controller

This block is exploited to realize binary search algorithm, storing the intermediate result and

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

14.4 Register Description

14.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
TSADC_USER_CON	0x0000	W	0x00000200	User-Define Mode Control Register
TSADC_AUTO_CON	0x0004	W	0x00000000	Auto Mode Control Register
TSADC_INT_EN	0x0008	W	0x00000000	Interrupt Enable Register
TSADC_INT_PD	0x000C	W	0x00000000	Interrupt Status Register
TSADC_DATA0	0x0020	W	0x00000000	A/D Conversion Data Register for Channel 0
TSADC_DATA1	0x0024	W	0x00000000	A/D Conversion Data Register for Channel 1
TSADC_COMP0_INT	0x0030	W	0x00000000	High Temperature Level Register for Source 0
TSADC_COMP1_INT	0x0034	W	0x00000000	High Temperature Level Register for Source 1
TSADC_COMP0_SHUT	0x0040	W	0x00000000	Shut Temperature Level Register for Source 0
TSADC_COMP1_SHUT	0x0044	W	0x00000000	Shut Temperature Level Register for Source 1
TSADC_HIGHT_INT_DEBOUNCE	0x0060	W	0x00000003	High Temperature Debounce Register
TSADC_HIGHT_TSHUT_DEBOUNCE	0x0064	W	0x00000003	Shut Temperature Debounce Register
TSADC_AUTO_PERIOD	0x0068	W	0x00010000	Auto Access Period Register
TSADC_AUTO_PERIOD_HT	0x006C	W	0x00010000	High Temperature Auto Access Period Register
TSADC_COMP0_LOW_INT	0x0080	W	0x00000000	Low Temperature Level Register for Source 0
TSADC_COMP1_LOW_INT	0x0084	W	0x00000000	Low Temperature Level Register for Source 1

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

14.4.2 Detail Register Description

TSADC_USER_CON

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15	RW	0x0	adc_status 1'b0: ADC stop 1'b1: Conversion in progress
14:6	RW	0x008	inter_pd_soc Interleave between power down and start of conversion.
5	RW	0x0	start When software write 1 to this bit, start-of-conversion will be assert. This bit will be cleared after TSADC access finishing. Only when TSADC_USER_CON[4] = 1'b1, this bit takes effect.

Bit	Attr	Reset Value	Description
4	RW	0x0	start_mode Start mode. 1'b0: TSADC controller will assert start_of_conversion after "inter_pd_soc" cycles 1'b1: The start_of_conversion will be controlled by TSADC_USER_CON[5]
3	RW	0x0	adc_power_ctrl 1'b0: ADC power down 1'b1: ADC power up and reset
2:0	RW	0x0	adc_input_src_sel 3'b000: Input source 0 (SARADC_AIN[0]) 3'b001: Input source 1 (SARADC_AIN[1]) Others: Reserved

TSADC_AUTO_CON

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:26	RO	0x00	reserved
25	RW	0x0	last_tshut_2cru TSHUT status. This bit will set to 1 when tshut is valid, and only be cleared when application write 1 to it. This bit will not be cleared by system reset.
24	RW	0x0	last_tshut_2gpio TSHUT status. This bit will set to 1 when tshut is valid, and only be cleared when application write 1 to it. This bit will not be cleared by system reset.
23:18	RO	0x00	reserved
17	RW	0x0	sample_dly_sel 1'b0: AUTO_PERIOD is used 1'b1: AUTO_PERIOD_HT is used
16	RW	0x0	auto_status 1'b0: Auto mode stop 1'b1: Auto mode in progress
15:14	RO	0x0	reserved
13	RW	0x0	src1_lt_en 1'b0: Do not care low temperature of source 1 1'b1: Enable the low temperature monitor of source 1
12	RW	0x0	src0_lt_en 1'b0: Do not care low temperature of source 0 1'b1: Enable the low temperature monitor of source 0
11:9	RO	0x0	reserved
8	RW	0x0	tshut_polarity 1'b0: Low active 1'b1: High active
7:6	RO	0x0	reserved
5	RW	0x0	src1_en 1'b0: Do not care the temperature of source 1 1'b1: If the temperature of source 1 is too high, TSHUT will be valid
4	RW	0x0	src0_en 1'b0: Do not care the temperature of source 0 1'b1: If the temperature of source 0 is too high, TSHUT will be valid

Bit	Attr	Reset Value	Description
3:2	RO	0x0	reserved
1	RW	0x0	tsadc_q_sel 1'b0: Use tsadc_q as output (positive temperature coefficient) 1'b1: Use (4096-tsadc_q) as output (negative temperature coefficient)
0	RW	0x0	auto_en 1'b0: TSADC controller works at user-define mode 1'b1: TSADC controller works at auto mode

TSADC_INT_EN

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:17	RO	0x0000	reserved
16	RW	0x0	eoc_int_en End-of-conversion interrupt enable in user-define mode 1'b0: Disable 1'b1: Enable
15:14	RO	0x0	reserved
13	RW	0x0	lt_inten_src1 Low temperature interrupt enable for src1. 1'b0: Disable 1'b1: Enable
12	RW	0x0	lt_inten_src0 Low temperature interrupt enable for src0. 1'b0: Disable 1'b1: Enable
11:10	RO	0x0	reserved
9	RW	0x0	tshut_2cru_en_src1 1'b0: TSHUT output to CRU disabled. TSHUT output will always keep low 1'b1: TSHUT output works
8	RW	0x0	tshut_2cru_en_src0 1'b0: TSHUT output to CRU disabled. TSHUT output will always keep low 1'b1: TSHUT output works
7:6	RO	0x0	reserved
5	RW	0x0	tshut_2gpio_en_src1 1'b0: TSHUT output to GPIO disabled. TSHUT output will always keep low 1'b1: TSHUT output works
4	RW	0x0	tshut_2gpio_en_src0 1'b0: TSHUT output to GPIO disabled. TSHUT output will always keep low 1'b1: TSHUT output works
3:2	RO	0x0	reserved
1	RW	0x0	ht_inten_src1 High temperature interrupt enable for src1. 1'b0: Disable 1'b1: Enable
0	RW	0x0	ht_inten_src0 High temperature interrupt enable for src0. 1'b0: Disable 1'b1: Enable

TSADC_INT_PD

Address: Operational Base + offset (0x000C)

Bit	Attr	Reset Value	Description
31:17	RO	0x0000	reserved
16	RW	0x0	eoc_int_pd This bit will be set to 1 when end-of-conversion. Set 0 to clear the interrupt.
15:14	RO	0x0	reserved
13	RW	0x0	lt_irq_src1 When TSADC output is lower than COMP_INT_LOW, this bit will be valid, which means temperature is low, and the application should in charge of this. write 1 to it, this bit will be cleared.
12	RW	0x0	lt_irq_src0 When TSADC output is lower than COMP_INT_LOW, this bit will be valid, which means temperature is low, and the application should in charge of this. write 1 to it, this bit will be cleared.
11:6	RO	0x00	reserved
5	RW	0x0	tshut_o_src1 TSHUT output status. When TSADC output is bigger than COMP_SHUT, this bit will be valid, which means temperature is VERY high, and the application should in charge of this. write 1 to it, this bit will be cleared.
4	RW	0x0	tshut_o_src0 TSHUT output status. When TSADC output is bigger than COMP_SHUT, this bit will be valid, which means temperature is VERY high, and the application should in charge of this. write 1 to it, this bit will be cleared.
3:2	RO	0x0	reserved
1	RW	0x0	ht_irq_src1 When TSADC output is bigger than COMP_INT, this bit will be valid, which means temperature is high, and the application should in charge of this. write 1 to it, this bit will be cleared.
0	RW	0x0	ht_irq_src0 When TSADC output is bigger than COMP_INT, this bit will be valid, which means temperature is high, and the application should in charge of this. write 1 to it, this bit will be cleared.

TSADC_DATA0

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:12	RO	0x00000	reserved
11:0	RW	0x000	adc_data ADC data. A/D value of the channel 0 last conversion (DOUT[11:0]).

TSADC_DATA1

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:12	RO	0x00000	reserved

Bit	Attr	Reset Value	Description
11:0	RW	0x000	adc_data ADC data. A/D value of the channel 1 last conversion (DOUT[11:0]).

TSADC_COMP0_INT

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:12	RO	0x00000	reserved
11:0	RW	0x000	tsadc_comp_src0 TSADC high temperature level. TSADC output is bigger than tsadc_comp, means the temperature is high. TSADC_INT will be valid.

TSADC_COMP1_INT

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:12	RO	0x00000	reserved
11:0	RW	0x000	tsadc_comp_src1 TSADC high temperature level. TSADC output is bigger than tsadc_comp, means the temperature is high. TSADC_INT will be valid.

TSADC_COMP0_SHUT

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:12	RO	0x00000	reserved
11:0	RW	0x000	tsadc_comp_src0 TSADC high temperature level. TSADC output is bigger than tsadc_comp, means the temperature is too high. TSHUT will be valid.

TSADC_COMP1_SHUT

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:12	RO	0x00000	reserved
11:0	RW	0x000	tsadc_comp_src1 TSADC high temperature level. TSADC output is bigger than tsadc_comp, means the temperature is too high. TSHUT will be valid.

TSADC_HIGHT_INT_DEBOUNCE

Address: Operational Base + offset (0x0060)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x03	debounce TSADC controller will only generate interrupt or TSHUT when temperature is higher than COMP_INT for "debounce" times.

TSADC_HIGHT_TSHUT_DEBOUNCE

Address: Operational Base + offset (0x0064)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	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)

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)

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)

Bit	Attr	Reset Value	Description
31:12	RO	0x00000	reserved
11:0	RW	0x000	tsadc_comp_src0 TSADC low temperature level. TSADC output is lower than tsadc_comp, means the temperature is low. TSADC_LOW_INT will be valid.

TSADC_COMP1_LOW_INT

Address: Operational Base + offset (0x0084)

Bit	Attr	Reset Value	Description
31:12	RO	0x00000	reserved
11:0	RW	0x000	tsadc_comp_src1 TSADC low temperature level. TSADC output is lower than tsadc_comp, means the temperature is low. TSADC_LOW_INT will be valid.

14.5 Application Notes

14.5.1 Channel Select

The system has two Temperature Sensors, channel 0 is for CPU and channel 1 is for NPU.

14.5.2 Single-Sample Conversion

- The start timing

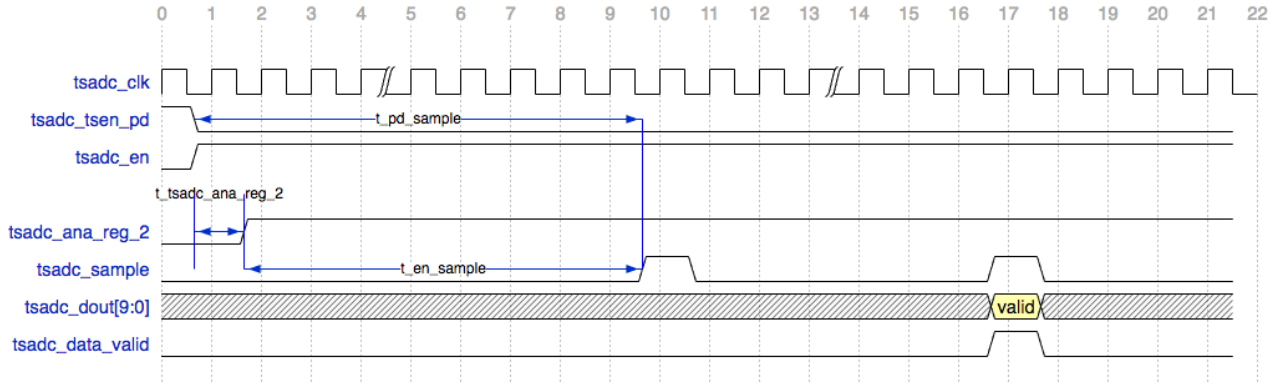


Fig. 14-2 Start Flow to Enable the Sensor and ADC

- Bypass mode(GRF_SOC_CON1[1] = 1'b1)
  When the ADC bypass mode is enable(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 untile the next clock cycles when the tsadc_sample is valid.

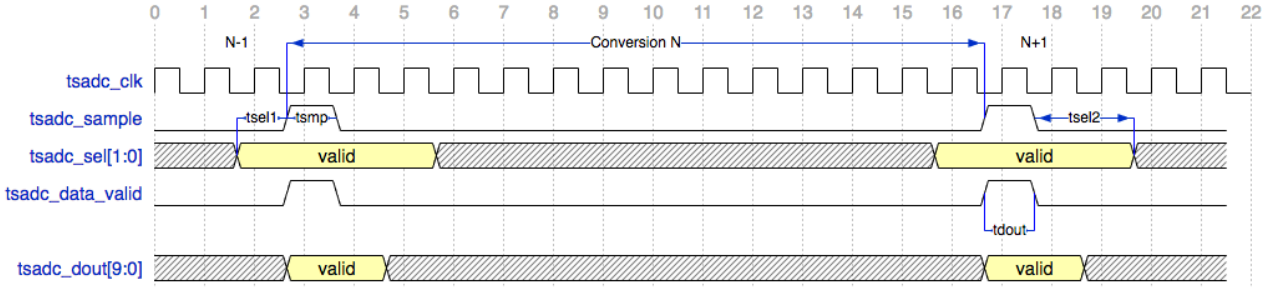


Fig. 14-3 TS-ADC Timing Diagram in Bypass Mode

- The Normal mode
  - tsadc_clk_sel = 1'b0(GRF_SOC_CON1[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.

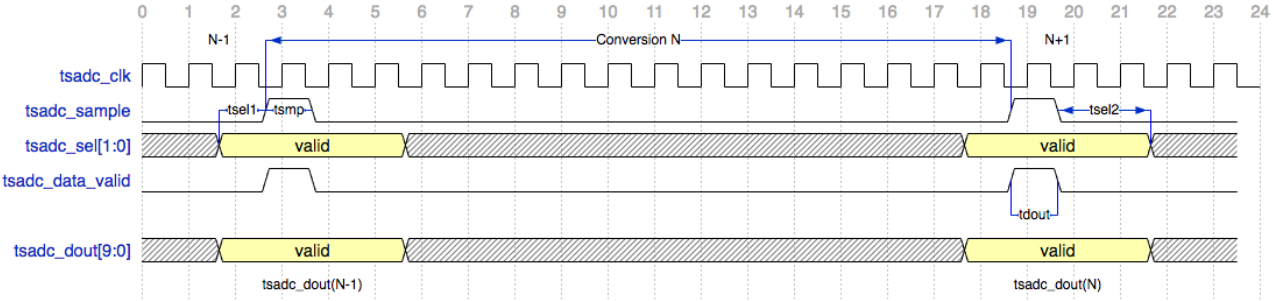


Fig. 14-4 TS-ADC Timing Diagram in Normal Mode with tsadc_clk_sel = 1'b0

  - tsadc_clk_sel = 1'b1(GRF_SOC_CON1[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.

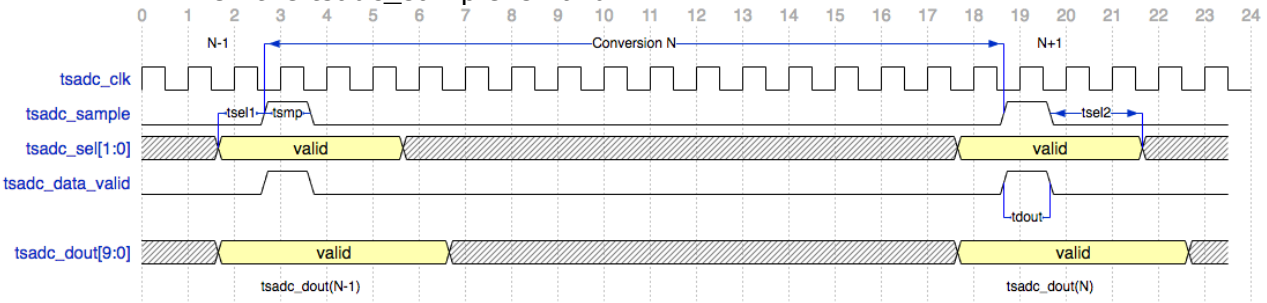


Fig. 14-5 TS-ADC Timing Diagram in Normal Mode with tsadc_clk_sel = 1'b1

14.5.3 Temperature-to-Code Mapping

Table 14-1 Temperature Code Mapping

Temperature (℃)	ADC Output Data (AUTO_CON[1] == 1'b0)		
	Min	Typ	Max
-40	-	2614.38	-
-20	-	2659.65	-
0	-	2704.91	-
25	-	2761.49	-
50	-	2818.07	-
75	-	2874.65	-
100	-	2931.23	-
125	-	2987.81	-

Note:
*Doutstd(T)=2.2632*T+2704.9*
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.

14.5.4 User-Define Mode

- In user-define mode, the PD_DVDD and CHSEL_DVDD are generate by setting register TSADC_USER_CON, bit[3] and bit[2:0]. In order to ensure timing between PD_DVDD and CHSEL_DVDD, the CHSEL_DVDD must be set before the PD_DVDD.
- In user-define mode, you can choose the method to control the START_OF_CONVERSION by setting bit[4] of TSADC_USER_CON. If set to 0, the start_of_conversion will be assert after "inter_pd_soc" cycles, which could be set by bit[11:6] of TSADC_USER_CON. And if start_mode was set 1, the start_of_conversion will be controlled by bit[5] of TSADC_USER_CON.
- Software can get the four channel temperature from TSADC_DATAn (n=0,1,2,3).

14.5.5 Automatic Mode

You can use the automatic mode with the following step:

- Set TSADC_AUTO_PERIOD to configure the interleave between every two accessing of TSADC in normal operation.
- Set TSADC_AUTO_PERIOD_HT to 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) to configure the high temperature level, if tsadc output is smaller than the value, means the temperature is high, tsadc_int will be asserted.
- Set TSADC_COMPn_SHUT(n=0,1) to configure the super high temperature level, if tsadc output is smaller than the value, means the temperature is too high, TSHUT will be asserted.
- Set TSADC_INT_EN to enable the high temperature interrupt for all channel; and you can also set TSHUT output to GPIO to reset the whole chip; and you can set TSHUT output to cru to reset the whole chip.
- Set TSADC_HIGHT_INT_DEBOUNCE and TSADC_HIGHT_TSHUT_DEBOUNCE, if the temperature is higher than COMP_INT or COMP_SHUT for "debounce" times, TSADC controller will generate interrupt or TSHUT.
    - Set TSADC_AUTO_CON to enable the TSADC controller.

Chapter 15 SAR-ADC

15.1 Overview

The ADC is a 6-channel signal-ended 10-bit Successive Approximation Register (SAR) A/D Converter. It uses the supply and ground as it reference which avoid use of any external reference. It converts the analog input signal into 10-bit binary digital codes at maximum conversion rate of 1MSPS with 12MHz A/D converter clock.

15.2 Block Diagram

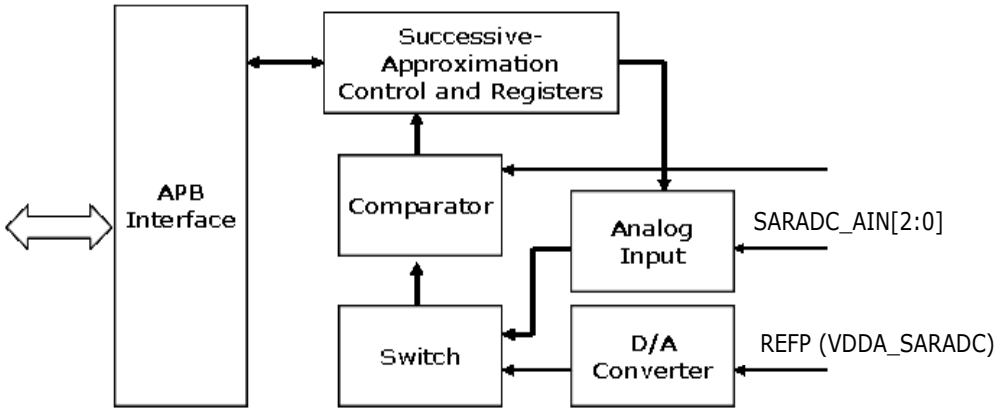


Fig. 15-1 SAR-ADC Block Diagram

Successive-Approximate Register and Control Logic Block

This block is exploited to realize binary search algorithm, storing the intermediate result and generate control signal for analog block.

Comparator Block

This block compares the analog input SARADC_AIN[2:0] with the voltage generated from D/A Converter, and output the comparison result to SAR and Control Logic Block for binary search. Three level amplifiers are employed in this comparator to provide enough gain.

15.3 Function Description

In RV1109/RV1126, SAR-ADC works at single-sample operation mode.

This mode is useful to sample an analog input when there is a gap between two samples to be converted. In this mode START is asserted only on the rising edge of CLKIN where conversion is needed. At the end of every conversion EOC signal is made high and valid output data is available at the rising edge of EOC. The detailed timing diagram will be shown in the following.

15.4 Register Description

15.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
SARADC_DATA	0x0000	W	0x00000000	Data after A/D Conversion
SARADC_STAS	0x0004	W	0x00000000	Status Register of A/D Converter
SARADC_CTRL	0x0008	W	0x00000000	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, **DW**-Double WORD (64 bits) access*

15.4.2 Detail Register Description

SARADC_DATA

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:10	RO	0x000000	reserved
9:0	RW	0x000	adc_data A/D value of the last conversion (DOUT[9:0]).

SARADC_STAS

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RW	0x0	adc_status ADC status (EOC). 1'b0: ADC stop 1'b1: Conversion in progress

SARADC_CTRL

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:7	RO	0x0000000	reserved
6	RW	0x0	int_status Interrupt status. This bit will be set to 1 when end-of-conversion. Set 0 to clear the interrupt.
5	RW	0x0	int_en Interrupt enable. 1'b0: Disable 1'b1: Enable
4	RO	0x0	reserved
3	RW	0x0	adc_power_ctrl ADC power down control bit. 1'b0: ADC power down 1'b1: ADC power up and reset Start signal will be asserted (DLY_PU_SOC + 2) sclk clock period later after power up.
2:0	RW	0x0	adc_input_src_sel ADC input source selection(CH_SEL[2:0]). 3'b000: Input source 0 (SARADC_AIN[0]) 3'b001: Input source 1 (SARADC_AIN[1]) 3'b010: Input source 2 (SARADC_AIN[2]) 3'b011: Input source 3 (SARADC_AIN[3]) 3'b100: Input source 4 (SARADC_AIN[4]) 3'b101: Input source 5 (SARADC_AIN[5]) Others: Reserved

SARADC_DLY_PU_SOC

Address: Operational Base + offset (0x000C)

Bit	Attr	Reset Value	Description
31:6	RO	0x0000000	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.

15.5 Timing Diagram

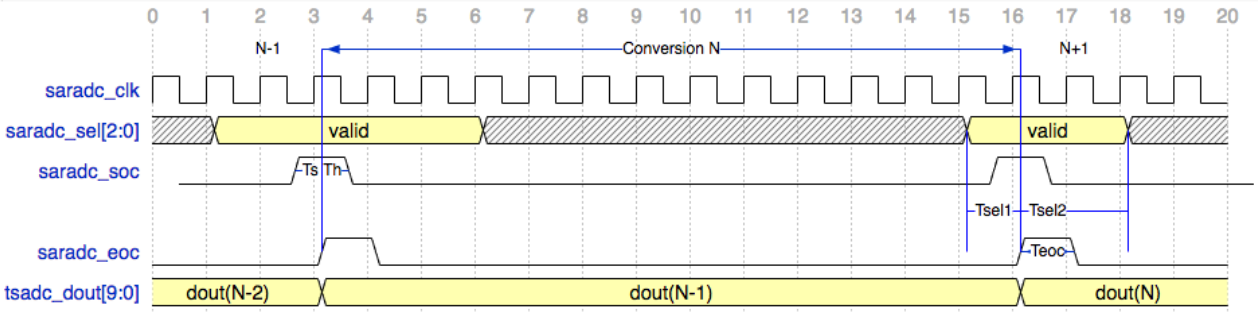


Fig. 15-2 SAR-ADC Timing Diagram in Single-sample Conversion Mode

The following table shows the detail value for timing parameters in the above diagram.

Table 15-1 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		0.72	0.8	0.88	V
Junction Temperature	Tj		-40		125	℃
Saradc Performance						
Resolution				10		bit
Differential Non-linearity	DNL			±1		LSB
Integral Non-linearity	INL			±1.5		LSB
Input Voltage Range	Vin		0.01* VREF		0.99* VREF	AVDD
Input Capacitance	Cin			6		pF
Sampling Rate	fs				1	MS/s
Timing Characteristic						
Clock Frequency	fCLK				12	MHz
Clock Period	tCLK		83.33			ns
Conversion Time			12			tCLK
Setup Time of soc signal	T_s		0			ns
Hold Time of soc signal	T_h		3			ns
EOC delay from rising edge of CLK	T_{d_EOC}				10	ns
B[9:0] delay from rising edge of CLK	T_{d_B}				10	ns

15.6 Application Notes

Steps of adc conversion:

- Write SARADC_CTRL[3] as 0 to power down ADC converter.

- Write SARADC_CTRL[2:0] as n to select ADC channel(n).
- Write SARADC_CTRL[5] as 1 to enable ADC interrupt.
- Write SARADC_CTRL[3] as 1 to power up ADC converter.
- Wait for ADC interrupt or poll SARADC_STAS register to assert whether the conversion is completed.
- Read the conversion result from SARADC_DATA[9:0].
- Note: The A/D converter was designed to operate at maximum 1MHz.

Chapter 16 DMA Controller (DMAC)

16.1 Overview

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

- Supports 27 peripheral requests
- Up to 64 bits data size
- 8 channel at the same time
- Up to burst 16
- 16 interrupts output and 1 abort output
- Supports 128 MFIFO depth

Following table shows the DMAC request mapping scheme.

Table 16-1 DMAC Request Mapping Table

Req number	Source	Polarity
0	SPI0 RX	High level
1	SPI0 TX	High level
2	SPI1 RX	High level
3	SPI1 TX	High level
4	UART0 RX	High level
5	UART0 TX	High level
6	UART1 RX	High level
7	UART1 TX	High level
8	UART2 RX	High level
9	UART2 TX	High level
10	UART3 RX	High level
11	UART3 TX	High level
12	UART4 RX	High level
13	UART4 TX	High level
14	UART5 RX	High level
15	UART5 TX	High level
16	PWM0	High level
17	PWM1	High level
18	PWM2	High level
19	I2S0 RX	High level
20	I2S0 TX	High level
21	I2S1 RX	High level
22	I2S1 TX	High level
23	I2S2 RX	High level
24	I2S2 TX	High level
25	PDM	High level
26	AudioPWM	High level

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

16.2 Block Diagram

Following figure shows the block diagram of DMAC.

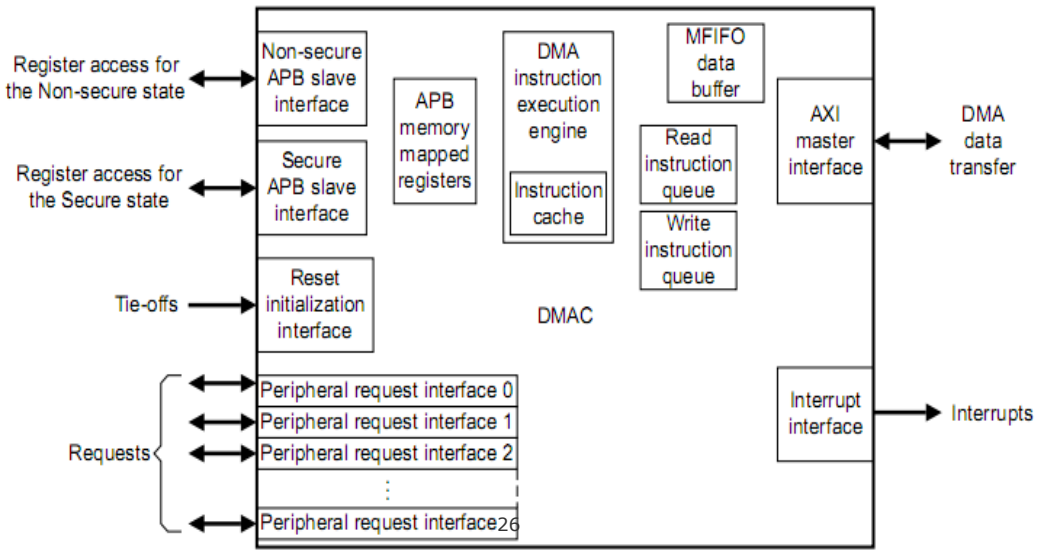


Fig. 16-1 Block Diagram of DMAC

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

16.3 Function Description

16.3.1 Introduction

The DMAC contains an instruction processing block that enables it to process program code that controls a DMA transfer. The program code is stored in a region of system memory that the DMAC accesses using its AXI interface. The DMAC stores instructions temporarily in a cache. It supports 8 channels, each channel capable of supporting a single concurrent thread of DMA operation. In addition, a single DMA manager thread exists, and you can use it to initialize the DMA channel threads. The DMAC executes up to one instruction for each AXI clock cycle. To ensure that it regularly executes each active thread, it alternates by processing the DMA manager thread and then a DMA channel thread. It uses a round-robin process when selecting the next active DMA channel thread to execute.

The DMAC uses variable-length instructions that consist of one to six bytes. It provides a separate Program Counter (PC) register for each DMA channel. When a thread requests an instruction from an address, the cache performs a look-up. If a cache hit occurs, then the cache immediately provides the data. Otherwise, the thread is stalled while the DMAC uses the AXI interface to perform a cache line fill. If an instruction is greater than 4 bytes, or spans the end of a cache line, the DMAC performs multiple cache accesses to fetch the instruction.

When a cache line fill is in progress, the DMAC enables other threads to access the cache, but if another cache miss occurs, this stalls the pipeline until the first line fill is complete. When a DMA channel thread executes a load or store instruction, the DMAC adds the instruction to the relevant read or write queue. The DMAC uses these queues as an instruction storage buffer prior to it issuing the instructions on the AXI bus. The DMAC also contains a Multi First-In-First-Out (MFIFO) data buffer that it uses to store data that it reads, or writes, during a DMA transfer.

16.3.2 Operating states

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

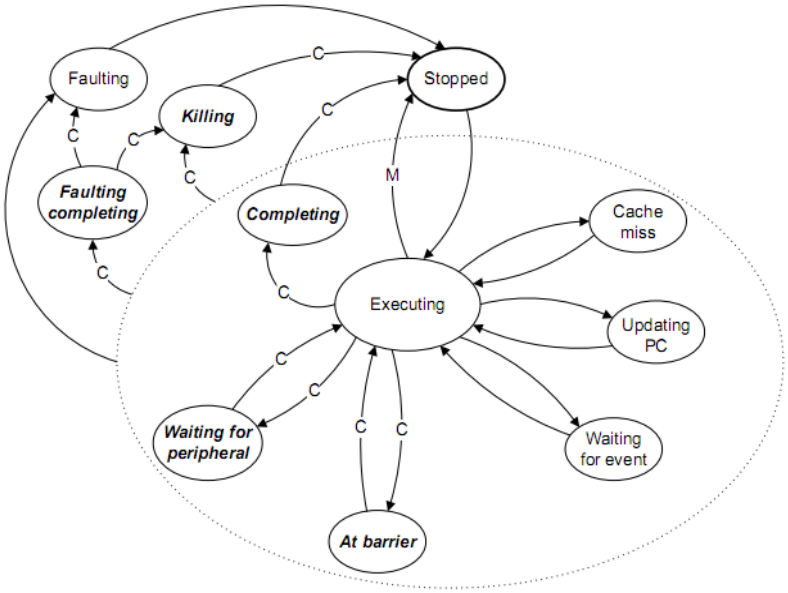


Fig. 16-2 DMAC Operation State

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

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

16.4 Register Description

16.4.1 Internal Address Mapping

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

16.4.2 Registers Summary

Name	Offset	Size	Reset Value	Description
DMAC_DSR	0x0000	W	0x00000000	DMA Manager Status Register
DMAC_DPC	0x0004	W	0x00000000	DMA Program Counter Register
DMAC_INTEN	0x0020	W	0x00000000	Interrupt Enable Register
DMAC_EVENT_RIS	0x0024	W	0x00000000	Event-Interrupt Raw Status Register
DMAC_INTMIS	0x0028	W	0x00000000	Interrupt Status Register
DMAC_INTCLR	0x002c	W	0x00000000	Interrupt Clear Register
DMAC_FSRD	0x0030	W	0x00000000	Fault Status DMA Manager Register
DMAC_FSRC	0x0034	W	0x00000000	Fault Status DMA Channel Register
DMAC_FTRD	0x0038	W	0x00000000	Fault Type DMA Manager Register
DMAC_FTR0	0x0040	W	0x00000000	Fault Type DMA Channel 0 Register
DMAC_FTR1	0x0044	W	0x00000000	Fault Type DMA Channel 1 Register
DMAC_FTR2	0x0048	W	0x00000000	Fault Type DMA Channel 2 Register

Name	Offset	Size	Reset Value	Description
DMAC_FTR3	0x004c	W	0x00000000	Fault Type DMA Channel 3 Register
DMAC_FTR4	0x0050	W	0x00000000	Fault Type DMA Channel 4 Register
DMAC_FTR5	0x0054	W	0x00000000	Fault Type DMA Channel 5 Register
DMAC_FTR6	0x0058	W	0x00000000	Fault Type DMA Channel 6 Register
DMAC_FTR7	0x005c	W	0x00000000	Fault Type DMA Channel 7 Register
DMAC_CSR0	0x0100	W	0x00000000	Channel 0 Status Register
DMAC_CPC0	0x0104	W	0x00000000	Channel 0 Program Counter Register
DMAC_CSR1	0x0108	W	0x00000000	Channel 1 Status Register
DMAC_CPC1	0x010c	W	0x00000000	Channel 1 Program Counter Register
DMAC_CSR2	0x0110	W	0x00000000	Channel 2 Status Register
DMAC_CPC2	0x0114	W	0x00000000	Channel 2 Program Counter Register
DMAC_CSR3	0x0118	W	0x00000000	Channel 3 Status Register
DMAC_CPC3	0x011c	W	0x00000000	Channel 3 Program Counter Register
DMAC_CSR4	0x0120	W	0x00000000	Channel 4 Status Register
DMAC_CPC4	0x0124	W	0x00000000	Channel 4 Program Counter Register
DMAC_CSR5	0x0128	W	0x00000000	Channel 5 Status Register
DMAC_CPC5	0x012c	W	0x00000000	Channel 5 Program Counter Register
DMAC_CSR6	0x0130	W	0x00000000	Channel 6 Status Register
DMAC_CPC6	0x0134	W	0x00000000	Channel 6 Program Counter Register
DMAC_CSR7	0x0138	W	0x00000000	Channel 7 Status Register
DMAC_CPC7	0x013c	W	0x00000000	Channel 7 Program Counter Register
DMAC_SAR0	0x0400	W	0x00000000	Channel 0 Source Address Register
DMAC_DAR0	0x0404	W	0x00000000	Channel 0 Destination Address Register
DMAC_CCR0	0x0408	W	0x00000000	Channel 0 Channel Control Register
DMAC_LC0_0	0x040c	W	0x00000000	Channel 0 Loop Counter 0 Register

Name	Offset	Size	Reset Value	Description
DMAC_LC1_0	0x0410	W	0x00000000	Channel 0 Loop Counter 1 Register
DMAC_SAR1	0x0420	W	0x00000000	Channel 1 Source Address Register
DMAC_DAR1	0x0424	W	0x00000000	Channel 1 Destination Address Register
DMAC_CCR1	0x0428	W	0x00000000	Channel 1 Channel Control Register
DMAC_LC0_1	0x042c	W	0x00000000	Channel 1 Loop Counter 0 Register
DMAC_LC1_1	0x0430	W	0x00000000	Channel 1 Loop Counter 1 Register
DMAC_SAR2	0x0440	W	0x00000000	Channel 2 Source Address Register
DMAC_DAR2	0x0444	W	0x00000000	Channel 2 Destination Address Register
DMAC_CCR2	0x0448	W	0x00000000	Channel 2 Channel Control Register
DMAC_LC0_2	0x044c	W	0x00000000	Channel 2 Loop Counter 0 Register
DMAC_LC1_2	0x0450	W	0x00000000	Channel 2 Loop Counter 1 Register
DMAC_SAR3	0x0460	W	0x00000000	Channel 3 Source Address Register
DMAC_DAR3	0x0464	W	0x00000000	Channel 3 Destination Address Register
DMAC_CCR3	0x0468	W	0x00000000	Channel 3 Channel Control Register
DMAC_LC0_3	0x046c	W	0x00000000	Channel 3 Loop Counter 0 Register
DMAC_LC1_3	0x0470	W	0x00000000	Channel 3 Loop Counter 1 Register
DMAC_SAR4	0x0480	W	0x00000000	Channel 4 Address Register
DMAC_DAR4	0x0484	W	0x00000000	Channel 4 Destination Address Register
DMAC_CCR4	0x0488	W	0x00000000	Channel 4 Channel Control Register
DMAC_LC0_4	0x048c	W	0x00000000	Channel 4 Loop Counter 0 Register
DMAC_LC1_4	0x0490	W	0x00000000	Channel 4 Loop Counter 1 Register
DMAC_SAR5	0x04a0	W	0x00000000	Channel 5 Address Register

Name	Offset	Size	Reset Value	Description
DMAC_DAR5	0x04a4	W	0x00000000	Channel 5 Destination Address Register
DMAC_CCR5	0x04a8	W	0x00000000	Channel 5 Channel Control Register
DMAC_LC0_5	0x04ac	W	0x00000000	Channel 5 Loop Counter 0 Register
DMAC_LC1_5	0x04b0	W	0x00000000	Channel 5 Loop Counter 1 Register
DMAC_SAR6	0x04c0	W	0x00000000	Channel 6 Source Address Register
DMAC_DAR6	0x04c4	W	0x00000000	Channel 6 Destination Address Register
DMAC_CCR6	0x04c8	W	0x00000000	Channel 6 Channel Control Register
DMAC_LC0_6	0x04cc	W	0x00000000	Channel 6 Loop Counter 0 Register
DMAC_LC1_6	0x04d0	W	0x00000000	Channel 6 Loop Counter 1 Register
DMAC_SAR7	0x04e0	W	0x00000000	Channel 7 Source Address Register
DMAC_DAR7	0x04e4	W	0x00000000	Channel 7 Destination Address Register
DMAC_CCR7	0x04e8	W	0x00000000	Channel 7 Channel Control Register
DMAC_LC0_7	0x04ec	W	0x00000000	Channel 7 Loop Counter 0 Register
DMAC_LC1_7	0x04f0	W	0x00000000	Channel 7 Loop Counter 1 Register
DMAC_DBGSTATUS	0x0d00	W	0x00000000	Debug Status Register
DMAC_DBGCMD	0x0d04	W	0x00000000	Debug Command Register
DMAC_DBGINST0	0x0d08	W	0x00000000	Debug Instruction-0 Register
DMAC_DBGINST1	0x0d0c	W	0x00000000	Debug Instruction-1 Register
DMAC_CR0	0x0e00	W	0x001f3075	Configuration Register 0
DMAC_CR1	0x0e04	W	0x000000b5	Configuration Register 1
DMAC_CR2	0x0e08	W	0x00000000	Configuration Register 2
DMAC_CR3	0x0e0c	W	0x0000ffff	Configuration Register 3
DMAC_CR4	0x0e10	W	0x000fffff	Configuration Register 4
DMAC_CRDn	0x0e14	W	0x07ff7f73	Configuration Register
DMAC_WD	0x0e80	W	0x00000000	DMA Watchdog Register

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

16.4.3 Detail Register Description

DMAC_DSR

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9	RO	0x0	dns 1'b0: DMA manager operates in the Secure state 1'b1: DMA manager operates in the Non-secure state
8:4	RO	0x00	wakeup_event 5'h0: event[0] 5'h1: event[1] 5'h2: event[2] … 5'h1f: event[31]
3:0	RO	0x0	dma_status 4'h0: Stopped 4'h1: Executing 4'h2: Cache miss 4'h3: Updating PC 4'h4: Waiting for event 4'hf: Faulting Others: Reserved

DMAC_DPC

Address: Operational Base + offset (0x0004)

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

DMAC_INTEN

Address: Operational Base + offset (0x0020)

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

DMAC_EVENT_RIS

Address: Operational Base + offset (0x0024)

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

DMAC_INTMIS

Address: Operational Base + offset (0x0028)

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

DMAC_INTCLR

Address: Operational Base + offset (0x002c)

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

DMAC_FSRD

Address: Operational Base + offset (0x0030)

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

DMAC_FSRC

Address: Operational Base + offset (0x0034)

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

DMAC_FTRD

Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30	RO	0x0	dbg_instr Memory or from the debug interface. 1'b0: Instruction that generated an abort was read from system memory 1'b1: Instruction that generated an abort was read from the debug interface
29:17	RO	0x0	reserved

Bit	Attr	Reset Value	Description
16	RO	0x0	instr_fetch_err Performs an instruction fetch. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response
15:6	RO	0x0	reserved
5	RO	0x0	mgr_evnt_err 1'b0: DMA manager has appropriate security to execute DMAWFE or DMASEV 1'b1: DMA manager thread in the Non-secure state attempted to execute either: a. DMAWFE to wait for a secure event b. DMASEV to create a secure event or secure interrupt
4	RO	0x0	dmago_err 1'b0: DMA manager has appropriate security to execute DMAGO 1'b1: DMA manager thread in the Non-secure state attempted to execute DMAGO to create a DMA channel operating in the Secure state
3:2	RO	0x0	reserved
1	RO	0x0	operand_invalid The configuration of the DMAC. 1'b0: Valid operand 1'b1: Invalid operand
0	RW	0x0	undef_instr 1'b0: Defined instruction 1'b1: Undefined instruction

DMAC_FTR0

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31	RO	0x0	lockup_err 1'b0: DMA channel has adequate resources 1'b1: DMA channel has locked-up because of insufficient resources This fault is an imprecise abort.
30	RO	0x0	dbg_instr Memory or from the debug interface. 1'b0: Instruction that generated an abort was read from system memory 1'b1: Instruction that generated an abort was read from the debug interface This fault is an imprecise abort but the bit is only valid when a precise abort occurs.
29:19	RO	0x0	reserved

Bit	Attr	Reset Value	Description
18	RO	0x0	data_read_err Thread performs a data read. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is an imprecise abort.
17	RO	0x0	data_write_err Thread performs a data write. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is an imprecise abort.
16	RO	0x0	instr_fetch_err Thread performs an instruction fetch. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is a precise abort.
15:14	RO	0x0	reserved
13	RO	0x0	st_data_unavailable 1'b0: MFIFO contains all the data to enable the DMAST to complete 1'b1: Previous DMALDs have not put enough data in the MFIFO to enable the DMAST to complete This fault is a precise abort.
12	RO	0x0	mfifo_err DMALD 1'b0: MFIFO contains sufficient space 1'b1: MFIFO is too small to hold the data that DMALD requires DMAST 1'b0: MFIFO contains sufficient data 1'b1: MFIFO is too small to store the data to enable DMAST to complete This fault is an imprecise abort.
11:8	RO	0x0	reserved
7	RO	0x0	ch_rdwr_err To perform a secure read or secure write. 1'b0: A DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: A DMA channel thread in the Non-secure state attempted to perform a secure read or secure write This fault is a precise abort.

Bit	Attr	Reset Value	Description
6	RO	0x0	ch_periph_err DMASTP, or DMAFLUSHP with inappropriate security permissions. 1'b0: a DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: a DMA channel thread in the Non-secure state attempted to execute either: a. DMAWFP to wait for a secure peripheral b. DMALDP or DMASTP to notify a secure peripheral c. DMAFLUSHP to flush a secure peripheral This fault is a precise abort.
5	RO	0x0	ch_event_err 1'b0: A DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: A DMA channel thread in the Non-secure state attempted to execute either: a. DMAWFE to wait for a secure event b. DMASEV to create a secure event or secure interrupt This fault is a precise abort.
4:2	RO	0x0	reserved
1	RO	0x0	operand_invalid Valid for the configuration of the DMAC. 1'b0: Valid operand 1'b1: Invalid operand This fault is a precise abort.
0	RO	0x0	undef_instr 1'b0: Defined instruction 1'b1: Undefined instruction This fault is a precise abort.

DMAC_FTR1

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31	RO	0x0	lockup_err 1'b0: DMA channel has adequate resources 1'b1: DMA channel has locked-up because of insufficient resources This fault is an imprecise abort.
30	RO	0x0	dbg_instr Memory or from the debug interface. 1'b0: Instruction that generated an abort was read from system memory 1'b1: Instruction that generated an abort was read from the debug interface This fault is an imprecise abort but the bit is only valid when a precise abort occurs.
29:19	RO	0x0	reserved

Bit	Attr	Reset Value	Description
18	RO	0x0	data_read_err Thread performs a data read. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is an imprecise abort.
17	RO	0x0	data_write_err Thread performs a data write. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is an imprecise abort.
16	RO	0x0	instr_fetch_err Thread performs an instruction fetch. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is a precise abort.
15:14	RO	0x0	reserved
13	RO	0x0	st_data_unavailable 1'b0: MFIFO contains all the data to enable the DMAST to complete 1'b1: Previous DMALDs have not put enough data in the MFIFO to enable the DMAST to complete This fault is a precise abort.
12	RO	0x0	mfifo_err DMALD 1'b0: MFIFO contains sufficient space 1'b1: MFIFO is too small to hold the data that DMALD requires DMAST 1'b0: MFIFO contains sufficient data 1'b1: MFIFO is too small to store the data to enable DMAST to complete This fault is an imprecise abort.
11:8	RO	0x0	reserved
7	RO	0x0	ch_rdwr_err To perform a secure read or secure write. 1'b0: A DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: A DMA channel thread in the Non-secure state attempted to perform a secure read or secure write This fault is a precise abort.

Bit	Attr	Reset Value	Description
6	RO	0x0	ch_periph_err DMASTP, or DMAFLUSHP with inappropriate security permissions. 1'b0: A DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: A DMA channel thread in the Non-secure state attempted to execute either: a. DMAWFP to wait for a secure peripheral b. DMALDP or DMASTP to notify a secure peripheral c. DMAFLUSHP to flush a secure peripheral This fault is a precise abort.
5	RO	0x0	ch_event_err 1'b0: A DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: A DMA channel thread in the Non-secure state attempted to execute either: a. DMAWFE to wait for a secure event b. DMASEV to create a secure event or secure interrupt This fault is a precise abort.
4:2	RO	0x0	reserved
1	RO	0x0	operand_invalid Valid for the configuration of the DMAC. 1'b0: Valid operand 1'b1: Invalid operand This fault is a precise abort.
0	RO	0x0	undef_instr 1'b0: Defined instruction 1'b1: Undefined instruction This fault is a precise abort.

DMAC_FTR2

Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31	RO	0x0	lockup_err 1'b0: DMA channel has adequate resources 1'b1: DMA channel has locked-up because of insufficient resources This fault is an imprecise abort.
30	RO	0x0	dbg_instr Memory or from the debug interface. 1'b0: Instruction that generated an abort was read from system memory 1'b1: Instruction that generated an abort was read from the debug interface This fault is an imprecise abort but the bit is only valid when a precise abort occurs.
29:19	RO	0x0	reserved

Bit	Attr	Reset Value	Description
18	RO	0x0	data_read_err Thread performs a data read. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is an imprecise abort.
17	RO	0x0	data_write_err Thread performs a data write. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is an imprecise abort.
16	RO	0x0	instr_fetch_err Thread performs an instruction fetch. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is a precise abort.
15:14	RO	0x0	reserved
13	RO	0x0	st_data_unavailable 1'b0: MFIFO contains all the data to enable the DMAST to complete 1'b1: Previous DMALDs have not put enough data in the MFIFO to enable the DMAST to complete This fault is a precise abort.
12	RO	0x0	mfifo_err DMALD 1'b0: MFIFO contains sufficient space 1'b1: MFIFO is too small to hold the data that DMALD requires DMAST 1'b0: MFIFO contains sufficient data 1'b1: MFIFO is too small to store the data to enable DMAST to complete This fault is an imprecise abort.
11:8	RO	0x0	reserved
7	RO	0x0	ch_rdwr_err To perform a secure read or secure write. 1'b0: A DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: A DMA channel thread in the Non-secure state attempted to perform a secure read or secure write This fault is a precise abort.

Bit	Attr	Reset Value	Description
6	RO	0x0	ch_periph_err DMASTP, or DMAFLUSHP with inappropriate security permissions. 1'b0: A DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: A DMA channel thread in the Non-secure state attempted to execute either: a. DMAWFP to wait for a secure peripheral b. DMALDP or DMASTP to notify a secure peripheral c. DMAFLUSHP to flush a secure peripheral This fault is a precise abort.
5	RO	0x0	ch_event_err 1'b0: A DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: A DMA channel thread in the Non-secure state attempted to execute either: a. DMAWFE to wait for a secure event b. DMASEV to create a secure event or secure interrupt This fault is a precise abort.
4:2	RO	0x0	reserved
1	RO	0x0	operand_invalid Valid for the configuration of the DMAC. 1'b0: Valid operand 1'b1: Invalid operand This fault is a precise abort.
0	RO	0x0	undef_instr 1'b0: Defined instruction 1'b1: Undefined instruction This fault is a precise abort.

DMAC_FTR3

Address: Operational Base + offset (0x004c)

Bit	Attr	Reset Value	Description
31	RO	0x0	lockup_err 1'b0: DMA channel has adequate resources 1'b1: DMA channel has locked-up because of insufficient resources This fault is an imprecise abort.
30	RO	0x0	dbg_instr Memory or from the debug interface. 1'b0: Instruction that generated an abort was read from system memory 1'b1: Instruction that generated an abort was read from the debug interface This fault is an imprecise abort but the bit is only valid when a precise abort occurs.

Bit	Attr	Reset Value	Description
29:19	RO	0x0	reserved
18	RO	0x0	data_read_err Thread performs a data read. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is an imprecise abort.
17	RO	0x0	data_write_err Thread performs a data write. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is an imprecise abort.
16	RO	0x0	instr_fetch_err Thread performs an instruction fetch. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is a precise abort.
15:14	RO	0x0	reserved
13	RO	0x0	st_data_unavailable 1'b0: MFIFO contains all the data to enable the DMAST to complete 1'b1: Previous DMALDs have not put enough data in the MFIFO to enable the DMAST to complete This fault is a precise abort.
12	RO	0x0	mfifo_err DMALD 1'b0: MFIFO contains sufficient space 1'b1: MFIFO is too small to hold the data that DMALD requires DMAST 1'b0: MFIFO contains sufficient data 1'b1: MFIFO is too small to store the data to enable DMAST to complete This fault is an imprecise abort.
11:8	RO	0x0	reserved
7	RO	0x0	ch_rdwr_err To perform a secure read or secure write. 1'b0: A DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: A DMA channel thread in the Non-secure state attempted to perform a secure read or secure write This fault is a precise abort.

Bit	Attr	Reset Value	Description
6	RO	0x0	ch_periph_err DMASTP, or DMAFLUSHP with inappropriate security permissions. 1'b0: A DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: A DMA channel thread in the Non-secure state attempted to execute either: a. DMAWFP to wait for a secure peripheral b. DMALDP or DMASTP to notify a secure peripheral c. DMAFLUSHP to flush a secure peripheral This fault is a precise abort.
5	RO	0x0	ch_event_err 1'b0: A DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: A DMA channel thread in the Non-secure state attempted to execute either: a. DMAWFE to wait for a secure event b. DMASEV to create a secure event or secure interrupt This fault is a precise abort.
4:2	RO	0x0	reserved
1	RO	0x0	operand_invalid Valid for the configuration of the DMAC. 1'b0: Valid operand 1'b1: Invalid operand This fault is a precise abort.
0	RO	0x0	undef_instr 1'b0: Defined instruction 1'b1: Undefined instruction This fault is a precise abort.

DMAC_FTR4

Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31	RO	0x0	lockup_err 1'b0: DMA channel has adequate resources 1'b1: DMA channel has locked-up because of insufficient resources This fault is an imprecise abort.
30	RO	0x0	dbg_instr Memory or from the debug interface. 1'b0: Instruction that generated an abort was read from system memory 1'b1: Instruction that generated an abort was read from the debug interface This fault is an imprecise abort but the bit is only valid when a precise abort occurs.
29:19	RO	0x0	reserved

Bit	Attr	Reset Value	Description
18	RO	0x0	data_read_err Thread performs a data read. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is an imprecise abort.
17	RO	0x0	data_write_err Thread performs a data write. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is an imprecise abort.
16	RO	0x0	instr_fetch_err Thread performs an instruction fetch. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is a precise abort.
15:14	RO	0x0	reserved
13	RO	0x0	st_data_unavailable 1'b0: MFIFO contains all the data to enable the DMAST to complete 1'b1: Previous DMALDs have not put enough data in the MFIFO to enable the DMAST to complete This fault is a precise abort.
12	RO	0x0	mfifo_err DMALD 1'b0: MFIFO contains sufficient space 1'b1: MFIFO is too small to hold the data that DMALD requires DMAST 1'b0: MFIFO contains sufficient data 1'b1: MFIFO is too small to store the data to enable DMAST to complete This fault is an imprecise abort.
11:8	RO	0x0	reserved
7	RO	0x0	ch_rdwr_err To perform a secure read or secure write. 1'b0: A DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: A DMA channel thread in the Non-secure state attempted to perform a secure read or secure write This fault is a precise abort.

Bit	Attr	Reset Value	Description
6	RO	0x0	ch_periph_err DMASTP, or DMAFLUSHP with inappropriate security permissions. 1'b0: A DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: A DMA channel thread in the Non-secure state attempted to execute either: a. DMAWFP to wait for a secure peripheral b. DMALDP or DMASTP to notify a secure peripheral c. DMAFLUSHP to flush a secure peripheral This fault is a precise abort.
5	RO	0x0	ch_event_err 1'b0: A DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: A DMA channel thread in the Non-secure state attempted to execute either: a. DMAWFE to wait for a secure event b. DMASEV to create a secure event or secure interrupt This fault is a precise abort.
4:2	RO	0x0	reserved
1	RO	0x0	operand_invalid Valid for the configuration of the DMAC. 1'b0: Valid operand 1'b1: Invalid operand This fault is a precise abort.
0	RO	0x0	undef_instr 1'b0: Defined instruction 1'b1: Undefined instruction This fault is a precise abort.

DMAC_FTR5

Address: Operational Base + offset (0x0054)

Bit	Attr	Reset Value	Description
31	RO	0x0	lockup_err 1'b0: DMA channel has adequate resources 1'b1: DMA channel has locked-up because of insufficient resources This fault is an imprecise abort.
30	RO	0x0	dbg_instr Memory or from the debug interface. 1'b0: Instruction that generated an abort was read from system memory 1'b1: Instruction that generated an abort was read from the debug interface This fault is an imprecise abort but the bit is only valid when a precise abort occurs.
29:19	RO	0x0	reserved

Bit	Attr	Reset Value	Description
18	RO	0x0	data_read_err Thread performs a data read. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is an imprecise abort.
17	RO	0x0	data_write_err Thread performs a data write. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is an imprecise abort.
16	RO	0x0	instr_fetch_err Thread performs an instruction fetch. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is a precise abort.
15:14	RO	0x0	reserved
13	RO	0x0	st_data_unavailable 1'b0: MFIFO contains all the data to enable the DMAST to complete 1'b1: Previous DMALDs have not put enough data in the MFIFO to enable the DMAST to complete This fault is a precise abort.
12	RO	0x0	mfifo_err DMALD 1'b0: MFIFO contains sufficient space 1'b1: MFIFO is too small to hold the data that DMALD requires DMAST 1'b0: MFIFO contains sufficient data 1'b1: MFIFO is too small to store the data to enable DMAST to complete This fault is an imprecise abort.
11:8	RO	0x0	reserved
7	RO	0x0	ch_rdwr_err To perform a secure read or secure write. 1'b0: A DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: A DMA channel thread in the Non-secure state attempted to perform a secure read or secure write This fault is a precise abort.

Bit	Attr	Reset Value	Description
6	RO	0x0	ch_periph_err DMASTP, or DMAFLUSHP with inappropriate security permissions. 1'b0: A DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: A DMA channel thread in the Non-secure state attempted to execute either: a. DMAWFP to wait for a secure peripheral b. DMALDP or DMASTP to notify a secure peripheral c. DMAFLUSHP to flush a secure peripheral This fault is a precise abort.
5	RO	0x0	ch_event_err 1'b0: A DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: A DMA channel thread in the Non-secure state attempted to execute either: a. DMAWFE to wait for a secure event b. DMASEV to create a secure event or secure interrupt This fault is a precise abort.
4:2	RO	0x0	reserved
1	RO	0x0	operand_invalid Valid for the configuration of the DMAC. 1'b0: Valid operand 1'b1: Invalid operand This fault is a precise abort.
0	RO	0x0	undef_instr 1'b0: Defined instruction 1'b1: Undefined instruction This fault is a precise abort.

DMAC_FTR6

Address: Operational Base + offset (0x0058)

Bit	Attr	Reset Value	Description
31	RO	0x0	lockup_err 1'b0: DMA channel has adequate resources 1'b1: DMA channel has locked-up because of insufficient resources This fault is an imprecise abort.
30	RO	0x0	dbg_instr Memory or from the debug interface. 1'b0: Instruction that generated an abort was read from system memory 1'b1: Instruction that generated an abort was read from the debug interface This fault is an imprecise abort but the bit is only valid when a precise abort occurs.
29:19	RO	0x0	reserved

Bit	Attr	Reset Value	Description
18	RO	0x0	data_read_err Thread performs a data read. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is an imprecise abort.
17	RO	0x0	data_write_err Thread performs a data write. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is an imprecise abort.
16	RO	0x0	instr_fetch_err Thread performs an instruction fetch. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is a precise abort.
15:14	RO	0x0	reserved
13	RO	0x0	st_data_unavailable 1'b0: MFIFO contains all the data to enable the DMAST to complete 1'b1: Previous DMALDs have not put enough data in the MFIFO to enable the DMAST to complete This fault is a precise abort.
12	RO	0x0	mfifo_err DMALD 1'b0: MFIFO contains sufficient space 1'b1: MFIFO is too small to hold the data that DMALD requires DMAST 1'b0: MFIFO contains sufficient data 1'b1: MFIFO is too small to store the data to enable DMAST to complete This fault is an imprecise abort.
11:8	RO	0x0	reserved
7	RO	0x0	ch_rdwr_err To perform a secure read or secure write. 1'b0: A DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: A DMA channel thread in the Non-secure state attempted to perform a secure read or secure write This fault is a precise abort.

Bit	Attr	Reset Value	Description
6	RO	0x0	ch_periph_err DMASTP, or DMAFLUSHP with inappropriate security permissions. 1'b0: A DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: A DMA channel thread in the Non-secure state attempted to execute either: a. DMAWFP to wait for a secure peripheral b. DMALDP or DMASTP to notify a secure peripheral c. DMAFLUSHP to flush a secure peripheral This fault is a precise abort.
5	RO	0x0	ch_event_err 1'b0: A DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: A DMA channel thread in the Non-secure state attempted to execute either: a. DMAWFE to wait for a secure event b. DMASEV to create a secure event or secure interrupt This fault is a precise abort.
4:2	RO	0x0	reserved
1	RO	0x0	operand_invalid Valid for the configuration of the DMAC. 1'b0: Valid operand 1'b1: Invalid operand This fault is a precise abort.
0	RO	0x0	undef_instr 1'b0: Defined instruction 1'b1: Undefined instruction This fault is a precise abort.

DMAC_FTR7

Address: Operational Base + offset (0x005c)

Bit	Attr	Reset Value	Description
31	RO	0x0	lockup_err 1'b0: DMA channel has adequate resources 1'b1: DMA channel has locked-up because of insufficient resources This fault is an imprecise abort.
30	RO	0x0	dbg_instr Memory or from the debug interface. 1'b0: Instruction that generated an abort was read from system memory 1'b1: Instruction that generated an abort was read from the debug interface This fault is an imprecise abort but the bit is only valid when a precise abort occurs.
29:19	RO	0x0	reserved

Bit	Attr	Reset Value	Description
18	RO	0x0	data_read_err Thread performs a data read. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is an imprecise abort.
17	RO	0x0	data_write_err Thread performs a data write. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is an imprecise abort.
16	RO	0x0	instr_fetch_err Thread performs an instruction fetch. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is a precise abort.
15:14	RO	0x0	reserved
13	RO	0x0	st_data_unavailable 1'b0: MFIFO contains all the data to enable the DMAST to complete 1'b1: Previous DMALDs have not put enough data in the MFIFO to enable the DMAST to complete This fault is a precise abort.
12	RO	0x0	mfifo_err DMALD 1'b0: MFIFO contains sufficient space 1'b1: MFIFO is too small to hold the data that DMALD requires DMAST 1'b0: MFIFO contains sufficient data 1'b1: MFIFO is too small to store the data to enable DMAST to complete This fault is an imprecise abort.
11:8	RO	0x0	reserved
7	RO	0x0	ch_rdwr_err To perform a secure read or secure write. 1'b0: A DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: A DMA channel thread in the Non-secure state attempted to perform a secure read or secure write This fault is a precise abort.

Bit	Attr	Reset Value	Description
6	RO	0x0	ch_periph_err DMASTP, or DMAFLUSHP with inappropriate security permissions. 1'b0: A DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: A DMA channel thread in the Non-secure state attempted to execute either: a. DMAWFP to wait for a secure peripheral b. DMALDP or DMASTP to notify a secure peripheral c. DMAFLUSHP to flush a secure peripheral This fault is a precise abort.
5	RO	0x0	ch_event_err 1'b0: A DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: A DMA channel thread in the Non-secure state attempted to execute either: a. DMAWFE to wait for a secure event b. DMASEV to create a secure event or secure interrupt This fault is a precise abort.
4:2	RO	0x0	reserved
1	RO	0x0	operand_invalid Valid for the configuration of the DMAC. 1'b0: Valid operand 1'b1: Invalid operand This fault is a precise abort.
0	RO	0x0	undef_instr 1'b0: Defined instruction 1'b1: Undefined instruction This fault is a precise abort.

DMAC_CSR0

Address: Operational Base + offset (0x0100)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RO	0x0	cns 1'b0: DMA channel operates in the Secure state 1'b1: DMA channel operates in the Non-secure state
20:16	RO	0x0	reserved
15	RO	0x0	dmawfp_periph 1'b0: DMAWFP executed with the periph operand not set 1'b1: DMAWFP executed with the periph operand set
14	RO	0x0	dmawfp_b_ns 1'b0: DMAWFP executed with the single operand set 1'b1: DMAWFP executed with the burst operand set
13:9	RO	0x0	reserved

Bit	Attr	Reset Value	Description
8:4	RO	0x00	wakeup_number Indicate the event or peripheral number that the channel is waiting for. 5'h0: DMA channel is waiting for event, or peripheral, 0 5'h1: DMA channel is waiting for event, or peripheral, 1 5'h2: DMA channel is waiting for event, or peripheral, 2 … 5'h1f: DMA channel is waiting for event, or peripheral, 31
3:0	RO	0x0	channel_status Channel 0 status. 4'h0: Stopped 4'h1: Executing 4'h2: Cache miss 4'h3: Updating PC 4'h4: Waiting for event 4'h5: At barrier 4'h7: Waiting for peripheral 4'h8: Killing 4'h9: Completing 4'he: Faulting completing 4'hf: Faulting Others: Reserved

DMAC_CPC0

Address: Operational Base + offset (0x0104)

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

DMAC_CSR1

Address: Operational Base + offset (0x0108)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RO	0x0	cns 1'b0: DMA channel operates in the Secure state 1'b1: DMA channel operates in the Non-secure state
20:16	RO	0x0	reserved
15	RO	0x0	dmawfp_periph 1'b0: DMAWFP executed with the periph operand not set 1'b1: DMAWFP executed with the periph operand set
14	RO	0x0	dmawfp_b_ns 1'b0: DMAWFP executed with the single operand set 1'b1: DMAWFP executed with the burst operand set
13:9	RO	0x0	reserved

Bit	Attr	Reset Value	Description
8:4	RO	0x00	wakeup_number Indicate the event or peripheral number that the channel is waiting for. 5'h0: DMA channel is waiting for event, or peripheral, 0 5'h1: DMA channel is waiting for event, or peripheral, 1 5'h2: DMA channel is waiting for event, or peripheral, 2 … 5'h1f: DMA channel is waiting for event, or peripheral, 31
3:0	RO	0x0	channel_status Channel 1 status. 4'h0: Stopped 4'h1: Executing 4'h2: Cache miss 4'h3: Updating PC 4'h4: Waiting for event 4'h5: At barrier 4'h7: Waiting for peripheral 4'h8: Killing 4'h9: Completing 4'he: Faulting completing 4'hf: Faulting Others: Reserved

DMAC_CPC1

Address: Operational Base + offset (0x010c)

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

DMAC_CSR2

Address: Operational Base + offset (0x0110)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RO	0x0	cns 1'b0: DMA channel operates in the Secure state 1'b1: DMA channel operates in the Non-secure state
20:16	RO	0x0	reserved
15	RO	0x0	dmawfp_periph 1'b0: DMAWFP executed with the periph operand not set 1'b1: DMAWFP executed with the periph operand set
14	RO	0x0	dmawfp_b_ns 1'b0: DMAWFP executed with the single operand set 1'b1: DMAWFP executed with the burst operand set
13:9	RO	0x0	reserved

Bit	Attr	Reset Value	Description
8:4	RO	0x00	wakeup_number Indicate the event or peripheral number that the channel is waiting for. 5'h0: DMA channel is waiting for event, or peripheral, 0 5'h1: DMA channel is waiting for event, or peripheral, 1 5'h2: DMA channel is waiting for event, or peripheral, 2 … 5'h1f: DMA channel is waiting for event, or peripheral, 31
3:0	RO	0x0	channel_status Channel 2 status. 4'h0: Stopped 4'h1: Executing 4'h2: Cache miss 4'h3: Updating PC 4'h4: Waiting for event 4'h5: At barrier 4'h7: Waiting for peripheral 4'h8: Killing 4'h9: Completing 4'he: Faulting completing 4'hf: Faulting Others: Reserved

DMAC_CPC2

Address: Operational Base + offset (0x0114)

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

DMAC_CSR3

Address: Operational Base + offset (0x0118)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RO	0x0	cns 1'b0: DMA channel operates in the Secure state 1'b1: DMA channel operates in the Non-secure state
20:16	RO	0x0	reserved
15	RO	0x0	dmawfp_periph 1'b0: DMAWFP executed with the periph operand not set 1'b1: DMAWFP executed with the periph operand set
14	RO	0x0	dmawfp_b_ns 1'b0: DMAWFP executed with the single operand set 1'b1: DMAWFP executed with the burst operand set
13:9	RO	0x0	reserved

Bit	Attr	Reset Value	Description
8:4	RO	0x00	wakeup_number Indicate the event or peripheral number that the channel is waiting for. 5'h0: DMA channel is waiting for event, or peripheral, 0 5'h1: DMA channel is waiting for event, or peripheral, 1 5'h2: DMA channel is waiting for event, or peripheral, 2 … 5'h1f: DMA channel is waiting for event, or peripheral, 31
3:0	RO	0x0	channel_status Channel 3 status. 4'h0: Stopped 4'h1: Executing 4'h2: Cache miss 4'h3: Updating PC 4'h4: Waiting for event 4'h5: At barrier 4'h7: Waiting for peripheral 4'h8: Killing 4'h9: Completing 4'he: Faulting completing 4'hf: Faulting Others: Reserved

DMAC_CPC3

Address: Operational Base + offset (0x011c)

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

DMAC_CSR4

Address: Operational Base + offset (0x0120)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RO	0x0	cns 1'b0: DMA channel operates in the Secure state 1'b1: DMA channel operates in the Non-secure state
20:16	RO	0x0	reserved
15	RO	0x0	dmawfp_periph 1'b0: DMAWFP executed with the periph operand not set 1'b1: DMAWFP executed with the periph operand set
14	RO	0x0	dmawfp_b_ns 1'b0: DMAWFP executed with the single operand set 1'b1: DMAWFP executed with the burst operand set
13:9	RO	0x0	reserved

Bit	Attr	Reset Value	Description
8:4	RO	0x00	wakeup_number Indicate the event or peripheral number that the channel is waiting for. 5'h0: DMA channel is waiting for event, or peripheral, 0 5'h1: DMA channel is waiting for event, or peripheral, 1 5'h2: DMA channel is waiting for event, or peripheral, 2 … 5'h1f: DMA channel is waiting for event, or peripheral, 31
3:0	RO	0x0	channel_status Channel 4 status. 4'h0: Stopped 4'h1: Executing 4'h2: Cache miss 4'h3: Updating PC 4'h4: Waiting for event 4'h5: At barrier 4'h7: Waiting for peripheral 4'h8: Killing 4'h9: Completing 4'he: Faulting completing 4'hf: Faulting Others: Reserved

DMAC_CPC4

Address: Operational Base + offset (0x0124)

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

DMAC_CSR5

Address: Operational Base + offset (0x0128)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RO	0x0	cns 1'b0: DMA channel operates in the Secure state 1'b1: DMA channel operates in the Non-secure state
20:16	RO	0x0	reserved
15	RO	0x0	dmawfp_periph 1'b0: DMAWFP executed with the periph operand not set 1'b1: DMAWFP executed with the periph operand set
14	RO	0x0	dmawfp_b_ns 1'b0: DMAWFP executed with the single operand set 1'b1: DMAWFP executed with the burst operand set
13:9	RO	0x0	reserved

Bit	Attr	Reset Value	Description
8:4	RO	0x00	wakeup_number Indicate the event or peripheral number that the channel is waiting for. 5'h0: DMA channel is waiting for event, or peripheral, 0 5'h1: DMA channel is waiting for event, or peripheral, 1 5'h2: DMA channel is waiting for event, or peripheral, 2 … 5'h1f: DMA channel is waiting for event, or peripheral, 31
3:0	RO	0x0	channel_status Channel 5 status. 4'h0: Stopped 4'h1: Executing 4'h2: Cache miss 4'h3: Updating PC 4'h4: Waiting for event 4'h5: At barrier 4'h7: Waiting for peripheral 4'h8: Killing 4'h9: Completing 4'he: Faulting completing 4'hf: Faulting Others: Reserved

DMAC_CPC5

Address: Operational Base + offset (0x012c)

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

DMAC_CSR6

Address: Operational Base + offset (0x0130)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RO	0x0	cns 1'b0: DMA channel operates in the Secure state 1'b1: DMA channel operates in the Non-secure state
20:16	RO	0x0	reserved
15	RO	0x0	dmawfp_periph 1'b0: DMAWFP executed with the periph operand not set 1'b1: DMAWFP executed with the periph operand set
14	RO	0x0	dmawfp_b_ns 1'b0: DMAWFP executed with the single operand set 1'b1: DMAWFP executed with the burst operand set
13:9	RO	0x0	reserved

Bit	Attr	Reset Value	Description
8:4	RO	0x00	wakeup_number Indicate the event or peripheral number that the channel is waiting for. 5'h0: DMA channel is waiting for event, or peripheral, 0 5'h1: DMA channel is waiting for event, or peripheral, 1 5'h2: DMA channel is waiting for event, or peripheral, 2 … 5'h1f: DMA channel is waiting for event, or peripheral, 31
3:0	RO	0x0	channel_status Channel 6 status. 4'h0: Stopped 4'h1: Executing 4'h2: Cache miss 4'h3: Updating PC 4'h4: Waiting for event 4'h5: At barrier 4'h7: Waiting for peripheral 4'h8: Killing 4'h9: Completing 4'he: Faulting completing 4'hf: Faulting Others: Reserved

DMAC_CPC6

Address: Operational Base + offset (0x0134)

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

DMAC_CSR7

Address: Operational Base + offset (0x0138)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RO	0x0	cns 1'b0: DMA channel operates in the Secure state 1'b1: DMA channel operates in the Non-secure state
20:16	RO	0x0	reserved
15	RO	0x0	dmawfp_periph 1'b0: DMAWFP executed with the periph operand not set 1'b1: DMAWFP executed with the periph operand set
14	RO	0x0	dmawfp_b_ns 1'b0: DMAWFP executed with the single operand set 1'b1: DMAWFP executed with the burst operand set
13:9	RO	0x0	reserved

Bit	Attr	Reset Value	Description
8:4	RO	0x00	wakeup_number Indicate the event or peripheral number that the channel is waiting for. 5'h0: DMA channel is waiting for event, or peripheral, 0 5'h1: DMA channel is waiting for event, or peripheral, 1 5'h2: DMA channel is waiting for event, or peripheral, 2 … 5'h1f: DMA channel is waiting for event, or peripheral, 31
3:0	RO	0x0	channel_status Channel 7 status. 4'h0: Stopped 4'h1: Executing 4'h2: Cache miss 4'h3: Updating PC 4'h4: Waiting for event 4'h5: At barrier 4'h7: Waiting for peripheral 4'h8: Killing 4'h9: Completing 4'he: Faulting completing 4'hf: Faulting Others: Reserved

DMAC_CPC7

Address: Operational Base + offset (0x013c)

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

DMAC_SAR0

Address: Operational Base + offset (0x0400)

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

DMAC_DAR0

Address: Operational Base + offset (0x0404)

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

DMAC_CCR0

Address: Operational Base + offset (0x0408)

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

Bit	Attr	Reset Value	Description
27:25	RO	0x0	dst_cache_ctrl Bit [27] 1'b0: AWCACHE[3] is LOW 1'b1: AWCACHE[3] is HIGH Bit [26] 1'b0: AWCACHE[1] is LOW 1'b1: AWCACHE[1] is HIGH Bit [25] 1'b0: AWCACHE[0] is LOW 1'b1: AWCACHE[0] is HIGH
24:22	RO	0x0	dst_prot_ctrl Bit [24] 1'b0: AWPROT[2] is LOW 1'b1: AWPROT[2] is HIGH Bit [23] 1'b0: AWPROT[1] is LOW 1'b1: AWPROT[1] is HIGH Bit [22] 1'b0: AWPROT[0] is LOW 1'b1: AWPROT[0] is HIGH
21:18	RO	0x0	dst_burst_len the destination data: 4'h0: 1 data transfer 4'h1: 2 data transfers 4'h2: 3 data transfers … 4'hf: 16 data transfers The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction is the product of dst_burst_len and dst_burst_size.
17:15	RO	0x0	dst_burst_size 3'h0: Writes 1 byte per beat 3'h1: Writes 2 bytes per beat 3'h2: Writes 4 bytes per beat 3'h3: Writes 8 bytes per beat 3'h4: Writes 16 bytes per beat Others: Reserved The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction is the product of dst_burst_len and dst_burst_size.
14	RO	0x0	dst_inc 1'b0: Fixed-address burst. The DMAC signals AWBURST[0] LOW. 1'b1: Incrementing-address burst. The DMAC signals AWBURST[0] HIGH.

Bit	Attr	Reset Value	Description
13:11	RO	0x0	src_cache_ctrl Bit [13] 1'b0: ARCACHE[2] is LOW 1'b1: ARCACHE[2] is HIGH Bit [12] 1'b0: ARCACHE[1] is LOW 1'b1: ARCACHE[1] is HIGH Bit [11] 1'b0: ARCACHE[0] is LOW 1'b1: ARCACHE[0] is HIGH
10:8	RO	0x0	src_prot_ctrl Bit [10] 1'b0: ARPROT[2] is LOW 1'b1: ARPROT[2] is HIGH Bit [9] 1'b0: ARPROT[1] is LOW 1'b1: ARPROT[1] is HIGH Bit [8] 1'b0: ARPROT[0] is LOW 1'b1: ARPROT[0] is HIGH
7:4	RO	0x0	src_burst_len 4'h0: 1 data transfer 4'h1: 2 data transfers 4'h2: 3 data transfers … 4'hf: 16 data transfers The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction is the product of src_burst_len and src_burst_size.
3:1	RO	0x0	src_burst_size 3'h0: Reads 1 byte per beat 3'h1: Reads 2 bytes per beat 3'h2: Reads 4 bytes per beat 3'h3: Reads 8 bytes per beat 3'h4: Reads 16 bytes per beat Others: Reserved The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction is the product of src_burst_len and src_burst_size.
0	RO	0x0	src_inc 1'b0: Fixed-address burst. The DMAC signals ARBURST[0] LOW. 1'b1: Incrementing-address burst. The DMAC signals ARBURST[0] HIGH.

DMAC_LC0_0

Address: Operational Base + offset (0x040c)

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

DMAC_LC1_0

Address: Operational Base + offset (0x0410)

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

DMAC_SAR1

Address: Operational Base + offset (0x0420)

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

DMAC_DAR1

Address: Operational Base + offset (0x0424)

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

DMAC_CCR1

Address: Operational Base + offset (0x0428)

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

Bit	Attr	Reset Value	Description
21:18	RO	0x0	dst_burst_len the destination data: 4'h0: 1 data transfer 4'h1: 2 data transfers 4'h2: 3 data transfers … 4'hf: 16 data transfers The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction is the product of dst_burst_len and dst_burst_size.
17:15	RO	0x0	dst_burst_size 3'h0: Writes 1 byte per beat 3'h1: Writes 2 bytes per beat 3'h2: Writes 4 bytes per beat 3'h3: Writes 8 bytes per beat 3'h4: Writes 16 bytes per beat Others: Reserved The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction is the product of dst_burst_len and dst_burst_size.
14	RO	0x0	dst_inc 1'b0: Fixed-address burst. The DMAC signals AWBURST[0] LOW. 1'b1: Incrementing-address burst. The DMAC signals AWBURST[0] HIGH.
13:11	RO	0x0	src_cache_ctrl Bit [13] 1'b0: ARCACHE[2] is LOW 1'b1: ARCACHE[2] is HIGH Bit [12] 1'b0: ARCACHE[1] is LOW 1'b1: ARCACHE[1] is HIGH Bit [11] 1'b0: ARCACHE[0] is LOW 1'b1: ARCACHE[0] is HIGH
10:8	RO	0x0	src_prot_ctrl Bit [10] 1'b0: ARPROT[2] is LOW 1'b1: ARPROT[2] is HIGH Bit [9] 1'b0: ARPROT[1] is LOW 1'b1: ARPROT[1] is HIGH Bit [8] 1'b0: ARPROT[0] is LOW 1'b1: ARPROT[0] is HIGH

Bit	Attr	Reset Value	Description
7:4	RO	0x0	src_burst_len 4'h0: 1 data transfer 4'h1: 2 data transfers 4'h2: 3 data transfers … 4'hf: 16 data transfers The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction is the product of src_burst_len and src_burst_size.
3:1	RO	0x0	src_burst_size 3'h0: Reads 1 byte per beat 3'h1: Reads 2 bytes per beat 3'h2: Reads 4 bytes per beat 3'h3: Reads 8 bytes per beat 3'h4: Reads 16 bytes per beat Others: Reserved The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction is the product of src_burst_len and src_burst_size.
0	RO	0x0	src_inc 1'b0: Fixed-address burst. The DMAC signals ARBURST[0] LOW. 1'b1: Incrementing-address burst. The DMAC signals ARBURST[0] HIGH.

DMAC_LC0_1

Address: Operational Base + offset (0x042c)

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

DMAC_LC1_1

Address: Operational Base + offset (0x0430)

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

DMAC_SAR2

Address: Operational Base + offset (0x0440)

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

DMAC_DAR2

Address: Operational Base + offset (0x0444)

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

DMAC_CCR2

Address: Operational Base + offset (0x0448)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:25	RO	0x0	dst_cache_ctrl Bit [27] 1'b0: AWCACHE[3] is LOW 1'b1: AWCACHE[3] is HIGH Bit [26] 1'b0: AWCACHE[1] is LOW 1'b1: AWCACHE[1] is HIGH Bit [25] 1'b0: AWCACHE[0] is LOW 1'b1: AWCACHE[0] is HIGH
24:22	RO	0x0	dst_prot_ctrl Bit [24] 1'b0: AWPROT[2] is LOW 1'b1: AWPROT[2] is HIGH Bit [23] 1'b0: AWPROT[1] is LOW 1'b1: AWPROT[1] is HIGH Bit [22] 1'b0: AWPROT[0] is LOW 1'b1: AWPROT[0] is HIGH
21:18	RO	0x0	dst_burst_len the destination data: 4'h0: 1 data transfer 4'h1: 2 data transfers 4'h2: 3 data transfers … 4'hf: 16 data transfers The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction is the product of dst_burst_len and dst_burst_size.

Bit	Attr	Reset Value	Description
17:15	RO	0x0	dst_burst_size 3'h0: Writes 1 byte per beat 3'h1: Writes 2 bytes per beat 3'h2: Writes 4 bytes per beat 3'h3: Writes 8 bytes per beat 3'h4: Writes 16 bytes per beat Others: Reserved The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction is the product of dst_burst_len and dst_burst_size.
14	RO	0x0	dst_inc 1'b0: Fixed-address burst. The DMAC signals AWBURST[0] LOW. 1'b1: Incrementing-address burst. The DMAC signals AWBURST[0] HIGH.
13:11	RO	0x0	src_cache_ctrl Bit [13] 1'b0: ARCACHE[2] is LOW 1'b1: ARCACHE[2] is HIGH Bit [12] 1'b0: ARCACHE[1] is LOW 1'b1: ARCACHE[1] is HIGH Bit [11] 1'b0: ARCACHE[0] is LOW 1'b1: ARCACHE[0] is HIGH
10:8	RO	0x0	src_prot_ctrl Bit [10] 1'b0: ARPROT[2] is LOW 1'b1: ARPROT[2] is HIGH Bit [9] 1'b0: ARPROT[1] is LOW 1'b1: ARPROT[1] is HIGH Bit [8] 1'b0: ARPROT[0] is LOW 1'b1: ARPROT[0] is HIGH
7:4	RO	0x0	src_burst_len 4'h0: 1 data transfer 4'h1: 2 data transfers 4'h2: 3 data transfers … 4'hf: 16 data transfers The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction is the product of src_burst_len and src_burst_size.

Bit	Attr	Reset Value	Description
3:1	RO	0x0	src_burst_size 3'h0: Reads 1 byte per beat 3'h1: Reads 2 bytes per beat 3'h2: Reads 4 bytes per beat 3'h3: Reads 8 bytes per beat 3'h4: Reads 16 bytes per beat Others: Reserved The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction is the product of src_burst_len and src_burst_size.
0	RO	0x0	src_inc 1'b0: Fixed-address burst. The DMAC signals ARBURST[0] LOW. 1'b1: Incrementing-address burst. The DMAC signals ARBURST[0] HIGH.

DMAC_LC0_2

Address: Operational Base + offset (0x044c)

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

DMAC_LC1_2

Address: Operational Base + offset (0x0450)

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

DMAC_SAR3

Address: Operational Base + offset (0x0460)

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

DMAC_DAR3

Address: Operational Base + offset (0x0464)

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

DMAC_CCR3

Address: Operational Base + offset (0x0468)

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

Bit	Attr	Reset Value	Description
27:25	RO	0x0	dst_cache_ctrl Bit [27] 1'b0: AWCACHE[3] is LOW 1'b1: AWCACHE[3] is HIGH Bit [26] 1'b0: AWCACHE[1] is LOW 1'b1: AWCACHE[1] is HIGH Bit [25] 1'b0: AWCACHE[0] is LOW 1'b1: AWCACHE[0] is HIGH
24:22	RO	0x0	dst_prot_ctrl Bit [24] 1'b0: AWPROT[2] is LOW 1'b1: AWPROT[2] is HIGH Bit [23] 1'b0: AWPROT[1] is LOW 1'b1: AWPROT[1] is HIGH Bit [22] 1'b0: AWPROT[0] is LOW 1'b1: AWPROT[0] is HIGH
21:18	RO	0x0	dst_burst_len the destination data: 4'h0: 1 data transfer 4'h1: 2 data transfers 4'h2: 3 data transfers … 4'hf: 16 data transfers The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction is the product of dst_burst_len and dst_burst_size.
17:15	RO	0x0	dst_burst_size 3'h0: Writes 1 byte per beat 3'h1: Writes 2 bytes per beat 3'h2: Writes 4 bytes per beat 3'h3: Writes 8 bytes per beat 3'h4: Writes 16 bytes per beat Others: Reserved The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction is the product of dst_burst_len and dst_burst_size.
14	RO	0x0	dst_inc 1'b0: Fixed-address burst. The DMAC signals AWBURST[0] LOW. 1'b1: Incrementing-address burst. The DMAC signals AWBURST[0] HIGH.

Bit	Attr	Reset Value	Description
13:11	RO	0x0	src_cache_ctrl Bit [13] 1'b0: ARCACHE[2] is LOW 1'b1: ARCACHE[2] is HIGH Bit [12] 1'b0: ARCACHE[1] is LOW 1'b1: ARCACHE[1] is HIGH Bit [11] 1'b0: ARCACHE[0] is LOW 1'b1: ARCACHE[0] is HIGH
10:8	RO	0x0	src_prot_ctrl Bit [10] 1'b0: ARPROT[2] is LOW 1'b1: ARPROT[2] is HIGH Bit [9] 1'b0: ARPROT[1] is LOW 1'b1: ARPROT[1] is HIGH Bit [8] 1'b0: ARPROT[0] is LOW 1'b1: ARPROT[0] is HIGH
7:4	RO	0x0	src_burst_len 4'h0: 1 data transfer 4'h1: 2 data transfers 4'h2: 3 data transfers … 4'hf: 16 data transfers The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction is the product of src_burst_len and src_burst_size.
3:1	RO	0x0	src_burst_size 3'h0: Reads 1 byte per beat 3'h1: Reads 2 bytes per beat 3'h2: Reads 4 bytes per beat 3'h3: Reads 8 bytes per beat 3'h4: Reads 16 bytes per beat Others: Reserved The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction is the product of src_burst_len and src_burst_size.
0	RO	0x0	src_inc 1'b0: Fixed-address burst. The DMAC signals ARBURST[0] LOW. 1'b1: Incrementing-address burst. The DMAC signals ARBURST[0] HIGH.

DMAC_LC0_3

Address: Operational Base + offset (0x046c)

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

DMAC_LC1_3

Address: Operational Base + offset (0x0470)

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

DMAC_SAR4

Address: Operational Base + offset (0x0480)

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

DMAC_DAR4

Address: Operational Base + offset (0x0484)

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

DMAC_CCR4

Address: Operational Base + offset (0x0488)

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

Bit	Attr	Reset Value	Description
21:18	RO	0x0	dst_burst_len the destination data: 4'h0: 1 data transfer 4'h1: 2 data transfers 4'h2: 3 data transfers … 4'hf: 16 data transfers The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction is the product of dst_burst_len and dst_burst_size.
17:15	RO	0x0	dst_burst_size 3'h0: Writes 1 byte per beat 3'h1: Writes 2 bytes per beat 3'h2: Writes 4 bytes per beat 3'h3: Writes 8 bytes per beat 3'h4: Writes 16 bytes per beat Others: Reserved The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction is the product of dst_burst_len and dst_burst_size.
14	RO	0x0	dst_inc 1'b0: Fixed-address burst. The DMAC signals AWBURST[0] LOW. 1'b1: Incrementing-address burst. The DMAC signals AWBURST[0] HIGH.
13:11	RO	0x0	src_cache_ctrl Bit [13] 1'b0: ARCACHE[2] is LOW 1'b1: ARCACHE[2] is HIGH Bit [12] 1'b0: ARCACHE[1] is LOW 1'b1: ARCACHE[1] is HIGH Bit [11] 1'b0: ARCACHE[0] is LOW 1'b1: ARCACHE[0] is HIGH
10:8	RO	0x0	src_prot_ctrl Bit [10] 1'b0: ARPROT[2] is LOW 1'b1: ARPROT[2] is HIGH Bit [9] 1'b0: ARPROT[1] is LOW 1'b1: ARPROT[1] is HIGH Bit [8] 1'b0: ARPROT[0] is LOW 1'b1: ARPROT[0] is HIGH

Bit	Attr	Reset Value	Description
7:4	RO	0x0	src_burst_len 4'h0: 1 data transfer 4'h1: 2 data transfers 4'h2: 3 data transfers … 4'hf: 16 data transfers The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction is the product of src_burst_len and src_burst_size.
3:1	RO	0x0	src_burst_size 3'h0: Reads 1 byte per beat 3'h1: Reads 2 bytes per beat 3'h2: Reads 4 bytes per beat 3'h3: Reads 8 bytes per beat 3'h4: Reads 16 bytes per beat Others: Reserved The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction is the product of src_burst_len and src_burst_size.
0	RO	0x0	src_inc 1'b0: Fixed-address burst. The DMAC signals ARBURST[0] LOW. 1'b1: Incrementing-address burst. The DMAC signals ARBURST[0] HIGH.

DMAC_LC0_4

Address: Operational Base + offset (0x048c)

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

DMAC_LC1_4

Address: Operational Base + offset (0x0490)

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

DMAC_SAR5

Address: Operational Base + offset (0x04a0)

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

DMAC_DAR5

Address: Operational Base + offset (0x04a4)

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

DMAC_CCR5

Address: Operational Base + offset (0x04a8)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:25	RO	0x0	dst_cache_ctrl Bit [27] 1'b0: AWCACHE[3] is LOW 1'b1: AWCACHE[3] is HIGH Bit [26] 1'b0: AWCACHE[1] is LOW 1'b1: AWCACHE[1] is HIGH Bit [25] 1'b0: AWCACHE[0] is LOW 1'b1: AWCACHE[0] is HIGH
24:22	RO	0x0	dst_prot_ctrl Bit [24] 1'b0: AWPROT[2] is LOW 1'b1: AWPROT[2] is HIGH Bit [23] 1'b0: AWPROT[1] is LOW 1'b1: AWPROT[1] is HIGH Bit [22] 1'b0: AWPROT[0] is LOW 1'b1: AWPROT[0] is HIGH
21:18	RO	0x0	dst_burst_len the destination data: 4'h0: 1 data transfer 4'h1: 2 data transfers 4'h2: 3 data transfers … 4'hf: 16 data transfers The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction is the product of dst_burst_len and dst_burst_size.

Bit	Attr	Reset Value	Description
17:15	RO	0x0	dst_burst_size 3'h0: Writes 1 byte per beat 3'h1: Writes 2 bytes per beat 3'h2: Writes 4 bytes per beat 3'h3: Writes 8 bytes per beat 3'h4: Writes 16 bytes per beat Others: Reserved The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction is the product of dst_burst_len and dst_burst_size.
14	RO	0x0	dst_inc 1'b0: Fixed-address burst. The DMAC signals AWBURST[0] LOW. 1'b1: Incrementing-address burst. The DMAC signals AWBURST[0] HIGH.
13:11	RO	0x0	src_cache_ctrl Bit [13] 1'b0: ARCACHE[2] is LOW 1'b1: ARCACHE[2] is HIGH Bit [12] 1'b0: ARCACHE[1] is LOW 1'b1: ARCACHE[1] is HIGH Bit [11] 1'b0: ARCACHE[0] is LOW 1'b1: ARCACHE[0] is HIGH
10:8	RO	0x0	src_prot_ctrl Bit [10] 1'b0: ARPROT[2] is LOW 1'b1: ARPROT[2] is HIGH Bit [9] 1'b0: ARPROT[1] is LOW 1'b1: ARPROT[1] is HIGH Bit [8] 1'b0: ARPROT[0] is LOW 1'b1: ARPROT[0] is HIGH
7:4	RO	0x0	src_burst_len 4'h0: 1 data transfer 4'h1: 2 data transfers 4'h2: 3 data transfers … 4'hf: 16 data transfers The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction is the product of src_burst_len and src_burst_size.

Bit	Attr	Reset Value	Description
3:1	RO	0x0	src_burst_size 3'h0: Reads 1 byte per beat 3'h1: Reads 2 bytes per beat 3'h2: Reads 4 bytes per beat 3'h3: Reads 8 bytes per beat 3'h4: Reads 16 bytes per beat Others: Reserved The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction is the product of src_burst_len and src_burst_size.
0	RO	0x0	src_inc 1'b0: Fixed-address burst. The DMAC signals ARBURST[0] LOW. 1'b1: Incrementing-address burst. The DMAC signals ARBURST[0] HIGH.

DMAC_LC0_5

Address: Operational Base + offset (0x04ac)

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

DMAC_LC1_5

Address: Operational Base + offset (0x04b0)

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

DMAC_SAR6

Address: Operational Base + offset (0x04c0)

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

DMAC_DAR6

Address: Operational Base + offset (0x04c4)

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

DMAC_CCR6

Address: Operational Base + offset (0x04c8)

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

Bit	Attr	Reset Value	Description
27:25	RO	0x0	dst_cache_ctrl Bit [27] 1'b0: AWCACHE[3] is LOW 1'b1: AWCACHE[3] is HIGH Bit [26] 1'b0: AWCACHE[1] is LOW 1'b1: AWCACHE[1] is HIGH Bit [25] 1'b0: AWCACHE[0] is LOW 1'b1: AWCACHE[0] is HIGH
24:22	RO	0x0	dst_prot_ctrl Bit [24] 1'b0: AWPROT[2] is LOW 1'b1: AWPROT[2] is HIGH Bit [23] 1'b0: AWPROT[1] is LOW 1'b1: AWPROT[1] is HIGH Bit [22] 1'b0: AWPROT[0] is LOW 1'b1: AWPROT[0] is HIGH
21:18	RO	0x0	dst_burst_len the destination data: 4'h0: 1 data transfer 4'h1: 2 data transfers 4'h2: 3 data transfers … 4'hf: 16 data transfers The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction is the product of dst_burst_len and dst_burst_size.
17:15	RO	0x0	dst_burst_size 3'h0: Writes 1 byte per beat 3'h1: Writes 2 bytes per beat 3'h2: Writes 4 bytes per beat 3'h3: Writes 8 bytes per beat 3'h4: Writes 16 bytes per beat Others: Reserved The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction is the product of dst_burst_len and dst_burst_size.
14	RO	0x0	dst_inc 1'b0: Fixed-address burst. The DMAC signals AWBURST[0] LOW. 1'b1: Incrementing-address burst. The DMAC signals AWBURST[0] HIGH.

Bit	Attr	Reset Value	Description
13:11	RO	0x0	src_cache_ctrl Bit [13] 1'b0: ARCACHE[2] is LOW 1'b1: ARCACHE[2] is HIGH Bit [12] 1'b0: ARCACHE[1] is LOW 1'b1: ARCACHE[1] is HIGH Bit [11] 1'b0: ARCACHE[0] is LOW 1'b1: ARCACHE[0] is HIGH
10:8	RO	0x0	src_prot_ctrl Bit [10] 1'b0: ARPROT[2] is LOW 1'b1: ARPROT[2] is HIGH Bit [9] 1'b0: ARPROT[1] is LOW 1'b1: ARPROT[1] is HIGH Bit [8] 1'b0: ARPROT[0] is LOW 1'b1: ARPROT[0] is HIGH
7:4	RO	0x0	src_burst_len 4'h0: 1 data transfer 4'h1: 2 data transfers 4'h2: 3 data transfers … 4'hf: 16 data transfers The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction is the product of src_burst_len and src_burst_size.
3:1	RO	0x0	src_burst_size 3'h0: Reads 1 byte per beat 3'h1: Reads 2 bytes per beat 3'h2: Reads 4 bytes per beat 3'h3: Reads 8 bytes per beat 3'h4: Reads 16 bytes per beat Others: Reserved The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction is the product of src_burst_len and src_burst_size.
0	RO	0x0	src_inc 1'b0: Fixed-address burst. The DMAC signals ARBURST[0] LOW. 1'b1: Incrementing-address burst. The DMAC signals ARBURST[0] HIGH.

DMAC_LC0_6

Address: Operational Base + offset (0x04cc)

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

DMAC_LC1_6

Address: Operational Base + offset (0x04d0)

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

DMAC_SAR7

Address: Operational Base + offset (0x04e0)

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

DMAC_DAR7

Address: Operational Base + offset (0x04e4)

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

DMAC_CCR7

Address: Operational Base + offset (0x04e8)

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

Bit	Attr	Reset Value	Description
21:18	RO	0x0	dst_burst_len the destination data: 4'h0: 1 data transfer 4'h1: 2 data transfers 4'h2: 3 data transfers … 4'hf: 16 data transfers The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction is the product of dst_burst_len and dst_burst_size.
17:15	RO	0x0	dst_burst_size 3'h0: Writes 1 byte per beat 3'h1: Writes 2 bytes per beat 3'h2: Writes 4 bytes per beat 3'h3: Writes 8 bytes per beat 3'h4: Writes 16 bytes per beat Others: Reserved The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction is the product of dst_burst_len and dst_burst_size.
14	RO	0x0	dst_inc 1'b0: Fixed-address burst. The DMAC signals AWBURST[0] LOW. 1'b1: Incrementing-address burst. The DMAC signals AWBURST[0] HIGH.
13:11	RO	0x0	src_cache_ctrl Bit [13] 1'b0: ARCACHE[2] is LOW 1'b1: ARCACHE[2] is HIGH Bit [12] 1'b0: ARCACHE[1] is LOW 1'b1: ARCACHE[1] is HIGH Bit [11] 1'b0: ARCACHE[0] is LOW 1'b1: ARCACHE[0] is HIGH
10:8	RO	0x0	src_prot_ctrl Bit [10] 1'b0: ARPROT[2] is LOW 1'b1: ARPROT[2] is HIGH Bit [9] 1'b0: ARPROT[1] is LOW 1'b1: ARPROT[1] is HIGH Bit [8] 1'b0: ARPROT[0] is LOW 1'b1: ARPROT[0] is HIGH

Bit	Attr	Reset Value	Description
7:4	RO	0x0	src_burst_len 4'h0: 1 data transfer 4'h1: 2 data transfers 4'h2: 3 data transfers … 4'hf: 16 data transfers The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction is the product of src_burst_len and src_burst_size.
3:1	RO	0x0	src_burst_size 3'h0: Reads 1 byte per beat 3'h1: Reads 2 bytes per beat 3'h2: Reads 4 bytes per beat 3'h3: Reads 8 bytes per beat 3'h4: Reads 16 bytes per beat Others: Reserved The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction is the product of src_burst_len and src_burst_size.
0	RO	0x0	src_inc 1'b0: Fixed-address burst. The DMAC signals ARBURST[0] LOW. 1'b1: Incrementing-address burst. The DMAC signals ARBURST[0] HIGH.

DMAC_LC0_7

Address: Operational Base + offset (0x04ec)

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

DMAC_LC1_7

Address: Operational Base + offset (0x04f0)

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

DMAC_DBGSTATUS

Address: Operational Base + offset (0x0d00)

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

DMAC_DBGCMD

Address: Operational Base + offset (0x0d04)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1:0	WO	0x0	dbgcmd 2'b00: Execute the instruction that the DMAC_DBGINST [1:0] Registers contain Others: Reserved

DMAC_DBGINST0

Address: Operational Base + offset (0x0d08)

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

DMAC_DBGINST1

Address: Operational Base + offset (0x0d0c)

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

DMAC_CR0

Address: Operational Base + offset (0x0e00)

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

Bit	Attr	Reset Value	Description
21:17	RO	0x0f	num_events 5'h0: 1 interrupt output, irq[0] 5'h1: 2 interrupt outputs, irq[1:0] 5'h2: 3 interrupt outputs, irq[2:0] … 5'h1f: 32 interrupt outputs, irq[31:0]
16:12	RO	0x13	num_periph_req 5'h0: 1 peripheral request interface 5'h1: 2 peripheral request interfaces 5'h2: 3 peripheral request interfaces … 5'h1f: 32 peripheral request interfaces
11:7	RO	0x0	reserved
6:4	RO	0x7	num_chnls 3'b000: 1 DMA channel 3'b001: 2 DMA channels 3'b010: 3 DMA channels … 3'b111: 8 DMA channels
3	RO	0x0	reserved
2	RO	0x1	mgr_ns_at_rst 1'b0: boot_manager_ns was LOW 1'b1: boot_manager_ns was HIGH
1	RO	0x0	boot_en 1'b0: boot_from_pc was LOW 1'b1: boot_from_pc was HIGH
0	RO	0x1	periph_req 1'b0: The DMAC does not provide a peripheral request interface 1'b1: The DMAC provides the number of peripheral request interfaces that the num_periph_req field specifies

DMAC_CR1

Address: Operational Base + offset (0x0e04)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RO	0xb	num_i_cache_lines 4'b0000: 1 i-cache line 4'b0001: 2 i-cache lines 4'b0010: 3 i-cache lines … 4'b1111: 16 i-cache lines
3	RO	0x0	reserved

Bit	Attr	Reset Value	Description
2:0	RO	0x5	i_cache_len 3'b010: 4 bytes 3'b011: 8 bytes 3'b100: 16 bytes 3'b101: 32 bytes Others: Reserved

DMAC_CR2

Address: Operational Base + offset (0x0e08)

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

DMAC_CR3

Address: Operational Base + offset (0x0e0c)

Bit	Attr	Reset Value	Description
31:0	RO	0x0000ffff	ins Bit [N] 1'b0: Assigns event<N> or irq[N] to the Secure state 1'b1: Assigns event<N> or irq[N] to the Non-secure state

DMAC_CR4

Address: Operational Base + offset (0x0e10)

Bit	Attr	Reset Value	Description
31:0	RO	0x000fffff	pns Bit [N] 1'b0: Assigns peripheral request interface N to the Secure state 1'b1: Assigns peripheral request interface N to the Non-secure state

DMAC_CRDn

Address: Operational Base + offset (0x0e14)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:20	RO	0x07f	data_buffer_dep 10'b000000000: 1 line 10'b000000001: 2 lines … 10'b111111111: 1024 lines
19:16	RO	0xf	rd_q_dep 4'b0000: 1 line 4'b0001: 2 lines … 4'b1111: 16 lines
15	RO	0x0	reserved

Bit	Attr	Reset Value	Description
14:12	RO	0x7	rd_cap 3'b000: 1 3'b001: 2 … 3'b111: 8
11:8	RO	0xf	wr_q_dep 4'b0000: 1 line 4'b0001: 2 lines … 4'b1111: 16 lines
7	RO	0x0	reserved
6:4	RO	0x7	wr_cap 3'b000: 1 3'b001: 2 … 3'b111: 8
3	RO	0x0	reserved
2:0	RO	0x3	data_width 3'b010: 32-bit 3'b011: 64-bit 3'b100: 128-bit Others:Reserved

DMAC_WD

Address: Operational Base + offset (0x0e80)

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

16.5 Timing Diagram

Following picture shows the relationship between dma_req and dma_ack.

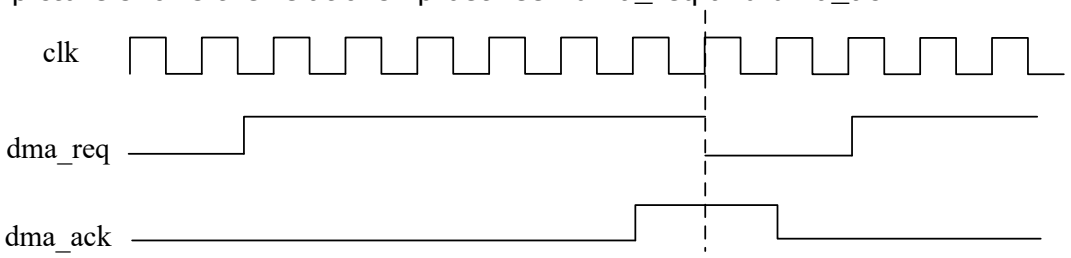


Fig. 16-3 DMAC Request and Acknowledge Timing

16.6 Interface Description

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

Table 16-2 DMAC Boot Interface

Interface	Reset Value	Control Source
boot_manager_ns	0x1	SGRF_DMAC_CON3[0]
boot_irq_ns	0xFFFF	SGRF_DMAC_CON0[15:0]
boot_periph_ns	0xFFFFFFFF	{SGRF_DMAC_CON2[15:0],SGRF_DMAC_CON1[15:0]}
grf_drtype_spi0_rx	0x1	SGRF_DMAC_CON4[1:0]
grf_drtype_spi0_tx	0x1	SGRF_DMAC_CON4[3:2]
grf_drtype_spi1_rx	0x1	SGRF_DMAC_CON4[5:4]
grf_drtype_spi1_tx	0x1	SGRF_DMAC_CON4[7:6]
grf_drtype_uart0_rx	0x1	SGRF_DMAC_CON4[9:8]
grf_drtype_uart0_tx	0x1	SGRF_DMAC_CON4[11:10]
grf_drtype_uart1_rx	0x1	SGRF_DMAC_CON4[13:12]
grf_drtype_uart1_tx	0x1	SGRF_DMAC_CON4[15:14]
grf_drtype_uart2_rx	0x1	SGRF_DMAC_CON5[1:0]
grf_drtype_uart2_tx	0x1	SGRF_DMAC_CON5[3:2]
grf_drtype_uart3_rx	0x1	SGRF_DMAC_CON5[5:4]
grf_drtype_uart3_tx	0x1	SGRF_DMAC_CON5[7:6]
grf_drtype_uart4_rx	0x1	SGRF_DMAC_CON5[9:8]
grf_drtype_uart4_tx	0x1	SGRF_DMAC_CON5[11:10]
grf_drtype_uart5_rx	0x1	SGRF_DMAC_CON5[13:12]
grf_drtype_uart5_tx	0x1	SGRF_DMAC_CON5[15:14]
grf_drtype_pwm0	0x1	SGRF_DMAC_CON6[1:0]
grf_drtype_pwm1	0x1	SGRF_DMAC_CON6[3:2]
grf_drtype_pwm2	0x1	SGRF_DMAC_CON6[5:4]
grf_drtype_i2s0_rx	0x1	SGRF_DMAC_CON6[7:6]
grf_drtype_i2s0_tx	0x1	SGRF_DMAC_CON6[9:8]
grf_drtype_i2s1_rx	0x1	SGRF_DMAC_CON6[11:10]
grf_drtype_i2s1_tx	0x1	SGRF_DMAC_CON6[13:12]
grf_drtype_i2s2_rx	0x1	SGRF_DMAC_CON6[15:14]
grf_drtype_i2s2_tx	0x1	SGRF_DMAC_CON7[1:0]
grf_drtype_pdm	0x1	SGRF_DMAC_CON7[3:2]
grf_drtype_audiopwm	0x1	SGRF_DMAC_CON7[5:4]

boot_manager_ns
When the DMAC exits from reset, this signal controls the security state of the DMA manager thread:
0 = assigns DMA manager to the Secure state
1 = assigns DMA manager to the Non-secure state.
boot_irq_ns
Controls the security state of an event-interrupt resource, when the DMAC exits from reset:
boot_irq_ns[x] is LOW
The DMAC assigns event<x> or irq[x] to the Secure state.
boot_irq_ns[x] is HIGH
The DMAC assigns event<x> or irq[x] to the Non-secure state.
boot_periph_ns
Controls the security state of a peripheral request interface, when the DMAC exits from reset:
boot_periph_ns[x] is LOW
The DMAC assigns peripheral request interface x to the Secure state.

boot_periph_ns[x] is HIGH
The DMAC assigns peripheral request interface x to the Non-secure state.
grf_drtype_<x>
The DMAC sets the state of the request_type flag:
grf_drtype_<x>[1:0]=b00: request_type<x> = Single.
grf_drtype_<x>[1:0]=b01: request_type<x> = Burst.

16.7 Application Notes

16.7.1 Using the APB Slave Interfaces

You must ensure that you use the appropriate APB interface, depending on the security state in which the boot_manager_ns initializes the DMAC to operate. For example, if the DMAC is in the secure state, you must issue the instruction using the secure APB interface, otherwise the DMAC ignores the instruction. You can use the secure APB interface, or the non-secure APB interface, to start or restart a DMA channel when the DMAC is in the Non-secure state.
The necessary steps to start a DMA channel thread using the debug instruction registers as following:
1. Create a program for the DMA channel.
2. Store the program in a region of system memory.
3. Poll the DMAC_DBGSTATUS Register to ensure that debug is idle, that is, the dbgstatus bit is 0.
4. Write to the DMAC_DBGINST0 Register and enter the:
- Instruction byte 0 encoding for DMAGO.
- Instruction byte 1 encoding for DMAGO.
- Debug thread bit to 0. This selects the DMA manager thread.

5. Write to the DBGINST1 Register with the DMAGO instruction byte [5:2] data, see Debug Instruction-1 Register o. You must set these four bytes to the address of the first instruction in the program that was written to system memory in step 2.
6. Writing zero to the DMAC_DBGCMD Register. The DMAC starts the DMA channel thread and sets the dbgstatus bit to 1.

16.7.2 Security Usage

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

Register.
4. Moves the DMA manager to the Faulting state.
ns = 1
The DMAC starts a DMA channel thread in the Non-secure state and programs the CNS bit to be non-secure.

DMAWFE

The DMAC uses the status of the corresponding INS bit, in the DMAC_CR3 Register, to control if it waits for the event. If:
INS = 0
The event is in the Secure state. The DMAC:
1. Executes a NOP.
2. Sets the DMAC_FSRD Register, see Fault Status DMA Manager Register.
3. Sets the mgr_evnt_err bit in the DMAC_FTRD Register, see Fault Type DMA Manager Register.
4. Moves the DMA manager to the Faulting state.
INS = 1
The event is in the Non-secure state. The DMAC halts execution of the thread and waits for the event to occur.

DMASEV

The DMAC uses the status of the corresponding INS bit, in the CR3Register, to control if it creates the event-interrupt. If:
INS = 0
The event-interrupt resource is in the secure state. The DMAC:
1. Executes a NOP.
2. Sets the DMAC_FSRD Register, see Fault Status DMA Manager Register.
3. Sets the mgr_evnt_err bit in the DMAC_FTRD Register, see Fault Type DMA Manager Register.
4. Moves the DMA manager to the Faulting state.
INS = 1
The event-interrupt resource is in the Non-secure state. The DMAC creates the event-interrupt.

DMA channel thread is in the secure state

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

DMAWFE

The DMAC halts execution of the thread until the event occurs. When the event occurs, the DMAC continues execution of the thread, irrespective of the security state of the corresponding INS bit, in the DMAC_CR3 Register.

DMASEV

The DMAC creates the event-interrupt, irrespective of the security state of the corresponding INS bit, in the DMAC_CR3 Register.

DMAWFP

The DMAC halts execution of the thread until the peripheral signals a DMA request. When this occurs, the DMAC continues execution of the thread, irrespective of the security state of the corresponding PNS bit, in the DMAC_CR4 Register.

DMALDP, DMASTP

The DMAC sends a message to the peripheral to communicate that data transfer is complete, irrespective of the security state of the corresponding PNS bit, in the DMAC_CR4 Register.

DMAFLUSHP

The DMAC clears the state of the peripheral and sends a message to the peripheral to resend its level status, irrespective of the security state of the corresponding PNS bit, in the DMAC_CR4 Register.
When a DMA channel thread is in the Secure state, it enables the DMAC to perform secure and non-secure AXI accesses.

DMA channel thread is in the Non-secure state

When the CNS bit is 1, the DMA channel thread is programmed to operate in the Non-secure state and it only performs non-secure instruction fetches.
When a DMA channel thread in the Non-secure state processes the following instructions:
DMAWFE
The DMAC uses the status of the corresponding INS bit, in the DMAC_CR3 Register, to control if it waits for the event. If:
INS = 0
The event is in the Secure state. The DMAC:
1. Executes a NOP.
2. Sets the appropriate bit in the DMAC_FSRC Register that corresponds to the DMA channel number. See Fault Status DMA Channel Register.
3. Sets the ch_evnt_err bit in the DMAC_FTRn(n=0~7) Register, see Fault Type DMA Channel Registers.
4. Moves the DMA channel to the Faulting completing state.
INS = 1
The event is in the Non-secure state. The DMAC halts execution of the thread and waits for the event to occur.
DMASEV
The DMAC uses the status of the corresponding INS bit, in the DMAC_CR3 Register, to control if it creates the event. If:
INS = 0
The event-interrupt resource is in the Secure state. The DMAC:
1. Executes a NOP.
2. Sets the appropriate bit in the DMAC_FSRC Register that corresponds to the DMA channel number. See Fault Status DMA Channel Register.
3. Sets the ch_evnt_err bit in the DMAC_FTRn(n=0~7) Register, see Fault Type DMA Channel Registers.
4. Moves the DMA channel to the Faulting completing state.
INS = 1
The event-interrupt resource is in the Non-secure state. The DMAC creates the event-interrupt.
DMAWFP
The DMAC uses the status of the corresponding PNS bit, in the DMAC_CR4 Register, to control if it waits for the peripheral to signal a request. If:
PNS = 0
The peripheral is in the Secure state. The DMAC:
1. Executes a NOP.
2. Sets the appropriate bit in the DMAC_FSRC Register that corresponds to the DMA channel number. See Fault Status DMA Channel Register.
3. Sets the ch_periph_err bit in the DMAC_FTRn(n=0~7) Register, see Fault Type DMA Channel Registers.
4. Moves the DMA channel to the Faulting completing state.
PNS = 1
The peripheral is in the Non-secure state. The DMAC halts execution of the thread and waits for the peripheral to signal a request.
DMALDP, DMASTP
The DMAC uses the status of the corresponding PNS bit, in the DMAC_CR4 Register, to control if it sends an acknowledgement to the peripheral. If:
PNS = 0
The peripheral is in the secure state. The DMAC:
1. Executes a NOP.
2. Sets the appropriate bit in the DMAC_FSRC Register that corresponds to the DMA channel number. See Fault Status DMA Channel Register.
3. Sets the ch_periph_err bit in the DMAC_FTRn(n=0~7) Register, see Fault Type DMA Channel Registers.
4. Moves the DMA channel to the Faulting completing state.
PNS = 1

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

DMAFLUSHP

The DMAC uses the status of the corresponding PNS bit, in the DMAC_CR4 Register, to control if it sends a flush request to the peripheral. If:

PNS = 0

The peripheral is in the secure state. The DMAC:

1. Executes a NOP.
2. Sets the appropriate bit in the DMAC_FSRC Register that corresponds to the DMA channel number. See Fault Status DMA Channel Register.
3. Sets the ch_periph_err bit in the DMAC_FTRn(n=0~7) Register, see Fault Type DMA Channel Registers.
4. Moves the DMA channel to the Faulting completing state.

PNS = 1

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

When a DMA channel thread is in the Non-secure state, and a DMAMOV CCR instruction attempts to program the channel to perform a secure AXI transaction, the DMAC:

1. Executes a DMANOP.
2. Sets the appropriate bit in the DMAC_FSRC Register that corresponds to the DMA channel number. See Fault Status DMA Channel Register.
3. Sets the ch_rdwr_err bit in the DMAC_FTRn(n=0~7) Register, see Fault Type DMA Channel Registers.
4. Moves the DMA channel thread to the Faulting completing state.

16.7.3 Programming Restrictions

Fixed unaligned bursts

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

Unaligned read

- src_inc field is 0 in the DMAC_CCRn(n=0~7) Register
- the DMAC_SARn(n=0~7) Register contains an address that is not aligned to the size of data that the src_burst_size field contain

Unaligned write

- dst_inc field is 0 in the DMAC_CCRn(n=0~7) Register
- the DMAC_DARn(n=0~7) Register contains an address that is not aligned to the size of data that the dst_burst_size field contains

Endian swap size restrictions

If you program the endian_swap_size field in the DMAC_CCRn(n=0~7) Register, to enable a DMA channel to perform an endian swap then you must set the corresponding DMAC_SARn(n=0~7) Register and the corresponding DMAC_DARn(n=0~7) Register to contain an address that is aligned to the value that the endian_swap_size field contains.

Updating DMA channel control registers during a DMA cycle restrictions

Prior to the DMAC executing a sequence of DMALD and DMAST instructions, the values you program in to the DMAC_CCRn(n=0~7) Register, DMAC_SARn(n=0~7) Register, and DMAC_DARn(n=0~7) Register control the data byte lane manipulation that the DMAC performs when it transfers the data from the source address to the destination address. You'd better not update these registers during a DMA cycle.

Resource sharing between DMA channels

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

16.7.4 Unaligned Transfers

For a configuration with more than one channel, if any of channels 1 to 7 is performing transfers between certain types of misaligned source and destination addresses, then the

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

Splitting data

Depending upon the programmed values for the DMA transfer, one beat of read data from the AXI interface need to be split across two lines in the internal data buffer. This occurs when the read data beat contains data bytes which will be written to addresses that wrap around at the AXI interface data width, so that these bytes could not be transferred by a single AXI write data beat of the full interface width.
Most applications of DMA do not split data in this way, so are NOT vulnerable to data corruption from this defect.
The following cases are NOT vulnerable to data corruption because they do not split data:

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

Table 16-3 Source Size in DMAC_CCRn

Source size in DMAC_CCRn	Allowed offset between DMAC_SARn and DMAC_DARn
SS8	any offset allowed.
SS16	0,2,4,6,8,10,12,14
SS32	0,4,8,12
SS64	0,8

16.7.5 Interrupt Sharing between Channels

As the DMAC does not record which channel (or list of channels) have asserted an interrupt. So it will depend on your program and whether any of the visible information for that program can be used to determine progress, and help identify the interrupt source.
There are 4 likely information sources that can be used to determine the progress made by a program:
- Program counter (PC)
- Source address
- Destination address
- Loop counters (LC)

For example, a program might emit an interrupt each time that it iterates around a loop. In this case, the interrupt service routine (ISR) would need to store the loop value of each channel when it is called, and then compare against the new value when it is next called. A change in value would indicate that the program has progressed.
The ISR must be carefully written to ensure that no interrupts are lost. The sequence of operations is as follows:
1. Disable interrupts
2. Immediately clear the interrupt in DMA-330
3. Check the relevant registers for both channels to determine which must be serviced
4. Take appropriate action for the channels
5. Re-enable interrupts and exit ISR

16.7.6 Instruction Sets

Table 16-4 DMAC Instruction Sets

Mnemonic	Instruction	Thread Usage
DMAADDH	Add Halfword	C
DMAEND	End	M/C

DMAFLUSHP	Flush and notify Peripheral	C
DMAGO	Go	M
DMAKILL	Kill	C
DMALD	Load	C
DMALDP	Load Peripheral	C
DMALP	Loop	C
DMALPEND	Loop End	C
DMALPFE	Loop Forever	C
DMAMOV	Move	C
DMANOP	No operation	M/C
DMARMB	Read Memory Barrier	C
DMASEV	Send Event	M/C
DMAST	Store	C
DMASTP	Store and notify Peripheral	C
DMASTZ	Store Zero	C
DMAWFE	Wait For Event M	M/C
DMAWFP	Wait For Peripheral	C
DMAWMB	Write Memory Barrier	C
DMAADNH	Add Negative Halfword	C

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

16.7.7 Assembler Directives

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

DMAADNH

Add Negative Halfword adds an immediate negative 16-bit value to the DMAC_SARn(n=0~7) Register or DMAC_DARn(n=0~7) Register, for the DMA channel thread. This enables the DMAC to support 2D DMA operations, or reading or writing an area of memory in a different order to naturally incrementing addresses. See Source Address Registers and Destination Address Registers.

The immediate unsigned 16-bit value is one-extended to 32 bits, to create a value that is the two's complement representation of a negative number between -65536 and -1, before the DMAC adds it to the address using 32-bit addition. The DMAC discards the carry bit so that addresses wrap from 0xFFFFFFFF to 0x00000000. The net effect is to subtract between 65536 and 1 from the current value in the Source or Destination Address Register.

Following table shows the instruction encoding.

Table 16-5 DMAC Instruction Encoding

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

Assembler syntax

DMAADNH <address_register>, <16-bit immediate>

where:

<address_register>

Selects the address register to use. It must be either:

SAR

DMAC_SARn(n=0~7) Register and sets ra to 0.

DAR

DMAC_DARn(n=0~7) Register and sets ra to 1.

<16-bit immediate>

The immediate value to be added to the <address_register>.

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

Chapter 17 I2S_TDM

17.1 Overview

The I2S/PCM/TDM controller is designed for interfacing between the AHB bus and the I2S bus.
The I2S bus (Inter-IC sound bus) is a serial link for digital audio data transfer between devices in the system and is invented by Philips Semiconductor. Now it is widely used by many semiconductor manufacturers.
I2S bus is widely used in the devices such as ADC, DAC, DSP, CPU, etc. With the I2S interface, we can connect audio devices and the embedded SoC platform together and provide an audio interface solution for the system.

17.1.1 Features

There is one I2S/PCM/TDM controller and two I2S/PCM controllers embed in the system. These three controllers are named as I2S0, I2S1 and I2S2. The I2S0 supports I2S, PCM and TDM mode while the I2S1 or I2S2 supports I2S and PCM mode stereo audio output and input. Unless stated separately, all of the following features apply to I2S0, I2S1 and I2S2.

- Support eight internal 32-bit wide and 32-location deep FIFOs, four for transmitting and the others for receiving audio data for I2S0
- Support two internal 32-bit wide and 32-location deep FIFOs, one for transmitting and the other for receiving audio data for each I2S1 and I2S2
- 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 a total of 10 channels transmitting and receiving in I2S mode at the same time for I2S0
- Support 2-channel audio transmitting and receiving in PCM mode for I2S0
- Support 2-channel audio receiving in I2S mode and 2 channel in PCM mode for I2S1
- Support 2-channel audio transmitting in I2S mode and 2-channel in PCM mode for I2S2
- Support up to 192kHz sample rate
- Support I2S normal, left and right justified mode serial audio data transfer
- Support PCM early, late1, late2, late3 mode serial audio data transfer
- Support TDM normal, 1/2 cycle left shift , 1 cycle left shift, 2 cycle left shift, right shift mode serial audio data transfer for I2S0
- 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
- Support a range of 16 to 32 programmable slot bit width in TDM mode for I2S0
- Support a range of 32 to 512 programmable frame width in TDM mode for I2S0
- Support programmable FSYNC width in TDM mode for I2S0

17.2 Controller Block Diagram

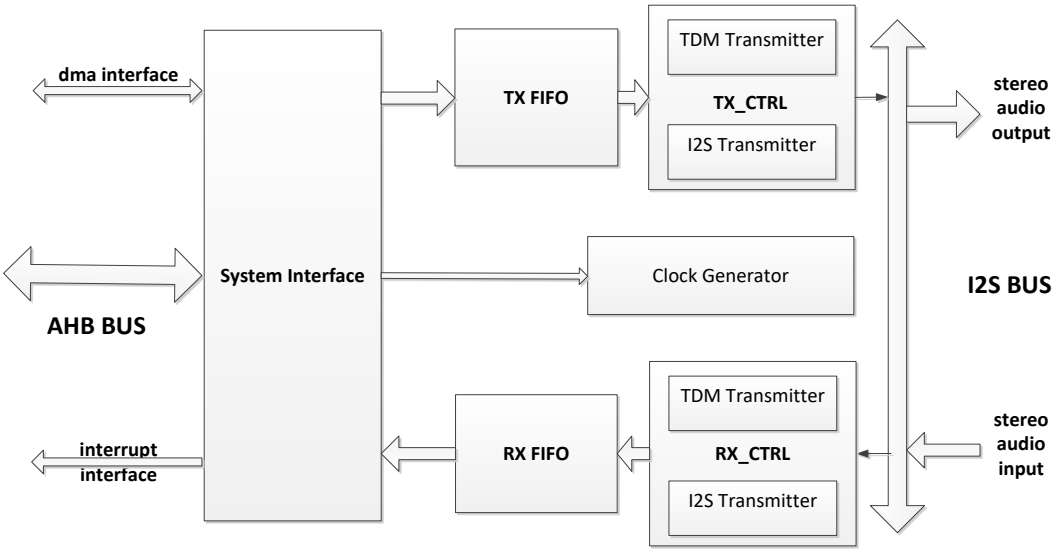


Fig.17-1 I2S/PCM/TDM Controller (8-channel) Block Diagram

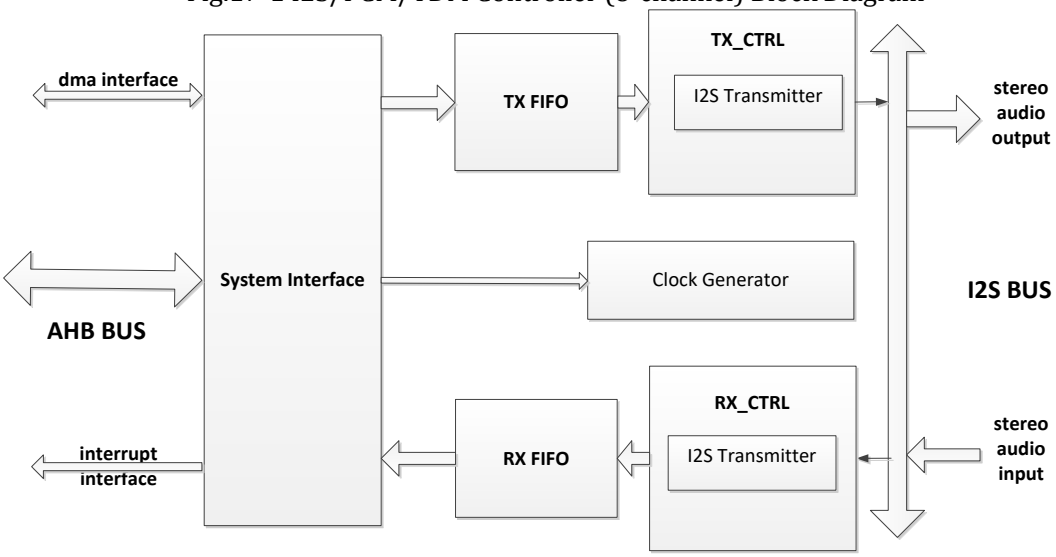


Fig.17-2 I2S/PCM Controller (2-channel) Block Diagram

System Interface

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

Clock Generator

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

Transmitters

The Transmitters implement transmission operation. The transmitters can act as either a master or a slave, with I2S, PCM or TDM mode surround serial audio interface.

Receiver

The Receiver implements receive operation. The receiver can act as either a master or a slave, with I2S, PCM or TDM mode stereo serial audio interface.

Transmit FIFO

The Transmit FIFO is the buffer to store transmitted audio data. The size of 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.

17.3 Function description

In the I2S/PCM/TDM or 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/TDM or 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/TDM or I2S/PCM controller is used as a receiver and external or internal audio CODEC is used as a transmitter. Either the I2S/PCM/TDM or 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.

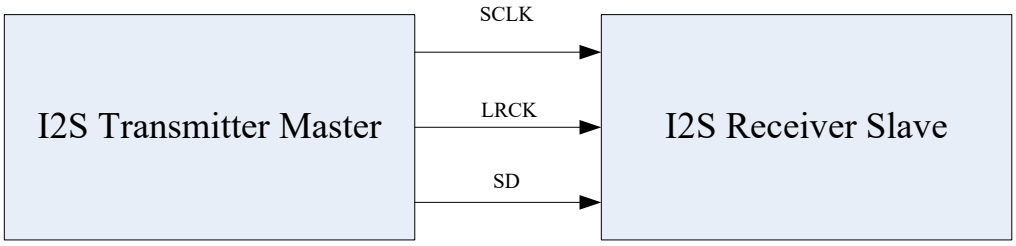


Fig.17-3 I2S Transmitter-Master & Receiver-Slave Condition

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.

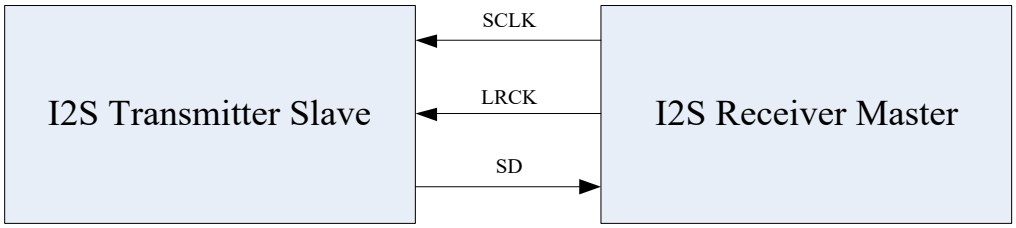


Fig.17-4 I2S Transmitter-Slave & Receiver-Master Condition

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.

17.3.1 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.

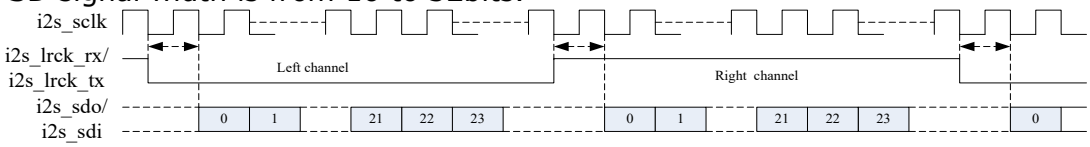


Fig.17-5 I2S Normal Mode Timing Format

17.3.2 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.

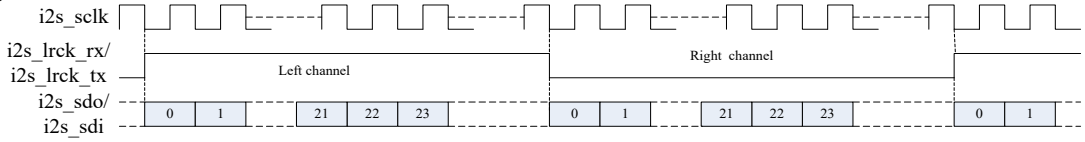


Fig.17-6 I2S Left Justified Mode Timing Format

17.3.3 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.

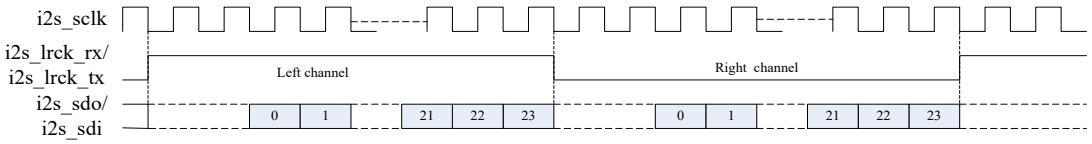


Fig.17-7 I2S Right Justified Mode Timing Format

17.3.4 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.

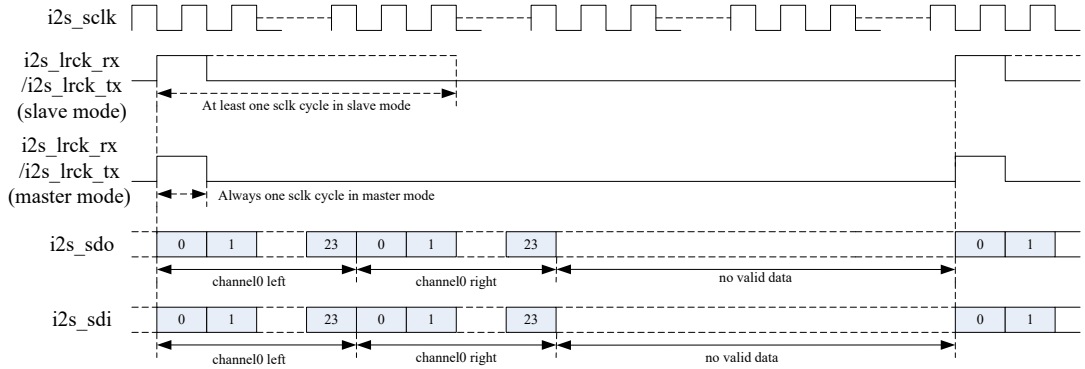


Fig.17-8 PCM Early Mode Timing Format

17.3.5 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.

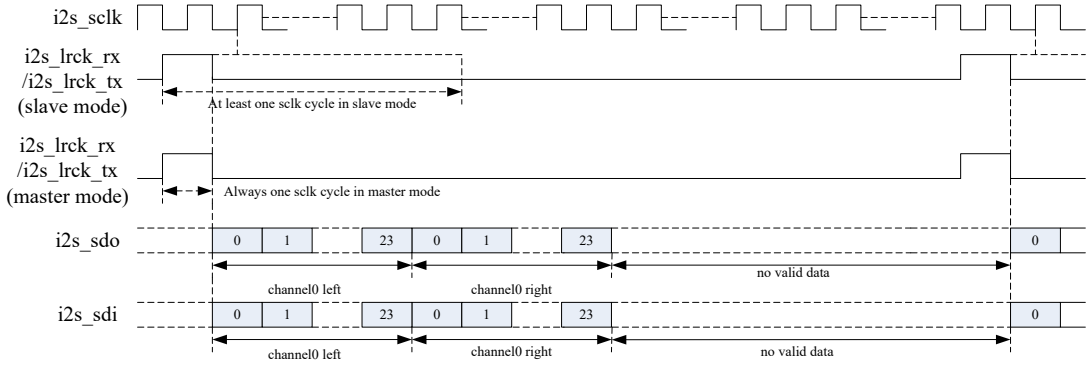


Fig.17-9 PCM Late1 Mode Timing Format

17.3.6 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.

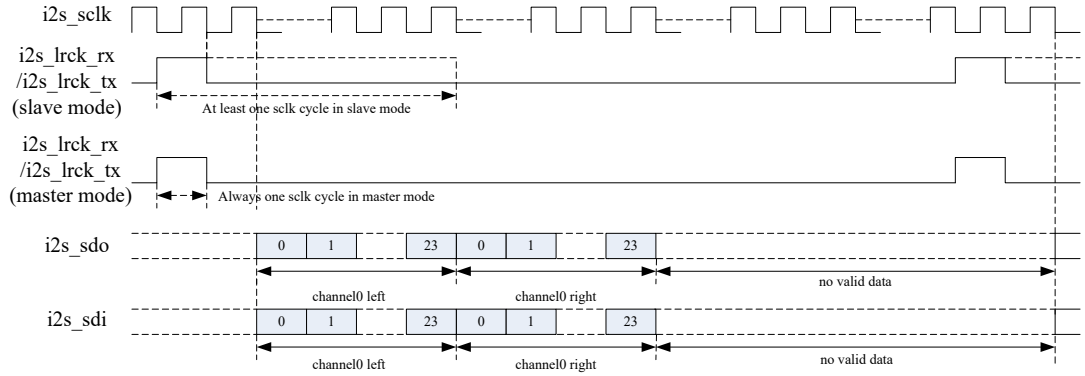


Fig.17-10 PCM Late2 Mode Timing Format

17.3.7 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.

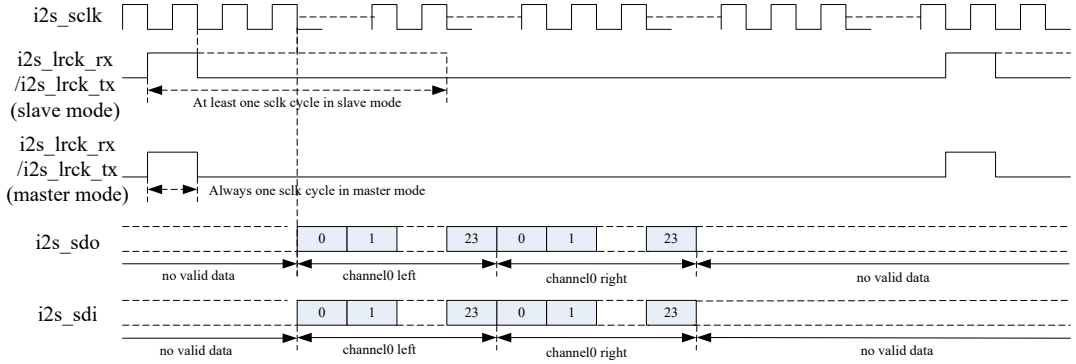


Fig.17-11 PCM Late3 Mode Timing Format

17.3.8 TDM Normal Mode (PCM Format)

This is the waveform of TDM normal 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) on the second falling edge of SCLK after LRCK goes high. The range of SD signal width is from 16 to 32bits.

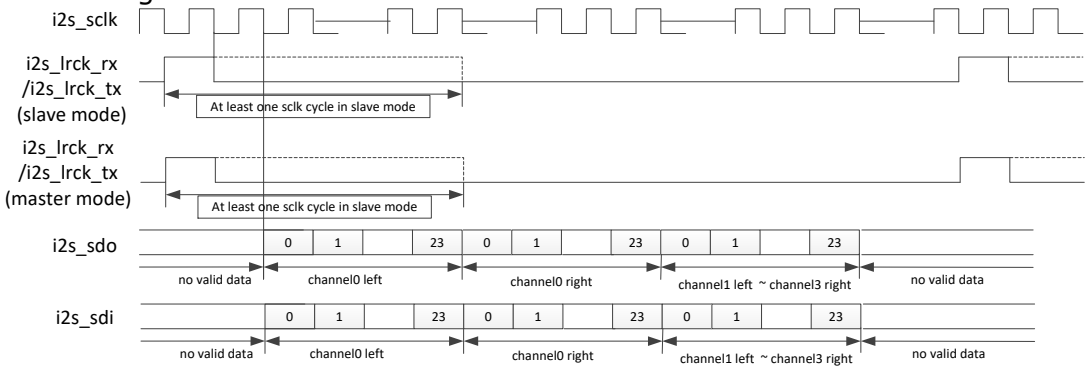


Fig.17-12 TDM Normal Mode Timing Format (PCM Format)

17.3.9 TDM Left Shift Mode0 (PCM Format)

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) on the second rising edge of SCLK after LRCK goes high. The range of SD signal width is from 16 to 32bits.

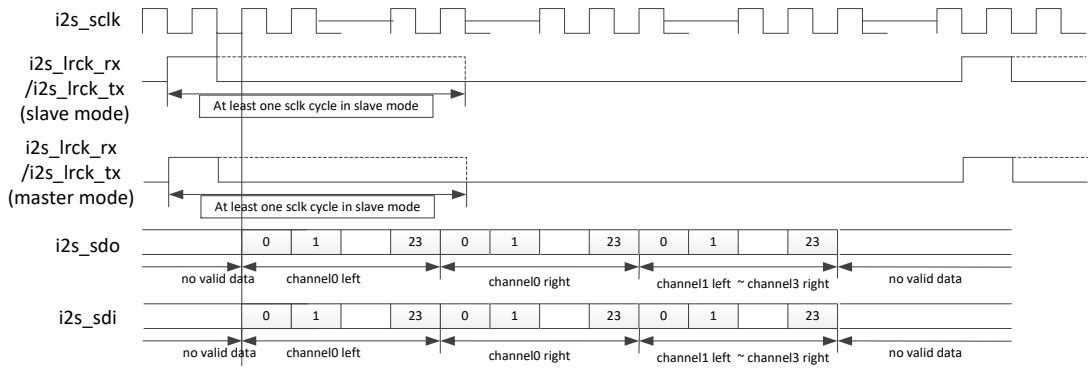


Fig.17-13 TDM Left Shift Mode 0 Timing Format (PCM Format)

17.3.10 TDM Left Shift Mode1 (PCM Format)

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) on the first falling edge of SCLK after LRCK goes high. The range of SD signal width is from 16 to 32bits.

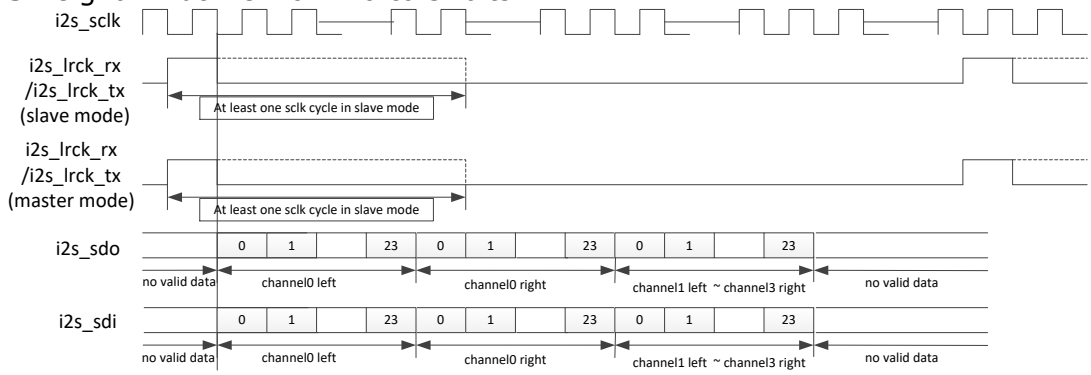


Fig.17-14 TDM Left Shift Mode 1 Timing Format (PCM Format)

17.3.11 TDM Left Shift Mode2 (PCM Format)

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) on the first rising edge of SCLK after LRCK goes high. The range of SD signal width is from 16 to 32bits.

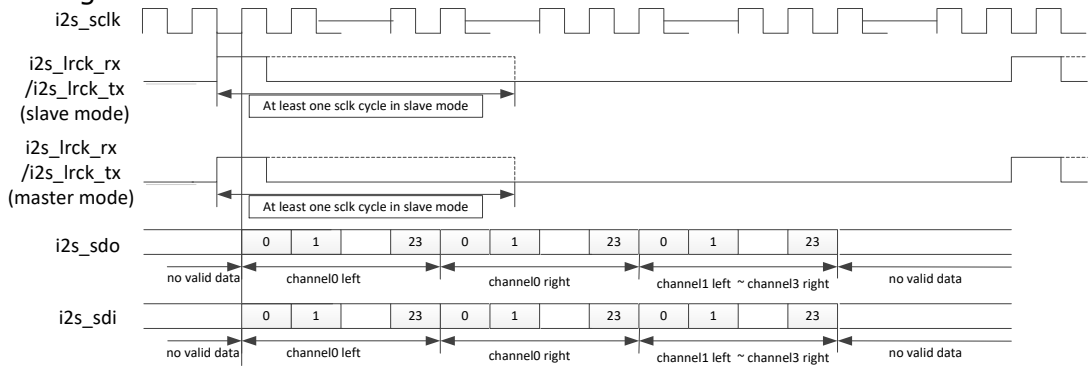


Fig.17-15 TDM Left Shift Mode 2 Timing Format (PCM Format)

17.3.12 TDM Left Shift Mode3 (PCM Format)

This is the waveform of PCM early mode. For LRCK (i2s1_lrck_rx/i2s1_lrck_tx) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s1_sdo, i2s1_sdi) signal, it sends the first bit (MSB or LSB) at the same time when LRCK goes high. The range of SD signal width is from 16 to 32bits.

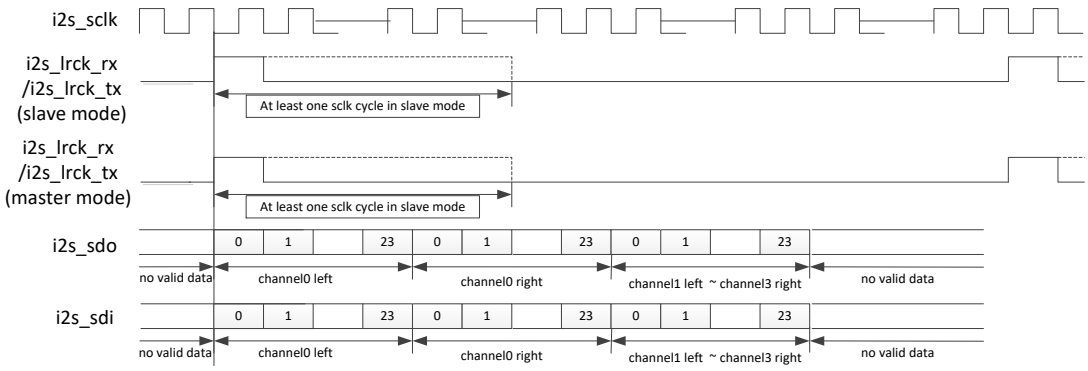
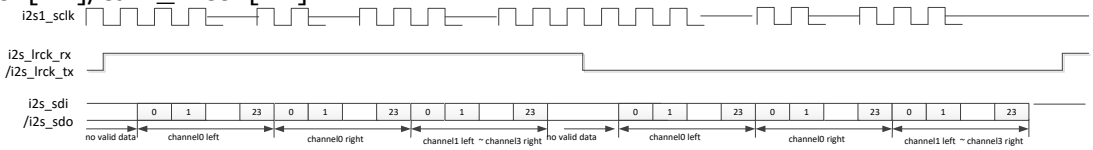


Fig.17-16 TDM Left Shift Mode 3 Timing Format (PCM Format)

17.3.13 TDM Normal Mode (I2S Format)

This is the waveform of I2S normal mode. For SD (i2s_sdo, i2s_sdi) signal, it starts sending the first bit (MSB or LSB)on the first falling edge of SCLK after LRCK changes. The range of SD signal width is from 16 to 32bits.

tdm_txctrl[17]/tdm_rxctrl[17]=1:



tdm_txctrl[17]/tdm_rxctrl[17]=0:

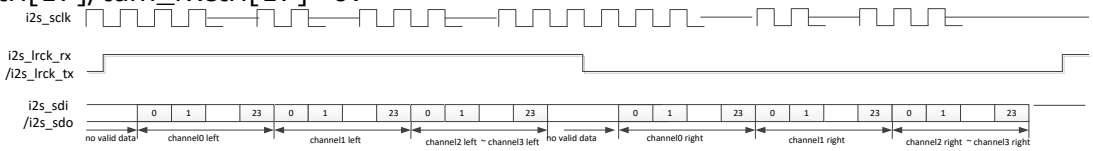
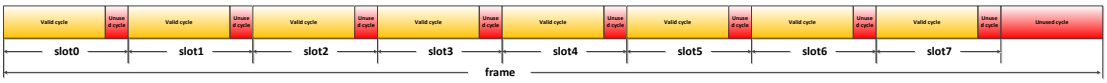


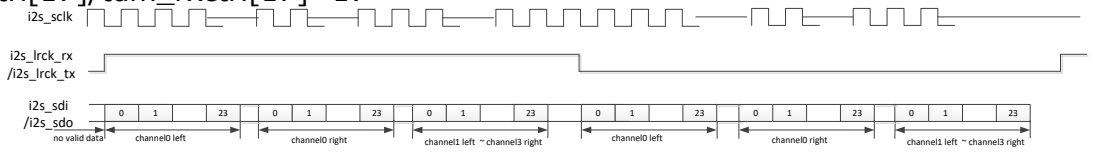
Fig.17-17 TDM Normal Mode Timing Format (I2S Format)

17.3.14 TDM Left Justified Mode (I2S Format)

This is the waveform of I2S left justified mode. For SD (i2s_sdo, i2s_sdi) signal, it starts sending the first bit (MSB or LSB) at the same time when LRCK changes. The range of SD signal width is from 16 to 32bits.



tdm_txctrl[17]/tdm_rxctrl[17]=1:



tdm_txctrl[17]/tdm_rxctrl[17]=0:

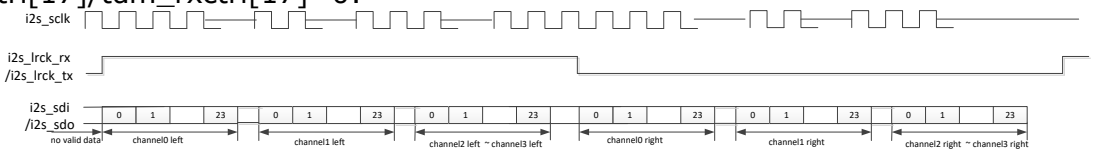
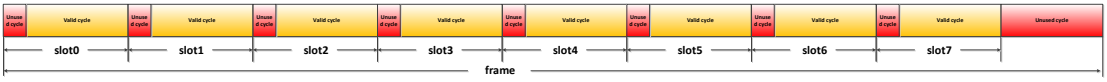


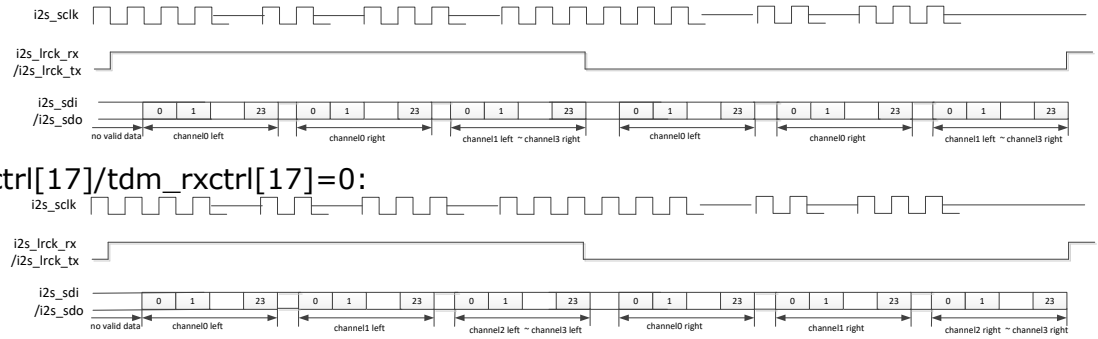
Fig.17-18 TDM Left Justified Mode Timing Format (I2S Format)

17.3.15 TDM Right Justified Mode (I2S Format)

This is the waveform of I2S right justified mode. For SD (i2s_sdo, i2s_sdi) signal, it transfers MSB or LSB first; but what is different from I2S normal or left justified mode. The range of SD signal width is from 16 to 32bits.



tdm_txctrl[17]/tdm_rxctrl[17]=1:



tdm_txctrl[17]/tdm_rxctrl[17]=0:

Fig.17-19 TDM Right Justified Mode Timing Format (I2S Format)

17.4 Register description

17.4.1 Internal Address Mapping

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

17.4.2 Registers Summary for I2S0

Name	Offset	Size	Reset Value	Description
I2S_TDM_8CH_TXCR	0x0000	W	0x7200000F	Transmit Operation Control Register
I2S_TDM_8CH_RXCR	0x0004	W	0x01C8000F	Receive Operation Control Register
I2S_TDM_8CH_CKR	0x0008	W	0x00070000	Clock Generation Register
I2S_TDM_8CH_TXFIFOLR	0x000C	W	0x00000000	TX FIFO Level Register
I2S_TDM_8CH_DMACR	0x0010	W	0x001F0000	DMA Control Register
I2S_TDM_8CH_INTCR	0x0014	W	0x01F00000	Interrupt Control Register
I2S_TDM_8CH_INTSR	0x0018	W	0x00000000	Interrupt Status Register
I2S_TDM_8CH_XFER	0x001C	W	0x00000000	Transfer Start Register
I2S_TDM_8CH_CLR	0x0020	W	0x00000000	Sclk Domain Logic Clear Register
I2S_TDM_8CH_TXDR	0x0024	W	0x00000000	Transmit FIFO Data Register
I2S_TDM_8CH_RXDR	0x0028	W	0x00000000	Receive FIFO Data Register
I2S_TDM_8CH_RXFIFOLR	0x002C	W	0x00000000	RX FIFO Level Register
I2S_TDM_8CH_TDM_TXCTRL	0x0030	W	0x00003EFF	TDM Mode Transmit Operation Control Register
I2S_TDM_8CH_TDM_RXCTRL	0x0034	W	0x00003EFF	TDM Mode Receive Operation Control Register
I2S_TDM_8CH_CLKDIV	0x0038	W	0x00000707	Clock Divider Register
I2S_TDM_8CH_VERSION	0x003C	W	0x20150001	Version Register

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

17.4.3 Detail Registers Description for I2S0

I2S_TDM_8CH_TXCR

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved

Bit	Attr	Reset Value	Description
30:29	RW	0x3	TX_PATH_SEL3 TX Path Select 3 2'b00: Sdo3 output data from path0 2'b01: Sdo3 output data from path1 2'b10: Sdo3 output data from path2 2'b11: Sdo3 output data from path3 Note: When TDM mode, only path0 enable.
28:27	RW	0x2	TX_PATH_SEL2 TX Path Select 2 2'b00: Sdo2 output data from path0 2'b01: Sdo2 output data from path1 2'b10: Sdo2 output data from path2 2'b11: Sdo2 output data from path3 Note: When TDM mode, only path0 enable.
26:25	RW	0x1	TX_PATH_SEL1 TX Path Select 1 2'b00: Sdo1 output data from path0 2'b01: Sdo1 output data from path1 2'b10: Sdo1 output data from path2 2'b11: Sdo1 output data from path3 Note: When TDM mode, only path0 enable.
24:23	RW	0x0	TX_PATH_SEL0 TX Path Select 0 2'b00: Sdo0 output data from path0 2'b01: Sdo0 output data from path1 2'b10: Sdo0 output data from path2 2'b11: Sdo0 output data from path3 Note: When TDM mode, only path0 enable.
22:17	RW	0x00	RCNT Can be written only when XFER[0] bit is 0. Only valid in I2S Right justified format and slave TX mode is selected. Start to transmit data RCNT sclk cycles after left channel valid. Note: Only function when TX TFS[1]=0.
16:15	RW	0x0	TCSR Transmit Channel Select Register 2'b00: Two channel 2'b01: Four channel 2'b10: Six channel 2'b11: Eight channel

Bit	Attr	Reset Value	Description
14	RW	0x0	HWT Halfword Word Transform Can be written only when XFER[0] bit is 0. Only valid when VDW select 16bit data. 1'b0: 32 bit data valid from AHB/APB bus. Low 16 bit for left channel and high 16 bit for right channel. 1'b1: Low 16bit data valid from AHB/APB bus, high 16 bit data invalid.
13	RO	0x0	reserved
12	RW	0x0	SJM Store Justified Mode Can be written only when XFER[0] bit is 0. 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: TDM format 0 (PCM mode) 2'b11: TDM format 1 (I2S mode)

Bit	Attr	Reset Value	Description
4:0	RW	0x0f	VDW Valid Data Width Can be written only when XFER[0] bit is 0. 5'b00000~5'b01110: Reserved 5'b01111: 16bit 5'b10000: 17bit 5'b10001: 18bit 5'b10010: 19bit ..... 5'b11100: 29bit 5'b11101: 30bit 5'b11110: 31bit 5'b11111: 32bit

I2S_TDM_8CH_RXCR

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:25	RO	0x00	reserved
24:23	RW	0x3	RX_PATH_SEL3 Rx Path Select 3 2'b00: Path3 data from sdi0 2'b01: Path3 data from sdi1 2'b10: Path3 data from sdi2 2'b11: Path3 data from sdi3 Note: Inoperative at TDM mode.
22:21	RW	0x2	RX_PATH_SEL2 Rx Path Select 2 2'b00: Path2 data from sdi0 2'b01: Path2 data from sdi1 2'b10: Path2 data from sdi2 2'b11: Path2 data from sdi3 Note: Inoperative at TDM mode.
20:19	RW	0x1	RX_PATH_SEL1 Rx Path Select 1 2'b00: Path1 data from sdi0 2'b01: Path1 data from sdi1 2'b10: Path1 data from sdi2 2'b11: Path1 data from sdi3 Note: Inoperative at TDM mode.
18:17	RW	0x0	RX_PATH_SEL0 Rx Path Select 0 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: TDM format 0 (PCM mode) 2'b11: TDM format 1 (I2S mode)
4:0	RW	0x0f	VDW Valid Data Width Can be written only when XFER[1] bit is 0. 5'b00000~5'b01110: Reserved 5'b01111: 16bit 5'b10000: 17bit 5'b10001: 18bit 5'b10010: 19bit ..... 5'b11100: 29bit 5'b11101: 30bit 5'b11110: 31bit 5'b11111: 32bit

I2S_TDM_8CH_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_lrck/rx_lrck are used as synchronous signal for TX /RX respectively. 2'b01: Only tx_lrck is used as synchronous signal for TX and RX. 2'b10: Only rx_lrck is used as synchronous signal for TX and RX.
27	RW	0x0	MSS Master/Slave Mode Select Can be written only when XFER[1] or XFER[0] bit is 0. 1'b0: Master mode(sclk output) 1'b1: Slave mode(sclk input)
26	RW	0x0	CKP Sclk Polarity Can be written only when XFER[1] or XFER[0] bit is 0. 1'b0: Sample data at posedge sclk and drive data at negedge sclk 1'b1: Sample data at negedge sclk and drive data at posedge sclk

Bit	Attr	Reset Value	Description
25	RW	0x0	RLP Receive Lrck Polarity Can be written only when XFER[1] or XFER[0] bit is 0. 1'b0: Normal polarity (I2S normal: Low for left channel, high for right channel I2S left/right just: High for left channel, low for right channel PCM start signal: High valid) 1'b1: Opposite polarity (I2S normal: High for left channel, low for right channel I2S left/right just: Low for left channel, high for right channel PCM start signal: Low valid)
24	RW	0x0	TLP Transmit Lrck Polarity Can be written only when XFER[1] or XFER[0] bit is 0. 1'b0: Normal polarity (I2S normal: Low for left channel, high for right channel I2S left/right just: High for left channel, low for right channel PCM start signal: High valid) 1'b1: Opposite polarity (I2S normal: High for left channel, low for right channel I2S left/right just: Low for left channel, high for right channel PCM start signal: Low valid)
23:16	RW	0x07	MDIV Reserved.
15:8	RW	0x00	RSD Receive Sclk Divider Can be written only when XFER[1] or XFER[0] bit is 0. 8'h00~8'h1e: Reserved 8'h1f~8'hff: Frequency of sclk = (receive sclk divider/2)*2*frequency of rx_lrck Note: Function when RX TFS[1:0] is 2'b00 or 2'b01.
7:0	RW	0x00	TSD Transmit Sclk Divider Can be written only when XFER[1] or XFER[0] bit is 0. 8'h00~8'h1e: Reserved 8'h1f~8'hff: Frequency of sclk = (Transmit sclk divider/2)*2*frequency of tx_lrck Note: Function when TX TFS[1:0] is 2'b00 or 2'b01.

I2S_TDM_8CH_TXFIFOLR

Address: Operational Base + offset (0x000C)

Bit	Attr	Reset Value	Description
31:24	RO	0x00	reserved

Bit	Attr	Reset Value	Description
23:18	RO	0x00	TFL3 Transmit FIFO3 Level Contains the number of valid data entries in the transmit FIFO3.
17:12	RO	0x00	TFL2 Transmit FIFO2 Level Contains the number of valid data entries in the transmit FIFO2.
11:6	RO	0x00	TFL1 Transmit FIFO1 Level Contains the number of valid data entries in the transmit FIFO1.
5:0	RO	0x00	TFL0 Transmit FIFO0 Level Contains the number of valid data entries in the transmit FIFO0.

I2S_TDM_8CH_DMACR

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:25	RO	0x00	reserved
24	RW	0x0	RDE Receive DMA Enable 1'b0: Receive DMA disabled 1'b1: Receive DMA enabled
23:21	RO	0x0	reserved
20:16	RW	0x1f	RDL Receive Data Level This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or above this field value + 1.
15:9	RO	0x00	reserved
8	RW	0x0	TDE Transmit DMA Enable 1'b0: Transmit DMA disabled 1'b1: Transmit DMA enabled
7:5	RO	0x0	reserved
4:0	RW	0x00	TDL Transmit Data Level This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of valid data entries in the TX FIFO(TX FIFO0 if CSR=00;TX FIFO1 if CSR=01,TX FIFO2 if CSR=10,TX FIFO3 if CSR=11)is equal to or below this field value.

I2S_TDM_8CH_INTCR

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:25	RO	0x00	reserved
24:20	RW	0x1f	RFT Receive FIFO Threshold When the number of receive FIFO entries is more than or equal to this threshold plus 1, the receive FIFO full interrupt is triggered.
19	RO	0x0	reserved
18	WO	0x0	RXOIC RX Overrun Interrupt Clear Write 1 to clear RX overrun interrupt.
17	RW	0x0	RXOIE RX Overrun Interrupt Enable 1'b0: Disable 1'b1: Enable
16	RW	0x0	RXFIE RX Full Interrupt Enable 1'b0: Disable 1'b1: Enable
15:9	RO	0x00	reserved
8:4	RW	0x00	TFT Transmit FIFO Threshold When the number of transmit FIFO entries is less than or equal to this threshold, the transmit FIFO empty interrupt is triggered.
3	RO	0x0	reserved
2	RW	0x0	TXUIC TX Underrun Interrupt Clear Write 1 to clear TX underrun interrupt.
1	RW	0x0	TXUIE TX Underrun Interrupt Enable 1'b0: Disable 1'b1: Enable
0	RW	0x0	TXEIE TX empty Interrupt Enable 1'b0: Disable 1'b1: Enable

I2S_TDM_8CH_INTSR

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:18	RO	0x0000	reserved
17	RO	0x0	RXOI RX Overrun Interrupt 1'b0: Inactive 1'b1: Active

Bit	Attr	Reset Value	Description
16	RO	0x0	RXFI RX Full Interrupt 1'b0: Inactive 1'b1: Active
15:2	RO	0x0000	reserved
1	RO	0x0	TXUI TX Underrun Interrupt 1'b0: Inactive 1'b1: Active
0	RO	0x0	TXEI TX Empty Interrupt 1'b0: Inactive 1'b1: Active

I2S_TDM_8CH_XFER

Address: Operational Base + offset (0x001C)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	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_TDM_8CH_CLR

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved
1	RW	0x0	RXC RX Logic Clear This is a self-cleared bit. Write 1 to clear all receive logic.
0	RW	0x0	TXC TX Logic Clear This is a self-cleared bit. Write 1 to clear all transmit logic.

I2S_TDM_8CH_TXDR

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	TXDR Transmit FIFO Data Register When it is written, data are moved into the transmit FIFO.

I2S_TDM_8CH_RXDR

Address: Operational Base + offset (0x0028)

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

I2S_TDM_8CH_RXFIFOLR

Address: Operational Base + offset (0x002C)

Bit	Attr	Reset Value	Description
31:24	RO	0x00	reserved
23:18	RO	0x00	RFL3 Receive FIFO3 Level Contains the number of valid data entries in the receive FIFO3.
17:12	RO	0x00	RFL2 Receive FIFO2 Level Contains the number of valid data entries in the receive FIFO2.
11:6	RO	0x00	RFL1 Receive FIFO1 Level Contains the number of valid data entries in the receive FIFO1.
5:0	RO	0x00	RFL0 Receive FIFO0 Level Contains the number of valid data entries in the receive FIFO0.

I2S_TDM_8CH_TDM_TXCTRL

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:21	RO	0x000	reserved
20:18	RW	0x0	TX_TDM_FSYNC_WIDTH_SEL1 TDM Transfer Fsync Width Sel1 Can be written only when XFER[0] is 0. 3'b000: Single period of the sclk_tx. 3'b001: 2 period of the sclk_tx. n: n+1 period of the sclk_tx. 3'b110: 7 period of the sclk_tx. 3'b111: The width is equivalent to a channel block. Note: Function when TX TFS[1:0] is 2 or 3.
17	RW	0x0	TX_TDM_FSYNC_WIDTH_SEL0 TDM Transfer Fsync Width Sel0 Can be written only when XFER[0] is 0. 1'b0: 1/2 frame width. It should be set to an even number. 1'b1: Frame width.

Bit	Attr	Reset Value	Description
16:14	RW	0x0	TDM_TX_SHIFT_CTRL TDM Transfer Shift Ctrl Can be written only when XFER[0] is 0. 3'b000: PCM format 0: Normal mode, drive data on the second negedge of sclk_tx after rising edge of TX LRCK. I2S format 0: Normal mode 3'b001: PCM format 1: 1/2 cycle shift left, drive data on second posedge of sclk_tx after rising edge of TX LRCK. I2S format 1: Left justified mode 3'b010: PCM format 2: 1 cycle shift left, drive data on first negedge of sclk_tx after rising edge of TX LRCK. I2S format 2: Right justified mode 3'b011: PCM format 3: 3/2 cycle shift left, drive data on first posedge of sclk_tx after rising edge of TX LRCK. I2S format: Not support 3'b100: PCM format 4: 2 cycle shift left, drive data aligned to the posedge of TX LRCK. I2S format: Not support 3'b101~3'b111: Not support Note: Function when TX TFS[1:0] is 2 or 3.
13:9	RW	0x1f	TDM_TX_SLOT_BIT_WIDTH TDM Transfer Slot Bits Can be written only when XFER[0] is 0. 5'h00~5'h0e: Reserved 5'h0f: 16bit 5'h10: 17bit 5'h11: 18bit 5'h12: 19bit ..... 5'h1f: 32bit Note: Function when TX TFS[1:0] is 2 or 3.

Bit	Attr	Reset Value	Description
8:0	RW	0x0ff	TDM_TX_FRAME_WIDTH TDM Transfer Frame Width Can be written only when XFER[0] is 0. 9'h000~9'h01e: Reserved 9'h01f: 32bit 9'h020: 33bit 9'h021: 34bit 9'h022: 35bit ..... 9'h1ff: 512bit Note: Functional when TX TFS[1:0] is 2 or 3.

I2S_TDM_8CH_TDM_RXCTRL

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:21	RO	0x000	reserved
20:18	RW	0x0	RX_TDM_FSYNC_WIDTH_SEL1 TDM Receive Fsync Width Sel1 Can be written only when XFER[1] is 0. 3'b000: Single period of the sclk_rx. 3'b001: 2 period of the sclk_rx. n: n+1 period of the sclk_rx. 3'b110: 7 period of the sclk_rx. 3'b111: The width is equivalent to a channel block Note: Function when RX TFS[1:0] is 2 or 3.
17	RW	0x0	RX_TDM_FSYNC_WIDTH_SEL0 TDM Receive Fsync Width Sel0 Can be written only when XFER[1] is 0. 1'b0: 1/2 frame width. It should be set to an even number. 1'b1: Frame width

Bit	Attr	Reset Value	Description
16:14	RW	0x0	TDM_RX_SHIFT_CTRL TDM Receive Shift Ctrl Can be written only when XFER[1] is 0. 3'b000: PCM format 0: Normal mode, sample data on the third posedge of sclk_rx after rising edge of RX LRCK. I2S format 0: Normal mode 3'b001: PCM format 1: 1/2 cycle shift left, sample data on second negedge of sclk_rx after rising edge of RX LRCK. I2S format 1: left justified mode 3'b010: PCM format 2: 1 cycle shift left, sample data on second posedge of sclk_rx after rising edge of RX LRCK. I2S format 2: Right justified mode 3'b011: PCM format 3: 3/2 cycle shift left, sample data on first negedge of sclk_rx after rising edge of RX LRCK. I2S format: Not support 3'b100: PCM format 4: 2 cycle shift left, sample data on the first posedge of sclk_rx after rising edge of RX LRCK. I2S format: Not support 3'b101~3'b111: Not support Note: Function when RX TFS[1:0] is 2 or 3.
13:9	RW	0x1f	TDM_RX_SLOT_BIT_WIDTH TDM Receive Slot Bits Can be written only when XFER[1] is 0. 5'h00~5'h0e: Reserved 5'h0f: 16bit 5'h10: 17bit 5'h11: 18bit 5'h12: 19bit ..... 5'h1f: 32bit Note: Function when RX TFS[1:0] is 2 or 3.

Bit	Attr	Reset Value	Description
8:0	RW	0x0ff	TDM_RX_FRAME_WIDTH TDM Receive Frame Width Can be written only when XFER[1] is 0. 9'h000~9'h01e: Reserved 9'h01f: 32bit 9'h020: 33bit 9'h021: 34bit 9'h022: 35bit ..... 9'h1ff: 512bit Note: Functional when RX TFS[1:0] is 2 or 3.

I2S_TDM_8CH_CLKDIV

Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15:8	RW	0x07	RX_MDIV RX Mclk Divider Can be written only when XFER[1] bit is 0. mclk_rx divider = (mclk_rx/sclk_rx)-1. For example, if mclk_rx divider is 5, then the frequency of sclk_rx is mclk_rx/6.
7:0	RW	0x07	TX_MDIV TX Mclk 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_TDM_8CH_VERSION

Address: Operational Base + offset (0x003C)

Bit	Attr	Reset Value	Description
31:0	RO	0x20150001	I2S_VERSION I2S Version

17.4.4 Registers Summary for I2S1/I2S2

Name	Offset	Size	Reset Value	Description
I2S_2CH_TXCR	0x0000	W	0x0000000F	Transmit Operation Control Register.
I2S_2CH_RXCR	0x0004	W	0x0000000F	Receive Operation Control Register
I2S_2CH_CKR	0x0008	W	0x00071F1F	Clock Generation Register
I2S_2CH_FIFOLR	0x000C	W	0x00000000	FIFO Level Register
I2S_2CH_DMACR	0x0010	W	0x001F0000	DMA Control Register
I2S_2CH_INTCR	0x0014	W	0x01F00000	Interrupt Control Register
I2S_2CH_INTSR	0x0018	W	0x00000000	Interrupt Status Register
I2S_2CH_XFER	0x001C	W	0x00000000	Transfer Start Register

Name	Offset	Size	Reset Value	Description
I2S_2CH_CLR	0x0020	W	0x00000000	SCLK Domain Logic Clear Register
I2S_2CH_TXDR	0x0024	W	0x00000000	Transmit FIFO Data Register
I2S_2CH_RXDR	0x0028	W	0x00000000	Receive FIFO Data Register

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

17.4.5 Detail Registers Description for I2S1/I2S2

I2S_2CH_TXCR

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:23	RO	0x000	reserved
22:17	RW	0x00	RCNT Right Justified Counter Can be written only when XFER[0] bit is 0. Only valid in I2S right justified format and slave TX mode is selected. Start to transmit data RCNT sclk cycles after left channel valid.
16:15	RW	0x0	CSR Channel Select Register 2'b00: 2 channel 2'b01~2'b11: Reserved
14	RW	0x0	HWT Halfword Word Transform Can be written only when XFER[0] bit is 0. Only valid when VDW select 16bit data. 1'b0: 32 bit data valid from AHB/APB bus. Low 16 bit for left channel and high 16 bit for right channel. 1'b1: Low 16bit data valid from AHB/APB bus, high 16 bit data invalid.
13	RO	0x0	reserved
12	RW	0x0	SJM Store Justified Mode Can be written only when XFER[0] bit is 0. 16bit~31bit DATA stored in 32 bits width FIFO. If VDW select 16bit data, this bit is valid only when HWT select 0.Because if HWT is 1, every FIFO unit contain two 16bit data and 32 bit space is full, it is impossible to choose justified mode. 1'b0: Right justified 1'b1: Left justified
11	RW	0x0	FBM First Bit Mode Can be written only when XFER[0] bit is 0. 1'b0: MSB 1'b1: LSB

Bit	Attr	Reset Value	Description
10:9	RW	0x0	IBM I2S Bus Mode Can be written only when XFER[0] bit is 0. 2'b00: I2S normal 2'b01: I2S Left justified 2'b10: I2S Right justified 2'b11: Reserved
8:7	RW	0x0	PBM PCM Bus Mode Can be written only when XFER[0] bit is 0. 2'b00: PCM no delay mode 2'b01: PCM delay 1 mode 2'b10: PCM delay 2 mode 2'b11: PCM delay 3 mode
6	RO	0x0	reserved
5	RW	0x0	TFS Transfer Format Select Can be written only when XFER[0] bit is 0. 1'b0: I2S format 1'b1: PCM format
4:0	RW	0x0f	VDW Valid Data Width Can be written only when XFER[0] bit is 0. 5'b00000~5'b01110: reserved 5'b01111: 16bit 5'b10000: 17bit 5'b10001: 18bit 5'b10010: 19bit ..... 5'b11100: 29bit 5'b11101: 30bit 5'b11110: 31bit 5'b11111: 32bit

I2S_2CH_RXCR

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:15	RO	0x00000	reserved

Bit	Attr	Reset Value	Description
14	RW	0x0	HWT Halfword Word Transform Can be written only when XFER[1] bit is 0. Only valid when VDW select 16bit data. 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 contain two 16bit data and 32 bit space is full, it is impossible to choose justified mode. 1'b0: Right justified 1'b1: Left justified
11	RW	0x0	FBM First Bit Mode (Can be written only when XFER[1] bit is 0.) 1'b0: MSB 1'b1: LSB
10:9	RW	0x0	IBM I2S Bus Mode (Can be written only when XFER[1] bit is 0.) 2'b00: I2S normal 2'b01: I2S Left justified 2'b10: I2S Right justified 2'b11: Reserved
8:7	RW	0x0	PBM PCM Bus Mode (Can be written only when XFER[1] bit is 0.) 2'b00: PCM no delay mode 2'b01: PCM delay 1 mode 2'b10: PCM delay 2 mode 2'b11: PCM delay 3 mode
6	RO	0x0	reserved
5	RW	0x0	TFS Transfer Format Select Can be written only when XFER[1] bit is 0. 1'b0: I2S format 1'b1: PCM format

Bit	Attr	Reset Value	Description
4:0	RW	0x0f	VDW Valid Data Width Can be written only when XFER[1] bit is 0. 5'b00000~5'b01110: reserved 5'b01111: 16bit 5'b10000: 17bit 5'b10001: 18bit 5'b10010: 19bit ..... 5'b11100: 29bit 5'b11101: 30bit 5'b11110: 31bit 5'b11111: 32bit

I2S_2CH_CKR

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:28	RW	0x0	TRCM TX/RX LRCK as common 2'b00/2'b11: Tx_lrck/rx_lrck are used as synchronous signal for TX /RX respectively. 2'b01: Only tx_lrck is used as synchronous signal for TX and RX. 2'b10: Only rx_lrck is used as synchronous signal for TX and RX. Note: When set to 2'b01 or 2'b10 , if user wants to use both transmitting and receiving in master mode, user should configure as following: a. The value of TSD and RSD should be same. b. User should start TX transfer and RX transfer at the same time.
27	RW	0x0	MSS Master/Slave Mode Select Can be written only when XFER[1] or XFER[0] bit is 0. 1'b0: Master mode(sclk output 1'b1: Slave mode(sclk input)
26	RW	0x0	CKP Sclk Polarity Can be written only when XFER[1] or XFER[0] bit is 0. 1'b0: Sample data at posedge sclk and drive data at negedge sclk 1'b1: Sample data at negedge sclk and drive data at posedge sclk

Bit	Attr	Reset Value	Description
25	RW	0x0	RLP Receive Lrck Polarity Can be written only when XFER[1] or XFER[0] bit is 0. 1'b0: Normal polarity (I2S normal: Low for left channel, high for right channel I2S left/right just: High for left channel, low for right channel PCM start signal: High valid) 1'b1: Opposite polarity (I2S normal: High for left channel, low for right channel I2S left/right just: Low for left channel, high for right channel PCM start signal: Low valid)
24	RW	0x0	TLP Transmit Lrck Polarity Can be written only when XFER[1] or XFER[0] bit is 0. 1'b0: normal polarity (I2S normal: Low for left channel, high for right channel I2S left/right just: High for left channel, low for right channel PCM start signal: High valid) 1'b1: Opposite polarity (I2S normal: High for left channel, low for right channel I2S left/right just: Low for left channel, high for right channel PCM start signal: Low valid)
23:16	RW	0x07	MDIV Mclk Divider Can be written only when XFER[1] or XFER[0] bit is 0. mclk divider = (mclk/sclk)-1. For example, if mclk divider is 5, then the frequency of sclk is mclk/6
15:8	RW	0x1f	RSD Receive Sclk Divider Can be written only when XFER[1] or XFER[0] bit is 0. 8'h00~8'h1e: Reserved 8'h1f~8'hff: Frequency of sclk = (receive sclk divider>>1 +1)*2*frequency of rx_lrck
7:0	RW	0x1f	TSD Transmit Sclk Divider Can be written only when XFER[1] or XFER[0] bit is 0. 8'h00~8'h1e: Reserved 8'h1f~8'hff: Frequency of sclk = (Transmit sclk divider>>1 + 1)*2*frequency of tx_lrck

I2S_2CH_FIFOLR

Address: Operational Base + offset (0x000C)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved

Bit	Attr	Reset Value	Description
29:24	RO	0x00	RFL Receive FIFO Level Contains the number of valid data entries in the receive FIFO.
23:6	RO	0x00000	reserved
5:0	RW	0x00	TFL Transmit FIFO0 Level Contains the number of valid data entries in the transmit FIFO.

I2S_2CH_DMACR

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:25	RO	0x00	reserved
24	RW	0x0	RDE Receive DMA Enable 1'b0: Receive DMA disabled 1'b1: Receive DMA enabled
23:21	RO	0x0	reserved
20:16	RW	0x1f	RDL Receive Data Level This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or above this field value + 1.
15:9	RO	0x00	reserved
8	RW	0x0	TDE Transmit DMA Enable 1'b0: Transmit DMA disabled 1'b1: Transmit DMA enabled
7:5	RO	0x0	reserved
4:0	RW	0x00	TDL Transmit Data Level This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of valid data entries in the TXFIFO is equal to or below this field value.

I2S_2CH_INTCR

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:25	RO	0x00	reserved
24:20	RW	0x1f	RFT Receive FIFO Threshold When the number of receive FIFO entries is more than or equal to this threshold plus 1, the receive FIFO full interrupt is triggered.
19	RO	0x0	reserved

Bit	Attr	Reset Value	Description
18	WO	0x0	RXOIC RX Overrun Interrupt Clear Write 1 to clear RX overrun interrupt.
17	RW	0x0	RXOIE RX Overrun Interrupt Enable 1'b0: Disable 1'b1: Enable
16	RW	0x0	RXFIE RX Full Interrupt Enable 1'b0: Disable 1'b1: Enable
15:9	RO	0x00	reserved
8:4	RW	0x00	TFT Transmit FIFO Threshold When the number of transmit FIFO entries is less than or equal to this threshold, the transmit FIFO empty interrupt is triggered.
3	RO	0x0	reserved
2	WO	0x0	TXUIC TX Under-run Interrupt Clear Write 1 to clear TX under-run 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_2CH_INTSR

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:18	RO	0x0000	reserved
17	RO	0x0	RXOI RX Overrun Interrupt 1'b0: Inactive 1'b1: Active
16	RO	0x0	RXFI RX Full Interrupt 1'b0: Inactive 1'b1: Active
15:2	RO	0x0000	reserved

Bit	Attr	Reset Value	Description
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_2CH_XFER

Address: Operational Base + offset (0x001C)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved
1	RW	0x0	RXS RX Transfer Start Bit 1'b0: Stop RX transfer 1'b1: Start RX transfer
0	RW	0x0	TXS TX Transfer Start Bit 1'b0: Stop TX transfer 1'b1: Start TX transfer

I2S_2CH_CLR

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved
1	RW	0x0	RXC RX Logic Clear This is a self-cleared bit. Write 1 to clear all receive logic.
0	RW	0x0	TXC TX Logic Clear This is a self-cleared bit. Write 1 to clear all transmit logic.

I2S_2CH_TXDR

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	TXDR Transmit FIFO Data Register When it is written to, data are moved into the transmit FIFO.

I2S_2CH_RXDR

Address: Operational Base + offset (0x0028)

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

17.5 Interface Description

Three I2S controllers embed in the chip are I2S0, I2S1 and I2S2. The I2S0 is connected to two groups of IO interfaces.
The following table shows the I2S0 group 0 interface description.

Table 17-1 I2S0 Group 0 Interface Description

Module Pin	Direction	Pad Name	IOMUX Setting
i2s0_mclk_tx or i2s0_mclk_rx	I/O	I2S0_MCLK_M0/GPIO3_D2_d	GRF_GPIO3D_IOMUX_L[10:8]=3'b001
i2s0_sclk_tx	I/O	I2S0_SCLK_TX_M0/ACODEC_DAC_CLK/GPIO3_D0_d	GRF_GPIO3D_IOMUX_L[2:0]=3'b001
i2s0_sclk_rx	I/O	I2S0_SCLK_RX_M0/PDM_CLk1_M0/ACODEC_ADC_CLK/GPIO3_D1_d	GRF_GPIO3D_IOMUX_L[6:4]=3'b001
i2s0_lrck_tx	I/O	I2S0_LRCK_TX_M0/ACODEC_DAC_SYNC/AUDPWM_L_M1/AUDDSM_LN/GPIO3_D3_d	GRF_GPIO3D_IOMUX_L[14:12]=3'b001
i2s0_lrck_rx	I/O	I2S0_LRCK_RX_M0/PDM_CLk0_M0/ACODEC_ADC_SYNC/GPIO3_D4_d	GRF_GPIO3D_IOMUX_H[2:0]=3'b001
i2s0_sdi0	I	I2S0_SDI0_M0/PDM_SDI0_M0/ACODEC_DAC_DATAL/GPIO3_D6_d	GRF_GPIO3D_IOMUX_H[10:8]=3'b001
i2s0_sdo0	O	I2S0_SDO0_M0/ACODEC_DAC_DATAR/AUDPWM_R_M1/AUDDSM_LP/GPIO3_D5_d	GRF_GPIO3D_IOMUX_H[6:4]=3'b001
i2s0_sdi1/i2s0_sdo3	I/O	I2S0_SDO3_SDI1_M0/PDM_SDI1_M0/AUDPWM_R_M0/I2C4_SDA_M1/AUDDSM_RP/GPIO4_A1_d	GRF_GPIO4A_IOMUX_L[6:4]=3'b001
i2s0_sdi2/i2s0_sdo2	I/O	I2S0_SDO2_SDI2_M0/PDM_SDI2_M0/AUDPWM_L_M0/I2C4_SCL_M1/AUDDSM_RN/GPIO4_A0_d	GRF_GPIO4A_IOMUX_L[2:0]=3'b001
i2s0_sdi3/i2s0_sdo1	I/O	I2S0_SDO1_SDI3_M0/PDM_SDI3_M0/ACODEC_ADC_DATA/GPIO3_D7_d	GRF_GPIO3D_IOMUX_H[14:12]=3'b001

The following table shows the I2S0 group 1 interface description.

Table 17-2 I2S0 Group 1 Interface Description

Module Pin	Direction	Pad Name	IOMUX Setting
i2s0_mclk_tx or i2s0_mclk_rx	I/O	CIF_D4_M0/RGMII_RXD3_M0/I2S0_MCLK_M1/UART5_RTSN_M0/I2C5_SCL_M1/GPIO3_B0_d	GRF_GPIO3B_IOMUX_L[2:0]=3'b011
i2s0_sclk_tx	I/O	CIF_D0_M0/I2S0_SCLK_TX_M1/UART4_TX_M0/I2C3_SCL_M0/PWM8_M0/GPIO3_A4_d	GRF_GPIO3A_IOMUX_H[2:0]=3'b011
i2s0_sclk_rx	I/O	CIF_D5_M0/RGMII_TXD2_M0/I2S0_SCLK_RX_M1/UART5_CTSN_M0/I2C5_SDA_M1/GPIO3_B1_d	GRF_GPIO3B_IOMUX_L[6:4]=3'b011

Module Pin	Direction	Pad Name	IOMUX Setting
i2s0_lrck_tx	I/O	CIF_D1_M0/RGMII_CRS_M0/I2S0_LRCK_TX_M1/UART4_RX_M0/I2C3_SDA_M0/PWM9_M0/GPIO3_A5_d	GRF_GPIO3A_IOMUX_H[6:4]=3'b011
i2s0_lrck_rx	I/O	CIF_D6_M0/RGMII_TXD3_M0/I2S0_LRCK_RX_M1/UART4_RTSN_M0/GPIO3_B2_d	GRF_GPIO3B_IOMUX_L[10:8]=3'b011
i2s0_sdi0	I	CIF_D3_M0/RGMII_RXD2_M0/I2S0_SDI0_M1/UART5_RX_M0/CAN_TXD_M1/PWM11_IR_M0/GPIO3_A7_d	GRF_GPIO3A_IOMUX_H[14:12]=3'b011
i2s0_sdo0	O	CIF_D2_M0/RGMII_COL_M0/I2S0_SDO0_M1/UART5_TX_M0/CAN_RXD_M1/PWM10_M0/GPIO3_A6_d	GRF_GPIO3A_IOMUX_H[10:8]=3'b011
i2s0_sdi1/ i2s0_sdo3	I/O	CIF_D9_M0/RGMII_TXEN_M0/I2S0_SDO3_SDI1_M1/SPI1_CS0n_M0/GPIO3_B5_d	GRF_GPIO3B_IOMUX_H[6:4]=3'b011
i2s0_sdi2/ i2s0_sdo2	I/O	CIF_D8_M0/RGMII_TXD1_M0/I2S0_SDO2_SDI2_M1/SPI1_CS1n_M0/GPIO3_B4_d	GRF_GPIO3B_IOMUX_H[2:0]=3'b011
i2s0_sdi3/ i2s0_sdo1	I/O	CIF_D7_M0/RGMII_TXD0_M0/I2S0_SDO1_SDI3_M1/UART4_CTSN_M0/GPIO3_B3_d	GRF_GPIO3B_IOMUX_L[14:12]=3'b011

The I2S0 also communicates with the Digital Audio Codec inside the chip by configuring GRF_SOC_CON2[12]. If GRF_SOC_CON2[12] is configured to 1'b1, then I2S0 will communicate with Digital Audio Codec, otherwise the I2S0 will interact with other I2S devices through I2S0 group 0 or group 1 interface. By configuring GRF_IOFUNC_SEL0[9] to 1'b0, i2s0_mclk_tx will be connected to the PAD. If GRF_IOFUNC_SEL0[9] is configured to 1'b1, i2s0_mclk_rx will be connected to the PAD.

The I2S1 is connected to three groups of IO interfaces. The following table shows the I2S1 group 0 interface description.

Table 17-3 I2S1 Group 0 Interface Description

Module Pin	Direction	Pad Name	IOMUX Setting
i2s1_mclk	I/O	FLASH_CS0n/FSPI_CS0n /I2S1_MCLK_M0/GPIO0_D4_u	GRF_GPIO0D_IOMUX_H[2:0]=3'b100
i2s1_sclk	I/O	FLASH_RDYn/FSPI_D1/I2S1_SCLK_M0/GPIO1_A1_u	GRF_GPIO1A_IOMUX_L[6:4]=3'b100
i2s1_lrck_tx or i2s1_lrck_rx	I/O	FLASH_ALE/FSPI_D0/I2S1_LRCK_M0/GPIO1_A0_d	GRF_GPIO1A_IOMUX_L[2:0]=3'b100
i2s1_sdi	I	FLASH_RDn/FSPI_D3/I2S1_SDI_M0/GPIO1_A2_u	GRF_GPIO1A_IOMUX_L[10:8]=3'b100
i2s1_sdo	O	FSPI_D2/I2S1_SDO_M0/GPIO0_D6_d	GRF_GPIO0D_IOMUX_H[10:8]=3'b100

Table 17-4 I2S1 Group 1 Interface Description

Module Pin	Direction	Pad Name	IOMUX Setting
i2s1_mclk	I/O	SPI0_CS1n_M1/I2S1_MCLK_M1/UART4_TX_M2/GPIO1_D5_d	GRF_GPIO1D_IOMUX_H[6:4]=3'b010
i2s1_sclk	I/O	SPI0_MOSI_M1/I2S1_SCLK_M1/I2C3_SCL_M2/GPIO1_D6_d	GRF_GPIO1D_IOMUX_H[10:8]=3'b010
i2s1_lrck_tx or i2s1_lrck_rx	I/O	SPI0_MISO_M1/I2S1_LRCK_M1/I2C3_SDA_M2/GPIO1_D7_d	GRF_GPIO1D_IOMUX_H[14:12]=3'b010
i2s1_sdi	I	SPI0_CS0n_M1/I2S1_SDI_M1/UART5_TX_M2/GPIO2_A0_d	GRF_GPIO2A_IOMUX_L[2:0]=3'b010

Module Pin	Direction	Pad Name	IOMUX Setting
i2s1_sdo	O	SPI0_CLK_M1/I2S1_SDO_M1/UART5_RX_M2/GPIO2_A1_d	GRF_GPIO2A_IOMUX_L[6:4]=3'b010

Table 17-5 I2S1 Group 2 Interface Description

Module Pin	Direction	Pad Name	IOMUX Setting
i2s1_mclk	I/O	LCDC_D19/RGMII_RXD2_M1/CIF_D15_M1/I2S1_MCLK_M2/GPIO2_C7_d	GRF_GPIO2C_IOMUX_H[14:12]=3'b110
i2s1_sclk	I/O	LCDC_D21/RGMII_TXD2_M1/CIF_CLKOUT_M1/I2S1_SCLK_M2/GPIO2_D1_d	GRF_GPIO2D_IOMUX_L[6:4]=3'b110
i2s1_lrck_tx or i2s1_lrck_rx	I/O	LCDC_D22/RGMII_TXCLK_M1/CIF_CLKIN_M1/I2S1_LRCK_M2/GPIO2_D2_d	GRF_GPIO2D_IOMUX_L[10:8]=3'b110
i2s1_sdi	I	LCDC_D23/RGMII_RXCLK_M1/CIF_HSYNC_M1/I2S1_SDI_M2/GPIO2_D3_d	GRF_GPIO2D_IOMUX_L[14:12]=3'b110
i2s1_sdo	O	LCDC_D20/RGMII_RXD3_M1/CIF_VSYNC_M1/I2S1_SDO_M2/GPIO2_D0_d	GRF_GPIO2D_IOMUX_L[2:0]=3'b110

By configuring GRF_IOFUNC_SEL0[10] to 1'b0, i2s1_lrck_rx will be connected to the PAD. If GRF_IOFUNC_SEL0[10] is configured to 1'b1, i2s1_lrck_tx will be connected to the PAD.
The I2S2 is connected to two groups of IO interfaces. The following table shows the I2S1 group 0 interface description.

Table 17-6 I2S2 Group 0Interface Description

Module Pin	Direction	Pad Name	IOMUX Setting
i2s2_mclk	I/O	I2S2_MCLK_M0/SDMMC1_DET/SPI1_CS1n_M1/I2C5_SCL_M2/UART1_TX_M1/GPIO1_D0_d	GRF_GPIO1D_IOMUX_L[2:0]=3'b001
i2s2_sclk	I/O	I2S2_SCLK_M0/SPI1_CLK_M1/PRELIGHT_TRIG_OUT/UART1_RTSn_M1/ GPIO1_C6_d	GRF_GPIO1C_IOMUX_H[10:8]=3'b001
i2s2_lrck_tx or i2s2_lrck_rx	I/O	I2S2_LRCK_M0/SPI1_CS0n_M1/UART1_CTSn_M1/GPIO1_C7_d	GRF_GPIO1C_IOMUX_H[14:12]=3'b001
i2s2_sdi	I	I2S2_SDI_M0/SPI1_MISO_M1/FLASH_TRIG_IN/GPIO1_C5_d	GRF_GPIO1C_IOMUX_H[6:4]=3'b001
i2s2_sdo	O	I2S2_SDO_M0/SPI1_MOSI_M1/FLASH_TRIG_OUT/GPIO1_C4_d	GRF_GPIO1C_IOMUX_H[2:0]=3'b001

Table 17-7 I2S2 Group 1Interface Description

Module Pin	Direction	Pad Name	IOMUX Setting
i2s2_mclk	I/O	LCDC_D7/I2S2_MCLK_M1/CIF_D3_M1/UART5_CTSN_M1/SPI0_CS1n_M2/PWM0_M1/I2C5_SDA_M0/ GPIO2_B3_d	GRF_GPIO2B_IOMUX_L[14:12]=3'b010
i2s2_sclk	I/O	LCDC_D5/I2S2_SCLK_M1/UART5_RX_M1/ PWM2_M1/SPI0_MISO_M2/GPIO2_B1_d	GRF_GPIO2B_IOMUX_L[6:4]=3'b010
i2s2_lrck_tx or i2s2_lrck_rx	I/O	LCDC_D6/I2S2_LRCK_M1/UART5_RTSN_M1/PWM1_M1/SPI0_CLK_M2/GPIO2_B2_d	GRF_GPIO2B_IOMUX_L[10:8]=3'b010
i2s2_sdi	I	LCDC_D4/I2S2_SDI_M1/UART5_TX_M1/PWM3_IR_M1/SPI0_MOSI_M2/GPIO2_B0_d	GRF_GPIO2B_IOMUX_L[2:0]=3'b010
i2s2_sdo	O	LCDC_D3/I2S2_SDO_M1/UART4_RX_M1/ PWM4_M1/SPI0_CS0n_M2/GPIO2_A7_d	GRF_GPIO2A_IOMUX_H[14:12]=3'b010

By configuring GRF_IOFUNC_SEL0[11] to 1'b0, i2s2_lrck_rx will be connected to the PAD. If GRF_IOFUNC_SEL0[11] is configured to 1'b1, i2s2_lrck_tx will be connected to the PAD.

17.6 Application Notes

17.6.1 Software Application Notes

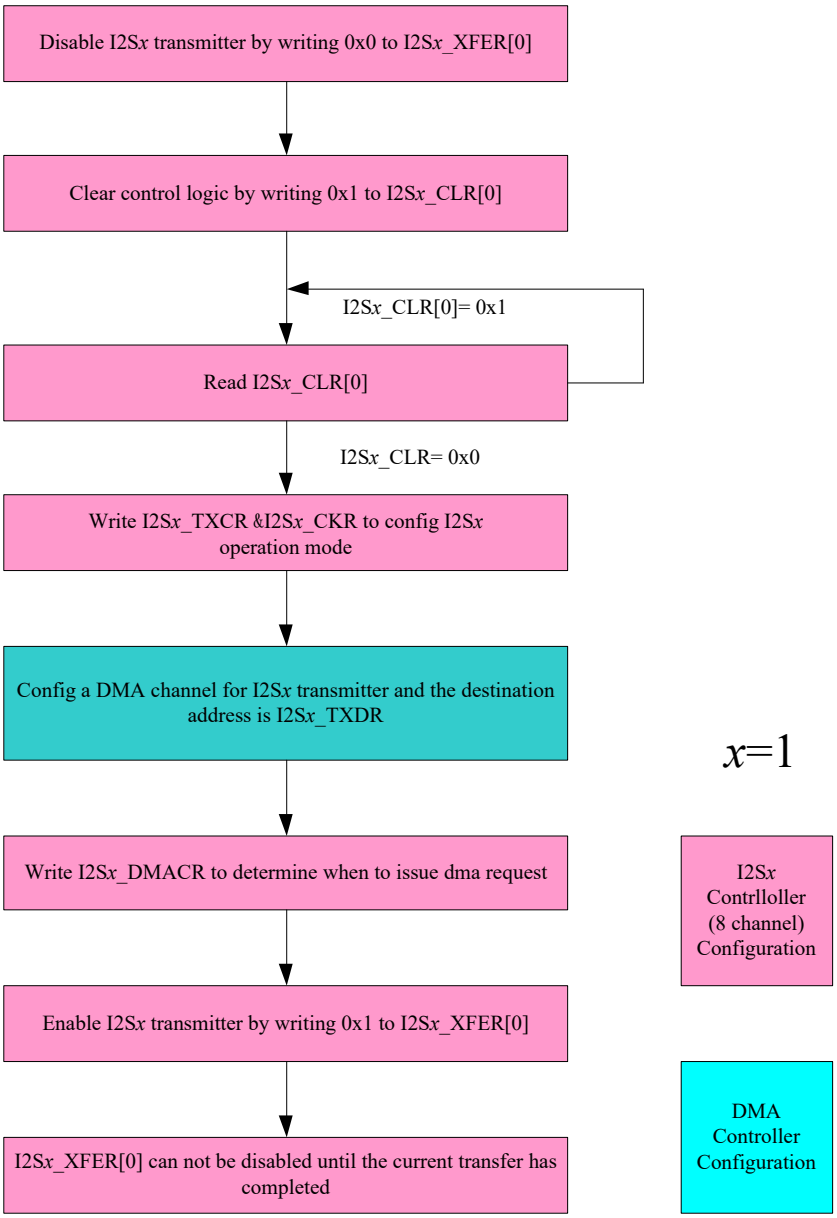


Fig.17-20 I2S/PCM/TDM Controller Transmit Operation Flow Chart

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

Chapter 18 Digital Audio Codec

18.1 Overview

Digital Audio Codec is a 24-bit digital audio encoder/decoder which supports multiple sample rates. It is mainly composed of digital ADC and digital DAC. The function of digital ADC is to convert pulse density modulation format data into PCM format audio data through a series of filters and volume control. Decoding result of digital ADC is sent out through I2S/PCM interface. The aim of digital DAC is to process the data received from I2S/PCM interface through filters, volume control and modulation. Finally the modulated result is sent to Analog Codec.

The Digital Audio Codec supports the following features.

- Support 8-bit APB bus slave interface
- Support 3-channel digital ADC
- Support 2-channel digital DAC
- Support I2S/PCM interface
- Support I2S/PCM master and slave mode
- Support 4-channel audio transmitting in I2S mode
- Support 2-channel audio receiving in I2S mode
- Support 2-channel audio transmitting or receiving in PCM mode
- Support I2S normal, left and right justified mode serial audio data transfer
- Support PCM early, late1, late2, late3 mode serial audio data transfer
- Support MSB or LSB first serial audio data transfer
- Support configurable SCLK and LRCK polarity
- Support 16~24 bit sample resolution for both digital ADC and digital DAC
- Support programmable left and right channel exchangeable in I2S mode and PCM mode for both digital ADC and digital DAC
- Support three modes of mixing for every digital DAC channel
- Both digital ADC and digital DAC support three groups of sample rates. Group 0 are 8kHz/16kHz/32kHz/64kHz/128kHz, group 1 are 11.025kHz/22.05kHz/44.1kHz/88.2kHz/176.4kHz and group 2 are 12kHz/24kHz/48kHz/96kHz/192kHz
- Support asynchronous mode and synchronous mode
- In asynchronous mode, there are no constraints on the sample rate of digital ADC and digital DAC. They are completely independent
- In synchronous mode, support digital ADC or digital DAC operating individually. The sample rate of digital ADC and digital DAC can be within any groups and any kind within one group
- In synchronous mode, support digital ADC and digital DAC operating at the same time within same sample rate group. Sample rates of digital ADC and digital DAC are within the same group
- In synchronous mode, support digital ADC and digital DAC operating at the same time within different sample rate group. Sample rate of digital ADC is within group 0 and digital DAC is within group 2 or digital ADC is within group 2 and digital DAC is within group 0, group 1 not supported
- The pass-band of digital ADC filters is 0.45625*fs
- Support digital ADC pass-band ripple within +/-0.1dB
- The stop-band of digital ADC filters is 0.5*fs
- Support digital ADC stop-band attenuation at least 60dB
- Support volume control for both digital ADC and digital DAC
- Support Automatic Level Control(ALC)and noise gate for digital ADC
- Support programmable negative and positive volume gain for both digital ADC and digital DAC
- Support communication with Analog Codec through I2C bus

18.2 Block Diagram

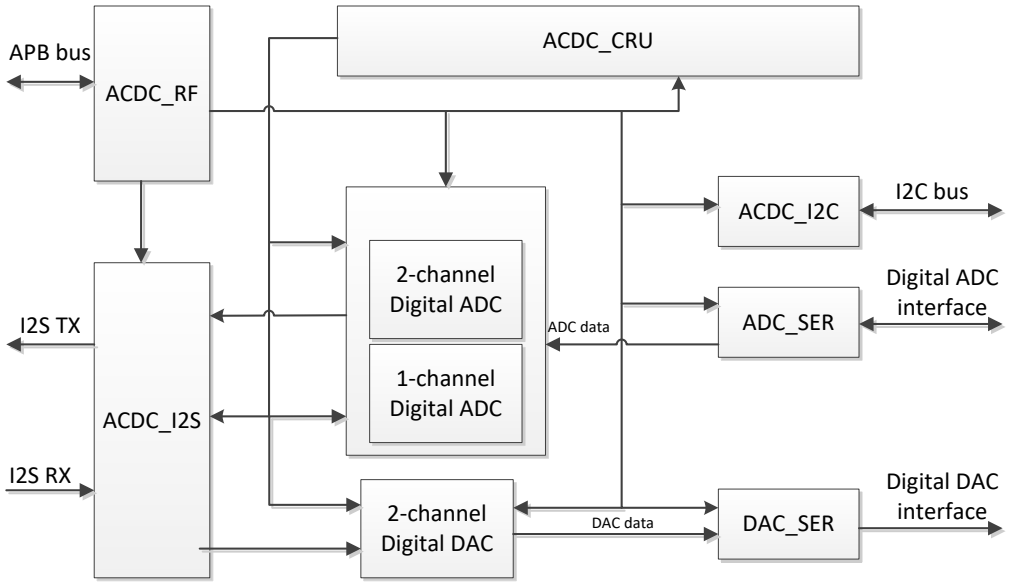


Fig.18-1 Digital Audio Codec Block Diagram

ACDC_RF

The ACDC_RF implements the 8-bit APB slave operation through APB bus. It is responsible for configuring the operation registers of other modules.

ACDC_CRU

The ACDC_CRU implements clock and reset generation function. It is responsible for generating sample clock, digital ADC/DAC operation clock and I2S operation clock.

Digital ADC

There are 3 digital ADC channels in total inside Digital Audio Codec. The 2-channel digital ADC is composed of left channel and right channel while the 1-channel digital ADC contains only one channel. The digital ADC receives three channels data from ADC_SER module. It includes CIC filters, compensation filters, low-pass decimation filters, high-pass filters and other audio signal processing related modules. The output of digital ADC is sent to the I2S module.

ADC_SER

This module receives serial data from analog ADC. It extracts three channels from input serial data and distributes them to the corresponding digital ADC channel.

Digital DAC

There are 2 digital DAC channels inside the Audio Codec. The digital DAC receives audio data from I2S RX. It includes one CIC filter, one high-pass filter, several low-pass decimation filters, modulator and other audio signal processing related modules. The output of digital DAC is sent to DAC_SER module and serialized before transmitting to analog DAC.

DAC_SER

This module receives parallel data from digital DAC. It then serializes the parallel data and sends it to analog DAC.

ACDC_I2S

The I2S/PCM audio interface can be configured to master mode or slave mode. In Master Mode, SCLK and LRCK are configured as output. In Slave Mode, SCLK and LRCK are configured as input. The ACDC_I2S module can operate in TX or RX mode. When in TX mode, it receives audio data from digital ADC and sends it out through I2S TX interface. When in RX mode, it receives audio data from I2S RX interface and sends it to digital DAC.

ACDC_I2C

The ACDC_I2C module is used to communicate with analog ADC/DAC. It is responsible for sending 4-bit PGA gain codes for each channel to analog ADC/DAC every interval of time.

18.3 Function description

The I2S/PCM interface of Digital Audio Codec is connected to the I2S0 controller. Please

refer to the I2S chapter for detailed information about I2S and PCM format that Digital Audio Codec supports.

18.3.1 Filters of Digital ADC

Digital Audio Codec receives 3 channels data such as DATA0, DATA1 and DATA2 from Analog Codec. DATA0 is connected to the left channel of 2-channel digital ADC, DATA1 is connected to the right channel of 2-channel digital ADC and DATA2 is connected to the 1-channel digital ADC.

In order to achieve PCM format audio data from DATA0, DATA1 and DATA2, a total of 8 filters for 2-channel digital ADC and 8 filters for 1-channel digital ADC are embedded. It includes a CIC decimation filter, a compensation filter, 4 half-band filters, a low-pass filter and a high-pass filter in 2-channel digital ADC or 1-channel digital ADC.

The CIC decimation filter achieves maximum 16-times decimation when in normal mode. It also can be programmed to 8-times in low power mode 1 or 4-times decimation in low power mode 2. When operates in low power mode 1 or 2, the operating clock of digital ADC can be reduced to half or quarter of the normal mode. The problem is that signal indicators will become worse and some sample rates not support in low power mode 1 or 2. So make sure that the CIC decimation filter works in 16-times decimation unless you don't care about signal indicators.

Compensation filter is connected in series following CIC filter. Its function is to reduce the ripple of CIC filter output. It can be software enabled or disabled.

The 4 half-band filters and a low-pass filter each perform 2-times decimation. That means a maximum of 32-times decimation can be achieved. Not all of these 5 decimation filters are working all the time. How many of them are needed to work depends on the sample rate. For example, in order to output a 192K sample rate signal, only one filter works, the rest of filters are idle.

The high-pass filter is used to filter DC components in audio data stream. Result of high-pass filter is sent to the volume control.

The equivalent parameters of digital ADC filters are as follows.

Table 18-1 Equivalent Parameters of Digital ADC Filters

Parameter	Test condition	Min	Typ	Max	Unit
Pass-band	+/- 0.1dB	20	N/A	0.45625fs	Hz
Pass-band ripple	N/A	N/A	N/A	+/- 0.1	dB
Stop-band	N/A	0.5fs	N/A	N/A	Hz
Stop-band attenuation	f>0.5fs	60	N/A	N/A	dB

18.3.2 Filters of Digital DAC

I2S module receives two-channel audio data from I2S RX interface and drives it to the digital DAC which supports mixing function. How to pour 2-channel audio data into 2-channel digital DAC can be achieved by programming mixing mode.

Audio processing of digital DAC is similar to that of digital ADC, which is almost the inverse process of digital ADC. The 2-channel digital DAC includes a high-pass filter and 5 half-band filters. The high-pass filter is used to filter DC components in audio data stream. Result of high-pass filter is sent to the digital DAC volume control module. The input of 5 half-band filters comes from output of volume control, each perform 2-times interpolation. But not all of them are working all the time. How many of them are needed to work depends on the sample rate of digital DAC. The result of half-band filters is sent to a third-order Sigma-Delta modulator.

18.3.3 Volume Control

PCM format data output from 3-channel digital ADC high-pass filters are fed into volume control module. The volume control module inside digital ADC contains several sub-modules such as noise gate, peak detect, frequency cross zero detect, ALC and digital gain control. It can be digitally attenuated over a range of -96dB~0dB in 0.375dB/step for negative gain and amplified over a range of 0dB~96dB in 0.375dB/step for positive gain. Whether is attenuated or amplified can be software programmed. The volume of each channel can be

controlled separately. Each channel has an 8-bit register for volume control. For negative gain, 0xff corresponds to digital mute while 0x00 corresponds to 0dB. For positive gain, 0xff corresponds to maximum gain while 0x00 corresponds to 0dB.
As for digital DAC, output of high-pass filter is fed into volume control module. Similar to digital ADC, The volume control module inside digital DAC contains several sub-modules such as peak detect, frequency cross zero detect, LIMITER and digital gain control. It also can be digitally attenuated over a range of -96dB~0dB in 0.375dB/step for negative gain and amplified over a range of 0dB~96dB in 0.375dB/step for positive gain. Whether is attenuated or amplified can be software programmed.

18.3.4 I2C Interface

The role of I2C inside the Digital Audio Codec is to send 4-bit PGA gain codes for each digital ADC channel to Analog Codec every interval of time. The interval of time is software programmable and the unit is the sample rate of digital ADC. There are 12 bits PGA gain codes in total for 3-channeldigital ADC, requiring the I2C inside Digital Audio Codec to initiate a request to transmit two bytes when interval of time reaches. The I2C inside Digital Audio Codec can be enabled or disabled by software. Its output bus will multiplex with RKI2C2 controller.

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.

18.4.2 Registers Summary

Name	Offset	Size	Reset Value	Description
ACDCDIG_SYSCTRL0	0x0000	W	0x00000000	System Control Register
ACDCDIG_ADCVUCTL	0x0040	W	0x00000001	ADC Volume Control Register
ACDCDIG_ADCVUCTIME	0x0044	W	0x00000000	ADC Volume Control Time Limit Register
ACDCDIG_ADCDIGEN	0x0048	W	0x00000000	ADC Digital Enable Register
ACDCDIG_ADCCLKCTRL	0x004c	W	0x00000000	ADC Clock Control Register
ACDCDIG_ADCINT_DIV	0x0054	W	0x00000007	ADC Integer Clock Divider Register
ACDCDIG_ADCSCLKTXINT_DIV	0x006c	W	0x0000001f	I2S SCLK TX Integer Divider Register
ACDCDIG_ADCCFG1	0x0084	W	0x00000004	ADC Configure Register 1
ACDCDIG_ADCVOLL0	0x0088	W	0x00000000	Volume of ADC Left Channel 0 Register
ACDCDIG_ADCVOLL1	0x008c	W	0x00000000	Volume of ADC Left Channel 2 Register
ACDCDIG_ADCVOLR0	0x0098	W	0x00000000	Volume of ADC right Channel 1 Register
ACDCDIG_ADCVOGP	0x00a8	W	0x00000000	ADC Volume Gain Polarity Register
ACDCDIG_ADCRVOLL0	0x00ac	W	0x000000ff	Internal Volume of ADC Left Channel 0 Register
ACDCDIG_ADCRVOLL1	0x00b0	W	0x000000ff	Internal Volume of ADC Left Channel 2 Register
ACDCDIG_ADCRVOLR0	0x00bc	W	0x000000ff	Internal Volume of ADC right Channel 1 Register
ACDCDIG_ADCALC0	0x00cc	W	0x00000000	Automatic Level Control Register 0
ACDCDIG_ADCALC1	0x00d0	W	0x00000000	Automatic Level Control Register 1

Name	Offset	Size	Reset Value	Description
ACDCDIG_ADCALC2	0x00d4	W	0x00000000	Automatic Level Control Register 2
ACDCDIG_ADCNG	0x00d8	W	0x00000000	ADC Noise Gate Control Register
ACDCDIG_ADCNGST	0x00dc	W	0x00000000	ADC Noise Gate Status Register
ACDCDIG_ADCHPFEN	0x00e0	W	0x00000000	ADC High-pass Filter Enable Register
ACDCDIG_ADCHPFCF	0x00e4	W	0x00000000	ADC High-pass Control Register
ACDCDIG_ADCPGL0	0x00ec	W	0x00000000	PGA Gain of Left Channel 0
ACDCDIG_ADCPGL1	0x00f0	W	0x00000000	PGA Gain of Left Channel 2
ACDCDIG_ADCPGR0	0x00fc	W	0x00000000	PGA Gain of right Channel 1
ACDCDIG_ADCLILMT1	0x010c	W	0x00000000	PGA Gain Limiter Register 1
ACDCDIG_ADCLILMT2	0x0110	W	0x00000000	PGA Gain Limiter Register 2
ACDCDIG_ADCDMICNG1	0x0114	W	0x00000000	Limiter Noise Gate Register 1
ACDCDIG_ADCDMICNG2	0x0118	W	0x00000000	Limiter Noise Gate Register 2
ACDCDIG_DACVUCTL	0x0140	W	0x00000001	DAC Volume Control Register
ACDCDIG_DACVUCTIME	0x0144	W	0x00000000	DAC Volume Control Time Limit Register
ACDCDIG_DACDIGEN	0x0148	W	0x00000000	DAC Digital Enable Register
ACDCDIG_DACCLKCTRL	0x014c	W	0x00000000	DAC Clock Control Register
ACDCDIG_DACINT_DIV	0x0154	W	0x00000007	DAC Integer Clock Divider Register
ACDCDIG_DACSCLKRXINT_DIV	0x0160	W	0x0000001f	I2S SCLK RX Integer Divider Register
ACDCDIG_DACPWM_DIV	0x0164	W	0x00000003	PWM Mode Integer Divider Register
ACDCDIG_DACPWM_CTRL	0x0168	W	0x00000000	PWM Mode Control Register
ACDCDIG_DACCFG1	0x0184	W	0x00000004	DAC Configure Register 1
ACDCDIG_DACMUTE	0x0188	W	0x00000000	DAC Mute Control Register
ACDCDIG_DACMUTEST	0x018c	W	0x00000001	DAC Mute Status Register
ACDCDIG_DACVOLL0	0x0190	W	0x00000000	Volume of DAC Left Channel 0 Register
ACDCDIG_DACVOLR0	0x01a0	W	0x00000000	Volume of DAC right Channel 1 Register
ACDCDIG_DACVOGP	0x01b0	W	0x00000000	ADC Volume Gain Polarity Register
ACDCDIG_DACRVOLL0	0x01b4	W	0x000000ff	Internal Volume of DAC Left Channel 0 Register
ACDCDIG_DACRVOLR0	0x01c4	W	0x000000ff	Internal Volume of DAC right Channel 1 Register
ACDCDIG_DACLMT0	0x01d4	W	0x00000000	DAC Limiter Register 0
ACDCDIG_DACLMT1	0x01d8	W	0x00000000	DAC Limiter Register 1
ACDCDIG_DACLMT2	0x01dc	W	0x00000000	DAC Limiter Register 2
ACDCDIG_DACMIXCTRLL	0x01e0	W	0x00000000	DAC Mixing Control Register Of Left Channels
ACDCDIG_DACMIXCTRLR	0x01e4	W	0x00000000	DAC Mixing Control Register Of right Channels
ACDCDIG_DACHPF	0x01e8	W	0x00000000	DAC High-pass Filter Control Register
ACDCDIG_I2C_FLT_CON0	0x0280	W	0x00000000	I2C Filter Control Register 0
ACDCDIG_I2C_FLT_CON1	0x0284	W	0x0000000f	I2C Filter Control Register 1
ACDCDIG_I2C_CON0	0x0288	W	0x00000000	I2C Control Register 0
ACDCDIG_I2C_CON1	0x028c	W	0x00000003	I2C Control Register 1

Name	Offset	Size	Reset Value	Description
ACDCDIG_I2C_CLKDIVL0	0x0290	W	0x00000006	I2C Scl Low Level Divider Register 0
ACDCDIG_I2C_CLKDIVL1	0x0294	W	0x00000000	I2C Scl Low Level Divider Register 1
ACDCDIG_I2C_CLKDIVH0	0x0298	W	0x00000006	I2C Scl High Level Divider Register 0
ACDCDIG_I2C_CLKDIVH1	0x029c	W	0x00000000	I2C Scl High Level Divider Register 1
ACDCDIG_I2C_MAXCNT	0x02a0	W	0x00000080	I2C Master Transmit Count Register.
ACDCDIG_I2C_SCLOE_DB0	0x02a4	W	0x00000020	Slave Hold Debounce Configure Register 0
ACDCDIG_I2C_SCLOE_DB1	0x02a8	W	0x00000000	Slave Hold Debounce Configure Register 1
ACDCDIG_I2C_SCLOE_DB2	0x02ac	W	0x00000000	Slave Hold Debounce Configure Register 2
ACDCDIG_I2C_SCLOE_DB3	0x02b0	W	0x00000000	Slave Hold Debounce Configure Register 3
ACDCDIG_I2C_TMOUTL	0x02b4	W	0x0000001f	I2C Transmit Request Time Out Register 0
ACDCDIG_I2C_TMOUTH	0x02b8	W	0x00000000	I2C Transmit Request Time Out Register 1
ACDCDIG_I2C_DEV_ADDR	0x02bc	W	0x00000048	I2C Slave Device Address Register
ACDCDIG_I2C_REG_ADDR	0x02c0	W	0x00000028	Register Address of I2C Slave
ACDCDIG_I2C_STATUS	0x02c4	W	0x00000000	I2C Status Register
ACDCDIG_I2S_TXCR0	0x0300	W	0x0000000f	Transmit Operation Control Register 0
ACDCDIG_I2S_TXCR1	0x0304	W	0x00000000	Transmit Operation Control Register 1
ACDCDIG_I2S_TXCR2	0x0308	W	0x00000000	Transmit Operation Control Register 2
ACDCDIG_I2S_RXCR0	0x030c	W	0x0000000f	Receive Operation Control Register 0
ACDCDIG_I2S_RXCR1	0x0310	W	0x00000000	Receive Operation Control Register 1
ACDCDIG_I2S_CKR0	0x0314	W	0x00000000	Clock Generation Register 0
ACDCDIG_I2S_CKR1	0x0318	W	0x00000000	Clock Generation Register 1
ACDCDIG_I2S_XFER	0x031c	W	0x00000000	Transfer Start Register
ACDCDIG_I2S_CLR	0x0320	W	0x00000000	SCLK Domain Logic Clear Register
ACDCDIG_VERSION	0x0380	W	0x00000002	Version Register

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

18.4.3 Detail Registers Description

ACDCDIG_SYSCTRL0

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:6	RO	0x0000000	reserved
5	RW	0x0	sync_mode 1'b0: Asynchronous mode. Input operation clocks of Digital ADC and Digital ADC are asynchronous to each other. 1'b1: Synchronous mode. Digital ADC and Digital ADC internally share one common clock.

Bit	Attr	Reset Value	Description
4	RW	0x0	clk_com_sel Selects the operation clock of Digital ADC or Digital DAC as common clock when Digital ADC and Digital DAC work in synchronous mode. keep the default value when Digital ADC and Digital DAC work in asynchronous mode. 1'b0: Selects operation clock of Digital ADC 1'b1: Selects operation clock of Digital DAC
3	RW	0x0	glbcke Global enable for synchronization signal of both Digital ADC and DAC. It works well when Digital ADC and DAC operate in synchronous mode where the Digital ADC and DAC share the same d2a_adc_clk and d2a_adc_sync to communicate with Analog ADC/DAC. If Digital ADC and DAC operate in asynchronous mode meaning that they are independent, glbcke is ineffective. 1'b0: Disable 1'b1: Enable
2	RO	0x0	reserved
1	RW	0x0	sync_sel Selects the synchronization signal generated by Digital ADC or by Digital DAC that is connected to d2a_adc_sync to communicate with Analogy ADC/DAC. Make sure that sync_sel is set to 1'b0 when in synchronous mode. If in asynchronous mode, sync_sel should be set according to the following illustration. 1'b0: Synchronization signal generated by Digital ADC is connected to d2a_adc_sync. 1'b1: Synchronization signal generated by Digital DAC is connected to d2a_adc_sync.

ACDCDIG_ADCVUCTL

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:3	RO	0x00000000	reserved
2	RW	0x0	adc_byps Digital ADC volume control bypass. 1'b0: ADC volume control enable 1'b1: ADC volume control bypass
1	RW	0x0	adcfade Digital ADC volume adjust mode. 1'b0: update to new volume immediately 1'b1: update volume as adczdt field describes
0	RW	0x1	adczdt Digital ADC volume cross zero detect enable. It works when adc_byps is 1'b0 and adcfade is 1'b1. 1'b0: Volume adjusts every sample. 1'b1: Volume adjusts only when audio waveform crosses zero or volume-control time-limit condition meets.

ACDCDIG_ADCVUCTIME

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved

Bit	Attr	Reset Value	Description
7:0	RW	0x00	adcvuct Volume control time limit, valid only in fade cross zero mode. Time limit = adcvuct*(1/sample rate) Unit: sample rate

ACDCDIG_ADCDIGEN

Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31:5	RO	0x0000000	reserved
4	RW	0x0	adcglben Global enable of all Digital ADC channels. Only when adcglben and the enable signal corresponding to each Digital ADC channel is valid before starting work. 1'b0: Disable 1'b1: Enable
3:2	RO	0x0	reserved
1	RW	0x0	adcen_l2 Digital ADC left channel 2 enable. 1'b0: Disable 1'b1: Enable
0	RW	0x0	adcen_l0r1 Digital ADC left channel 0 and right channel 1 enable. 1'b0: Disable 1'b1: Enable

ACDCDIG_ADCCLKCTRL

Address: Operational Base + offset (0x004c)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:6	RW	0x0	cic_ds_ratio CIC filter decimation ratio. 2'b00: 16 times decimation 2'b01: 8 times decimation other: 4 times decimation
5	RW	0x0	adc_cke Digital ADC operation clock enable. 1'b0: Disable 1'b1: Enable
4	RW	0x0	i2stx_cke Clock enable of internal I2S TX channel. 1'b0: Disable 1'b1: Enable
3	RW	0x0	cke_bclktx Clock enable of sclk_out_tx. 1'b0: Disable 1'b1: Enable
2	RW	0x0	filter_gate_en Filter gate enable. There are some filters in Digital ADC, they work depends on the sample rate. If any 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. 1'b0: Don't gate filters' clock 1'b1: Gate filters' clock

Bit	Attr	Reset Value	Description
1	RW	0x0	adc_sync_ena Enable of the synchronization signal generated by Digital ADC. Note that only when both glbcke and adc_sync_ena are equal to 1'b1, the synchronization signal of Digital ADC can be generated. 1'b0: Disable 1'b1: Enable
0	RW	0x0	adc_sync_status There is a counter to generate synchronization signal of Digital ADC. In order to ensure the integrity of synchronization signal, it is necessary to read back adc_sync_status to judge whether the counter stops working when adc_sync_ena is set from 1'b1 to 1'b0. If the signal is read back to 1'b0, it means that the counter stops working and the synchronization signal of Digital ADC is complete.

ACDCDIG_ADCINT_DIV

Address: Operational Base + offset (0x0054)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x07	int_div_con Interger clock divider to provide 6.144/5.644/4.096MHz sample clock for internal filters. Make sure that int_div_con is an odd number between 7(8 times division) and 15(16 times division).

ACDCDIG_ADCSCLKTXINT_DIV

Address: Operational Base + offset (0x006c)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x1f	scktxdiv Integer clock divider to generate sclk_out_tx when I2S TX works in master mode. It is ignored when in slave mode.

ACDCDIG_ADCCFG1

Address: Operational Base + offset (0x0084)

Bit	Attr	Reset Value	Description
31:5	RO	0x0000000	reserved
4:2	RW	0x1	adcsrt Sample rates of Digital ADC. 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
1	RW	0x0	sig_scale_mode Signal scale mode select. 1'b0: CIC output the normal latitude. 1'b1: Scale the CIC output to half of the normal latitude and scale 2 times after high-pass filter.
0	RW	0x0	fir_com_bps FIR compensate filter bypass control. 1'b0: Not bypass 1'b1: Bypass

ACDCDIG_ADCVOLL0

Address: Operational Base + offset (0x0088)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x00	adclv0 Volume of Digital ADC left channel 0. 0db~-95.625db, 0.375db/step. 8'h0: 0db 8'h1: -0.375db 8'h2: -0.75db 8'h3: -1.125db ..... 8'hff: -95.625db

ACDCDIG_ADCVOLL1

Address: Operational Base + offset (0x008c)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x00	adclv1 Volume of Digital ADC left channel 2. 0db~-95.625db, 0.375db/step. 8'h0: 0db 8'h1: -0.375db 8'h2: -0.75db 8'h3: -1.125db ..... 8'hff: -95.625db

ACDCDIG_ADCVOLR0

Address: Operational Base + offset (0x0098)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x00	adcrv0 Volume of Digital ADC right channel 1. 0db~-95.625db, 0.375db/step. 8'h0: 0db 8'h1: -0.375db 8'h2: -0.75db 8'h3: -1.125db ..... 8'hff: -95.625db

ACDCDIG_ADCVOGP

Address: Operational Base + offset (0x00a8)

Bit	Attr	Reset Value	Description
31:3	RO	0x00000000	reserved
2	RW	0x0	volgpl2 Gain polarity for the volume of Digital ADC left channel 2. 1'b0: Negative gain 1'b1: Positive gain
1	RW	0x0	volgpr1 Gain polarity for the volume of Digital ADC right channel 1. 1'b0: Negative gain 1'b1: Positive gain

Bit	Attr	Reset Value	Description
0	RW	0x0	volgpl0 Gain polarity for the volume of Digital ADC left channel 0. 1'b0: Negative gain 1'b1: Positive gain

ACDCDIG_ADCRVOLL0

Address: Operational Base + offset (0x00ac)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RO	0xff	rvoll0 Internal real-time volume of Digital ADC left channel 0.

ACDCDIG_ADCRVOLL1

Address: Operational Base + offset (0x00b0)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0xff	rvoll1 Internal real-time volume of Digital ADC left channel 2.

ACDCDIG_ADCRVOLR0

Address: Operational Base + offset (0x00bc)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RO	0xff	rvolr0 Internal real-time volume of Digital ADC right channel 1.

ACDCDIG_ADCALC0

Address: Operational Base + offset (0x00cc)

Bit	Attr	Reset Value	Description
31:3	RO	0x00000000	reserved
2	RW	0x0	alcl2 Automatic level control enable for Digital ADC left channel 2. 1'b0: Disable 1'b1: Enable
1	RW	0x0	alcr1 Automatic level control enable for Digital ADC right channel 1. 1'b0: Disable 1'b1: Enable
0	RW	0x0	alcl0 Automatic level control enable for Digital ADC left channel 0. 1'b0: Disable 1'b1: Enable

ACDCDIG_ADCALC1

Address: Operational Base + offset (0x00d0)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:4	RW	0x0	alcarate ALC attack rate=sample rate/(8*power(2,alcarate)).
3:0	RW	0x0	alcrrate ALC release rate=sample rate/(8*power(2,alcrrate)).

ACDCDIG_ADCALC2

Address: Operational Base + offset (0x00d4)

Bit	Attr	Reset Value	Description
31:7	RO	0x0000000	reserved
6:4	RW	0x0	alcmax The highest threshold of ALC. 3'b000~3'b100: 0db~-12db, 3db/step 3'b101~3'b111: -18db~-30db, 6db/step
3	RO	0x0	reserved
2:0	RW	0x0	alcmin The lowest threshold of ALC. 3'b000~3'b100: 0db~-12db, 3db/step 3'b101~3'b111: -18db~-30db, 6db/step

ACDCDIG_ADCNG

Address: Operational Base + offset (0x00d8)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7	RW	0x0	ngchl Noise gate channel. 1'b0: Individual channel(or) 1'b1: Both channel(and)
6	RW	0x0	ngen Noise gate enable. 1'b0: Noise gate disable 1'b1: Noise gate enable
5	RW	0x0	ngboost Noise gate boost. 1'b0: Normal noise gate 1'b1: Boost noise gate
4:2	RW	0x0	nggate Noise gate threshold. If ngboost is 1'b0: 3'b000~3'b111: -63db~-84db, 3db/step If ngboost is 1'b1: 3'b000~3'b111: -33db~-54db, 3db/step
1:0	RW	0x0	ngdly Noise gate delay. The delay time before the noise gate attacks. 2'b00: 2048 samples 2'b01: 4096 samples 2'b10: 8192 samples 2'b11: 16384 samples

ACDCDIG_ADCNGST

Address: Operational Base + offset (0x00dc)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved
1	RO	0x0	ngstl2 Noise gates valid status of left channel 2. 1'b0: Not in NG status 1'b1: Now in NG status
0	RO	0x0	ngstl0r1 Noise gates valid status of left channel 0 and right channel 1. 1'b0: Not in NG status 1'b1: Now in NG status

ACDCDIG_ADCHPFEN

Address: Operational Base + offset (0x00e0)

Bit	Attr	Reset Value	Description
31:3	RO	0x00000000	reserved
2	RW	0x0	hpfen_l2 High-pass filter enable for left channel 2. 1'b0: High-pass filter is disabled. 1'b1: High-pass filter is enabled.
1	RW	0x0	hpfen_r1 High-pass filter enable for right channel 1. 1'b0: High-pass filter is disabled. 1'b1: High-pass filter is enabled.
0	RW	0x0	hpfen_l0 High-pass filter enable for left channel 0. 1'b0: High-pass filter is disabled. 1'b1: High-pass filter is enabled.

ACDCDIG_ADCHPFCF

Address: Operational Base + offset (0x00e4)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved
1:0	RW	0x0	hpfcf High-pass filter control. 2'b00: 3.79Hz 2'b01: 60Hz 2'b10: 243Hz 2'b11: 493Hz

ACDCDIG_ADCPGL0

Address: Operational Base + offset (0x00ec)

Bit	Attr	Reset Value	Description
31:4	RO	0x0000000	reserved
3:0	RW	0x0	pga_l0 PGA gain of left channel 0 for analogy ADC. Minimal gain: -18dB, maximum gain: 27dB, step: 3dB. The minimal gain corresponds to 4'b0000 and the maximum gain corresponds to 4'b1111.

ACDCDIG_ADCPGL1

Address: Operational Base + offset (0x00f0)

Bit	Attr	Reset Value	Description
31:4	RO	0x0000000	reserved
3:0	RW	0x0	pga_l1 PGA gain of left channel 2 for analogy ADC. Minimal gain: -18dB, maximum gain: 27dB, step: 3dB. The minimal gain corresponds to 4'b0000 and the maximum gain corresponds to 4'b1111.

ACDCDIG_ADCPGR0

Address: Operational Base + offset (0x00fc)

Bit	Attr	Reset Value	Description
31:4	RO	0x0000000	reserved
3:0	RW	0x0	pga_r0 PGA gain of right channel 1 for analogy ADC. Minimal gain: -18dB, maximum gain: 27dB, step: 3dB. The minimal gain corresponds to 4'b0000 and the maximum gain corresponds to 4'b1111.

ACDCDIG_ADCLILMT1

Address: Operational Base + offset (0x010c)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7	RW	0x0	lmt_en PGA gain LIMITER enable. 1'b0: Disable 1'b1: Enable
6:4	RW	0x0	max_lilmt The highest threshold of LIMITER. 3'b000~3'b100: 0db~-12db, 3db/step 3'b101~3'b111: -18db~-30db, 6db/step
3	RO	0x0	reserved
2:0	RW	0x0	min_lilmt The lowest threshold of LIMITER. 3'b000~3'b100: 0db~-12db, 3db/step 3'b101~3'b111: -18db~-30db, 6db/step

ACDCDIG_ADCLILMT2

Address: Operational Base + offset (0x0110)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:4	RW	0x0	atk_rate LIMITER attack rate=(power(2, atk_rate)*(8*clk)), clk is such as 4.096MHz, 5.6448MHz, 6.144MHz.
3:0	RW	0x0	rls_rate LIMITER release rate=(power(2, rls_rate)*(8*clk)), clk is such as 4.096MHz, 5.6448MHz, 6.144MHz.

ACDCDIG_ADCDMICNG1

Address: Operational Base + offset (0x0114)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7	RW	0x0	ngchl_li Noise gate channel. 1'b0: Individual channel(or) 1'b1: Both channel(and)
6	RW	0x0	ngen_li Noise gate enable. 1'b0: Noise gate disable 1'b1: Noise gate enable
5	RW	0x0	ngboost_li Noise gate boost. 1'b0: Normal noise gate 1'b1: Boost noise gate
4:2	RW	0x0	nggate_li Noise gate threshold. If ngboost is 1'b0: 3'b000~3'b111: -63db~-84db, 3db/step If ngboost is 1'b1: 3'b000~3'b111: -33db~-54db, 3db/step

Bit	Attr	Reset Value	Description
1:0	RW	0x0	ngdly_li Noise gate delay. The delay time before the noise gate attacks. 2'b00: 2048 samples 2'b01: 4096 samples 2'b10: 8192 samples 2'b11: 16384 samples

ACDCDIG_ADCDMICNG2

Address: Operational Base + offset (0x0118)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved
1	RO	0x0	ngvalid_li_l2 Noise gates valid status of Limiter for left channel 2. 1'b0: Not in NG status 1'b1: Now in NG status
0	RO	0x0	ngvalid_li_l0r1 Noise gates valid status of Limiter for left channel 0 and right channel 1. 1'b0: Not in NG status 1'b1: Now in NG status

ACDCDIG_DACVUCTL

Address: Operational Base + offset (0x0140)

Bit	Attr	Reset Value	Description
31:3	RO	0x00000000	reserved
2	RW	0x0	dac_byps Digital DAC volume control bypass. 1'b0: Digital DAC volume control enable 1'b1: Digital DAC volume control bypass
1	RW	0x0	dacfade Digital DAC volume adjust mode. 1'b0: Update to new volume immediately. 1'b1: Update volume as daczdt field describes.
0	RW	0x1	daczdt Digital DAC volume cross zero detect enable. It works when adc_byps is 1'b0 and adcfade is 1'b1. 1'b0: Volume adjusts every sample. 1'b1: Volume adjusts only when audio waveform crosses zero or volume-control time-limit condition meets.

ACDCDIG_DACVUCTIME

Address: Operational Base + offset (0x0144)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x00	dacvuct Volume control time limit, valid only in fade cross zero mode. Time limit = dacvuct*(1/sample rate) Unit: sample rate

ACDCDIG_DACDIGEN

Address: Operational Base + offset (0x0148)

Bit	Attr	Reset Value	Description
31:5	RO	0x0000000	reserved

Bit	Attr	Reset Value	Description
4	RW	0x0	adcglben Global enable of all Digital DAC channels. Only when adcglben and the enable signal corresponding to each Digital DAC channel is valid before starting work. 1'b0: Disable 1'b1: Enable
3:1	RO	0x0	reserved
0	RW	0x0	dacen_l0r1 Digital DAC left channel 0 and right channel 1 enable. 1'b0: Disable 1'b1: Enable

ACDCDIG_DACCLKCTRL

Address: Operational Base + offset (0x014c)

Bit	Attr	Reset Value	Description
31:6	RO	0x0000000	reserved
5	RW	0x0	dac_cke Digital DAC operation clock enable. 1'b0: Disable 1'b1: Enable
4	RW	0x0	i2srx_cke Clock enable of internal I2S RX channel. 1'b0: Disable 1'b1: Enable
3	RW	0x0	cke_bclkrx Clock enable of sclk_out_rx. 1'b0: Disable 1'b1: Enable
2	RW	0x0	dac_sync_ena Enable of the synchronization signal generated by Digital DAC. Note that only when both glbcke and dac_sync_ena are equal to 1'b1, the synchronization signal of Digital DAC can be generated. 1'b0: Disable 1'b1: Enable
1	RW	0x0	dac_sync_status There is a counter to generate synchronization signal of Digital DAC. In order to ensure the integrity of synchronization signal, it is necessary to read back dac_sync_status to judge whether the counter stops working when dac_sync_ena is set from 1'b1 to 1'b0. If the signal is read back to 1'b0, it means that the counter stops working and the synchronization signal of Digital DAC is complete.
0	RW	0x0	dac_mod_attenu_en When enabled, the input of the Digital DAC modulator is attenuated by 6dB, and the output of the Digital DAC modulator is increased by 6dB. 1'b0: Disable 1'b1: Enable

ACDCDIG_DACINT_DIV

Address: Operational Base + offset (0x0154)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved

Bit	Attr	Reset Value	Description
7:0	RW	0x07	int_div_con Interger clock divider to provide 6.144/5.644/4.096MHz sample clock for internal filters. Make sure that int_div_con is an odd number between 7(8 times division) and 15(16 times division).

ACDCDIG_DACSCLKRXINT_DIV

Address: Operational Base + offset (0x0160)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x1f	sckrxdiv Interger clock divider to generate sclk_out_rx when I2S RX works in master mode. It is ignored when in slave mode.

ACDCDIG_DACPWM_DIV

Address: Operational Base + offset (0x0164)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x03	audio_pwm_div PWM mode division of Digital DAC's operation clock.

ACDCDIG_DACPWM_CTRL

Address: Operational Base + offset (0x0168)

Bit	Attr	Reset Value	Description
31:7	RO	0x0000000	reserved
6	RW	0x0	pwm_mode_cke Clock enable of 1-bit PWM modulator 1'b0: Disable 1'b1: Enable
5:4	RW	0x0	pwm_mode Audio PWM mode or Audio DAC mode selection 2'b00: Audio DAC mode 2'b01: Audio PWM mode 0. The input of 1-bit PWM modulator is from the last filter of Audio DAC. 2'b01: Audio PWM mode 1. The input of 1-bit PWM modulator is directly from output of I2S RX inside the ACDCDIG.
3	RW	0x0	pwm_en 1-bit PWM modulator enable 1'b0: Disable 1'b1: Enable
2:0	RW	0x0	dith_sel Dith mode selection of 1-bit PWM modulator

ACDCDIG_DACCFG1

Address: Operational Base + offset (0x0184)

Bit	Attr	Reset Value	Description
31:5	RO	0x0000000	reserved
4:2	RW	0x1	dacsrt Sample rates of Digital DAC. 3'b000: 12kHz/11.024kHz/8kHz 3'b001: 24kHz/22.05kHz/16kHz 3'b010: 32kHz/48kHz/44.1kHz 3'b011: 96kHz/88.2kHz/64kHz 3'b100: 192kHz/176.4kHz/128kHz 3'b101~3'b111: Reserved

ACDCDIG_DACMUTE

Address: Operational Base + offset (0x0188)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved
1	RW	0x0	dacunmt 1'b0: DAC normal mode 1'b1: DAC unmute mode. In this mode, DAC volume control block will adjust volume to match the value in DACVOLL* and DACVOLR*.
0	RW	0x0	dacmt 1'b0: DAC normal mode 1'b1: DAC mute mode

ACDCDIG_DACMUTEST

Address: Operational Base + offset (0x018c)

Bit	Attr	Reset Value	Description
31:5	RO	0x0000000	reserved
4	RO	0x0	unmutest_l0r1 Unmute status for Digital DAC left channel 0 and right channel 1. When unmute is done, it indicates that internal volume is equal to the value programmed in DACVOLL* and DACVOLR*. 1'b0: Unmute not done 1'b1: Unmute done
3:1	RO	0x0	reserved
0	RO	0x0	mutest_l0r1 Mute status for Digital DAC left channel 0 and right channel 1. 1'b0: Not mute 1'b1: Mute

ACDCDIG_DACVOLL0

Address: Operational Base + offset (0x0190)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x00	daclv0 Volume of Digital DAC left channel 0. 0db~-95.625db, 0.375db/step. 8'h0: 0db 8'h1: -0.375db 8'h2: -0.75db 8'h3: -1.125db ..... 8'hff: -95.625db

ACDCDIG_DACVOLR0

Address: Operational Base + offset (0x01a0)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x00	dacrv0 Volume of Digital DAC right channel 1. 0db~-95.625db, 0.375db/step. 8'h0: 0db 8'h1: -0.375db 8'h2: -0.75db 8'h3: -1.125db ..... 8'hff: -95.625db

ACDCDIG_DACVOGP

Address: Operational Base + offset (0x01b0)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved
1	RW	0x0	volgpr1 Gain polarity for the volume of Digital DAC right channel 1. 1'b0: Negative gain 1'b1: Positive gain
0	RW	0x0	volgpl0 Gain polarity for the volume of Digital DAC left channel 0. 1'b0: Negative gain 1'b1: Positive gain

ACDCDIG_DACRVOLL0

Address: Operational Base + offset (0x01b4)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RO	0xff	rvoll0 Internal real-time volume of Digital DAC left channel 0.

ACDCDIG_DACRVOLR0

Address: Operational Base + offset (0x01c4)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RO	0xff	rvolr0 Internal real-time volume of Digital DAC right channel 1.

ACDCDIG_DACLMT0

Address: Operational Base + offset (0x01d4)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved
1	RW	0x0	limen LIMITER enable. 1'b0: Disable 1'b1: Enable
0	RW	0x0	limdct Limiter detect mode. 1'b0: (Left channel + right channel)/2 1'b0: Left channel or right channel independently

ACDCDIG_DACLMT1

Address: Operational Base + offset (0x01d8)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:4	RW	0x0	atk_rate LIMITER attack rate=(power(2, atk_rate)*(8*clk)), clk is such as 4.096MHz, 5.6448MHz, 6.144MHz.
3:0	RW	0x0	rls_rate LIMITER release rate=(power(2, rls_rate)*(8*clk)), clk is such as 4.096MHz, 5.6448MHz, 6.144MHz.

ACDCDIG_DACLMT2

Address: Operational Base + offset (0x01dc)

Bit	Attr	Reset Value	Description
31:7	RO	0x0000000	reserved

Bit	Attr	Reset Value	Description
6:4	RW	0x0	max_lilmt The highest threshold of LIMITER. 3'b000~3'b100: 0db~-12db, 3db/step 3'b101~3'b111: -18db~-30db, 6db/step
3	RO	0x0	reserved
2:0	RW	0x0	min_lilmt The lowest threshold of LIMITER. 3'b000~3'b100: 0db~-12db, 3db/step 3'b101~3'b111: -18db~-30db, 6db/step

ACDCDIG_DACMIXCTRLL

Address: Operational Base + offset (0x01e0)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved
1:0	RW	0x0	mixmode_l0 Digital DAC left channel 0 mixing mode. 2'b00: Left channel 2'b01: Right channel 2'b10~2'b11: (Left channel + right channel)/2

ACDCDIG_DACMIXCTRLR

Address: Operational Base + offset (0x01e4)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved
1:0	RW	0x0	mixmode_r0 Digital DAC right channel 1 mixing mode. 2'b00: Left channel 2'b01: Right channel 2'b10~2'b11: (Left channel + right channel)/2

ACDCDIG_DACHPF

Address: Operational Base + offset (0x01e8)

Bit	Attr	Reset Value	Description
31:6	RO	0x0000000	reserved
5:4	RW	0x0	hpfcf High-pass filter control. 2'b00: 80Hz 2'b01: 100Hz 2'b10: 120Hz 2'b11: 140Hz
3:1	RO	0x0	reserved
0	RW	0x0	hpfen_l0r1 High-pass filter enable for left channel 0 and right channel 1. 1'b0: High-pass filter is disabled. 1'b1: High-pass filter is enabled.

ACDCDIG_I2C_FLT_CON0

Address: Operational Base + offset (0x0280)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
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.

ACDCDIG_I2C_FLT_CON1

Address: Operational Base + offset (0x0284)

Bit	Attr	Reset Value	Description
31:7	RO	0x0000000	reserved
6	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)
5	RW	0x0	nak_release_scl 1'b0: Hold scl as low when received nack. 1'b1: Release scl as high when received nack.
4	RW	0x0	flt_en SCL edge glitch filter enable. 1'b0: Disable 1'b1: Enable
3:0	RW	0xf	slv_hold_scl_th Slave hold scl threshold = slv_hold_scl_db * Tclk_i2c.

ACDCDIG_I2C_CON0

Address: Operational Base + offset (0x0288)

Bit	Attr	Reset Value	Description
31:7	RO	0x0000000	reserved
6	RW	0x0	act2nak Operation when NAK handshake is received. 1'b0: Ignored 1'b1: Stop transaction
5:3	RO	0x0	reserved
2:1	RW	0x0	i2c_mode I2C mode select. 2'b00: Transmit only 2'b01~2'b11: Reserved
0	RW	0x0	i2c_en I2C enable. 1'b0: Not enable 1'b1: Enable

ACDCDIG_I2C_CON1

Address: Operational Base + offset (0x028c)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:6	RW	0x0	stop_setup Stop setup configure. TSU: sto = (stop_setup + 1) * T(SCL_HIGH) + Tclk_i2c
5:4	RW	0x0	start_setup Start setup configure. TSU: sta = (start_setup + 1) * T(SCL_HIGH) + Tclk_i2c THD: sta = (start_setup + 2) * T(SCL_HIGH) - Tclk_i2c
3:0	RW	0x3	data_upd_st SDA update point configure. Used to configure sda change state when scl is low, used to adjust setup/hold time. 4'bn: Thold = (n + 1) * Tclk_i2c Note: 0 <= n <= 5

ACDCDIG_I2C_CLKDIVL0

Address: Operational Base + offset (0x0290)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x06	clkdivl0 Low 8 bits of scl low level clock divider. The value of 16 bits scl low level clock divider updates only when clkdivl1 is programmed. So ensure to program clkdivl0 firstly, followed by clkdivl1.

ACDCDIG_I2C_CLKDIVL1

Address: Operational Base + offset (0x0294)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x00	clkdivl1 High 8 bits of scl low level clock divider. The value of 16 bits scl low level clock divider updates only when clkdivl1 is programmed. So ensure to program clkdivl0 firstly, followed by clkdivl1. T(SCL_LOW) = Tclk_i2c * (CLKDIVL + 1) * 8, where the CLKDIVL is equal to clkdivl0+clkdivl1*256.

ACDCDIG_I2C_CLKDIVH0

Address: Operational Base + offset (0x0298)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x06	clkdivh0 Low 8 bits of scl high level clock divider. The value of 16 bits scl high level clock divider updates only when clkdivh1 is programmed. So ensure to program clkdivh0 firstly, followed by clkdivh1.

ACDCDIG_I2C_CLKDIVH1

Address: Operational Base + offset (0x029c)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x00	clkdivh1 High 8 bits of scl high level clock divider. The value of 16 bits scl high level clock divider updates only when clkdivh1 is programmed. So ensure to program clkdivh0 firstly, followed by clkdivh1. T(SCL_LOW) = Tclk_i2c * (CLKDIVH + 1) * 8, where the CLKDIVH is equal to clkdivh0+clkdivh1*256.

ACDCDIG_I2C_MAXCNT

Address: Operational Base + offset (0x02a0)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7	RO	0x1	idle 1'b0: I2C module is busy. 1'b1: I2C module is idle.
6	RO	0x0	reserved
5:0	RW	0x00	mtxcnt Master transmit number. Specify the total bytes to be transmitted (0~32).

ACDCDIG_I2C_SCLOE_DB0

Address: Operational Base + offset (0x02a4)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x20	scloedb0 Bit7~bit0 of slave hold scl debounce register. Cycles for debounce (unit: Tclk_i2c).

ACDCDIG_I2C_SCLOE_DB1

Address: Operational Base + offset (0x02a8)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x00	scloedb1 Bit15~bit8 of slave hold scl debounce register. Cycles for debounce (unit: Tclk_i2c).

ACDCDIG_I2C_SCLOE_DB2

Address: Operational Base + offset (0x02ac)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x00	scloedb2 Bit23~bit16 of slave hold scl debounce register. Cycles for debounce (unit: Tclk_i2c).

ACDCDIG_I2C_SCLOE_DB3

Address: Operational Base + offset (0x02b0)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x00	scloedb3 Bit31~bit24 of slave hold scl debounce register. Cycles for debounce (unit: Tclk_i2c).

ACDCDIG_I2C_TMOUTL

Address: Operational Base + offset (0x02b4)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x1f	tmoutl Low 8 bits of 16 bits I2C transmit request time out register. The value of 16 bits I2C transmit request time out register updates only when tmouth is programmed. So ensure to program tmoutl firstly, followed by tmouth.

ACDCDIG_I2C_TMOUTH

Address: Operational Base + offset (0x02b8)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x00	tmouth High 8 bits of 16 bits I2C transmit request time out register. The value of 16 bits I2C transmit request time out register updates only when tmouth is programmed. So ensure to program tmoutl firstly, followed by tmouth.

ACDCDIG_I2C_DEV_ADDR

Address: Operational Base + offset (0x02bc)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved

Bit	Attr	Reset Value	Description
7:0	RW	0x48	dev_addr I2C slave device address.

ACDCDIG_I2C_REG_ADDR

Address: Operational Base + offset (0x02c0)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x28	regaddr I2C slave register address to be accessed.

ACDCDIG_I2C_STATUS

Address: Operational Base + offset (0x02c4)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7	RO	0x0	slavehdsclst Slave hold scl status bit. 1'b0: Slave not hold scl. 1'b1: Slave hold scl, write 1'b1 to clear.
6	RO	0x0	nakrcvst NAK handshake received status bit. 1'b0: No NAK handshake received. 1'b1: NAK handshake received, write 1'b1 to clear.
5	RO	0x0	stopst Stop operation finished status bit. 1'b0: No finished. 1'b1: Stop operation finished, write 1'b1 to clear.
4	RO	0x0	startst Start operation finished status bit. 1'b0: Not finished. 1'b1: Start operation finished, write 1'b1 to clear.
3	RO	0x0	reserved
2	RO	0x0	mbtfst I2C_MTXCNT data transfer finished status bit. 1'b0: Not finished. 1'b1: I2C_MTXCNT data transfer finished, write 1'b1 to clear.
1	RO	0x0	reserved
0	RO	0x0	btfst Byte tx finished status bit. 1'b0: Byte tx not finish. 1'b1: Byte tx finished, write 1'b1 to clear.

ACDCDIG_I2S_TXCR0

Address: Operational Base + offset (0x0300)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:6	RW	0x0	pbm 2'b00: PCM no delay mode 2'b01: PCM delay 1 mode 2'b10: PCM delay 2 mode 2'b11: PCM delay 3 mode
5	RW	0x0	tfs 1'b0: I2S format 1'b1: PCM format

Bit	Attr	Reset Value	Description
4:0	RW	0x0f	vdw 5'b00000~5'b01110: Reserved 5'b01111: 16bit 5'b10000: 17bit 5'b10001: 18bit 5'b10010: 19bit ..... 5'b11100: 29bit 5'b11101: 30bit 5'b11110: 31bit 5'b11111: 32bit

ACDCDIG_I2S_TXCR1

Address: Operational Base + offset (0x0304)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:6	RW	0x0	tcsr 2'b00: Two channel 2'b01: Four channel Others: Reserved
5	RO	0x0	reserved
4	RW	0x0	cex Exchange left channel and right channel in the every transmit line. 1'b0: Not exchange 1'b1: Exchange
3	RO	0x0	reserved
2	RW	0x0	fbm 1'b0: MSB 1'b1: LSB
1:0	RW	0x0	ibm 2'b00: I2S normal 2'b01: I2S Left justified 2'b10: I2S Right justified 2'b11: Reserved

ACDCDIG_I2S_TXCR2

Address: Operational Base + offset (0x0308)

Bit	Attr	Reset Value	Description
31:6	RO	0x0000000	reserved
5:0	RW	0x00	rcnt Only valid in I2S Right justified format and slave tx mode is selected. Start to transmit data rcnt sclk cycles after left channel valid.

ACDCDIG_I2S_RXCR0

Address: Operational Base + offset (0x030c)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:6	RW	0x0	pbm 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
5	RW	0x0	tfs 1'b0: I2S format 1'b1: PCM format
4:0	RW	0x0f	vdw 5'b00000~5'b01110: Reserved 5'b01111: 16bit 5'b10000: 17bit 5'b10001: 18bit 5'b10010: 19bit ..... 5'b11100: 29bit 5'b11101: 30bit 5'b11110: 31bit 5'b11111: 32bit

ACDCDIG_I2S_RXCR1

Address: Operational Base + offset (0x0310)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:6	RW	0x0	rcsr 2'b00: Two channel Others: Reserved
5	RO	0x0	reserved
4	RW	0x0	cex Exchange left channel and right channel in the every receive line. 1'b0: Not exchange 1'b1: Exchange
3	RO	0x0	reserved
2	RW	0x0	fbm 1'b0: MSB 1'b1: LSB
1:0	RW	0x0	ibm 2'b00: I2S normal 2'b01: I2S Left justified 2'b10: I2S Right justified 2'b11: Reserved

ACDCDIG_I2S_CKR0

Address: Operational Base + offset (0x0314)

Bit	Attr	Reset Value	Description
31:4	RO	0x0000000	reserved
3:2	RW	0x0	rsd I2S rx sclk divider for rx_lrck generator. 2'b00: 64 2'b01: 128 2'b10~2'b11: 256
1:0	RW	0x0	tsd I2S tx sclk divider for tx_lrck generator. 2'b00: 64 2'b01: 128 2'b10~2'b11: 256

ACDCDIG_I2S_CKR1

Address: Operational Base + offset (0x0318)

Bit	Attr	Reset Value	Description
31:4	RO	0x0000000	reserved
3	RW	0x0	mss 1'b0: Master mode(sclk output) 1'b1: Slave mode(sclk input)
2	RW	0x0	ckp 1'b0: Sample data at posedge sclk and drive data at negedge sclk. 1'b1: Sample data at negedge sclk and drive data at posedge sclk.
1	RW	0x0	rlp 1'b0: Normal polarity (I2S normal: low for left channel, high for right channel I2S left/right just: high for left channel, low for right channel PCM start signal: high valid) 1'b1: Opposite polarity (I2S normal: high for left channel, low for right channel I2S left/right just: low for left channel, high for right channel PCM start signal: low valid)
0	RW	0x0	tlp 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)

ACDCDIG_I2S_XFER

Address: Operational Base + offset (0x031c)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved
1	RW	0x0	rxs 1'b0: Stop RX transfer 1'b1: Start RX transfer
0	RW	0x0	txs 1'b0: Stop TX transfer 1'b1: Start TX transfer

ACDCDIG_I2S_CLR

Address: Operational Base + offset (0x0320)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved
1	RW	0x0	rxc This is a self-cleared bit. Write 1 to clear all receive logic.
0	RW	0x0	txc This is a self-cleared bit. Write 1 to clear all transmit logic.

ACDCDIG_VERSION

Address: Operational Base + offset (0x0380)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x02	ver Version of ACDCDIG.

18.5 Interface Description

Digital Audio Codec supports audio DAC mode and audio PWM mode. The following table shows the Digital Audio Codec interface description for audio DAC mode.

Table 18-2 Digital Audio Codec Interface Description for Audio DAC Mode

Module Pin	Direction	Pad Name	IOMUX Setting
D2A_ADC_CLK	O	I2S0_SCLK_RX_M0/PDM_CLk1_M0/ACODEC_ADC_CLK/GPIO3_D1_d	GRF_GPIO3D_IOMUX_L[6:4]=3'b011
D2A_ADC_SYNC	O	I2S0_LRCK_RX_M0/PDM_CLk0_M0/ACODEC_ADC_SYNC/GPIO3_D4_d	GRF_GPIO3D_IOMUX_H[2:0]=3'b011
A2D_ADC_DATA	I	I2S0_SDO1_SDI3_M0/PDM_SDI3_M0/ACODEC_ADC_DATA/GPIO3_D7_d	GRF_GPIO3D_IOMUX_H[14:12]=3'b011
D2A_DAC_CLK	O	I2S0_SCLK_TX_M0/ACODEC_DAC_CLK/GPIO3_D0_d	GRF_GPIO3D_IOMUX_L[2:0]=3'b011
D2A_DAC_SYNC	O	I2S0_LRCK_TX_M0/ACODEC_DAC_SYNC/AUDPWM_L_M1/AUDDSM_LN/GPIO3_D3_d	GRF_GPIO3D_IOMUX_L[14:12]=3'b011
D2A_DAC_DATA_L	O	I2S0_SDI0_M0/PDM_SDI0_M0/ACODEC_DAC_DATAL/GPIO3_D6_d	GRF_GPIO3D_IOMUX_H[10:8]=3'b011
D2A_DAC_DATA_R	O	I2S0_SDO0_M0/ACODEC_DAC_DATAR/AUDPWM_R_M1/AUDDSM_LP/GPIO3_D5_d	GRF_GPIO3D_IOMUX_H[6:4]=3'b011

When operating in audio PWM mode, Digital Audio Codec outputs two pairs of differential signals encoded in PWM format. The following table shows the Digital Audio Codec interface description for audio PWM mode.

Table 18-3 Digital Audio Codec Interface Description for Audio PWM Mode

Module Pin	Direction	Pad Name	IOMUX Setting
AUDIO_PWM_L_P	O	I2S0_SDO0_M0/ACODEC_DAC_DATAR/AUDPWM_R_M1/AUDDSM_LP/GPIO3_D5_d	GRF_GPIO3D_IOMUX_H[6:4]=3'b101
AUDIO_PWM_L_N	O	I2S0_LRCK_TX_M0/ACODEC_DAC_SYNC/AUDPWM_L_M1/AUDDSM_LN/GPIO3_D3_d	GRF_GPIO3D_IOMUX_L[14:12]=3'b101
AUDIO_PWM_R_P	O	I2S0_SDO3_SDI1_M0/PDM_SDI1_M0/AUDPWM_R_M0/I2C4_SDA_M1/AUDDSM_RP/GPIO4_A1_d	GRF_GPIO4A_IOMUX_L[6:4]=3'b101
AUDIO_PWM_R_N	O	I2S0_SDO2_SDI2_M0/PDM_SDI2_M0/AUDPWM_L_M0/I2C4_SCL_M1/AUDDSM_RN/GPIO4_A0_d	GRF_GPIO4A_IOMUX_L[2:0]=3'b101

18.6 Application Notes

18.6.1 Frequency Configuration

ACDC_ADC_CLK and ACDC_DAC_CLK are input clocks of Digital Audio Codec. Operation clock of digital ADC and I2S master TX mode are generated from ACDC_ADC_CLK, while operation clock of digital DAC and I2S master RX mode are generated from ACDC_DAC_CLK. ACDC_ADC_CLK and ACDC_DAC_CLK are asynchronous to each other. ACDC_ADC_CLK must be homologous to the MCLK_TX of I2S which is connected to Digital Audio Codec when the I2S TX module inside Digital Audio Codec acts as a slave. It is similar that ACDC_DAC_CLK must be homologous to the MCLK_RX of I2S which is connected to Digital Audio Codec when the I2S RX module inside Digital Audio Codec acts as a slave.
Clocks named D2A_ADC_CLK and D2A_DAC_CLK acting as interface clock generated from

ACDC_ADC_CLK and ACDC_DAC_CLK respectively are sent out to Analog Codec. In order to transfer audio data between digital Codec and Analog Codec, in addition to the audio data lines of digital ADC and DAC, two periodic synchronous signal named D2A_ADC_SYNC and D2A_DAC_SYNC are needed to indicate the start of every sample data.
The Digital ADC supports three application scenarios that are normal mode, low power mode 1 and low power mode 2. Different application mode requires different frequency of ACDC_ADC_CLK. If ACDCDIG_ADCCLKCTRL.cic_ds_ratio is programmed to 16, 8 and 4, the Digital ADC operates in normal mode, low power mode 1 and low power mode 2 respectively.
The relationship of ACDC_ADC_CLK, D2A_ADC_CLK, D2A_ADC_SYNC and sample rates is as follows where the D2A_ADC_CLK is 8 times of D2A_ADC_SYNC.

Table 18-4 Relationship of ACDC_ADC_CLK, D2A_ADC_CLK, D2A_ADC_SYNC and Sample Rates in Normal Mode

ACDC_ADC_CLK	D2A_ADC_CLK	D2A_ADC_SYNC	Sample rates supported
49.152MHz	49.152MHz	6.144MHz	12/24/48/96/192kHz
45.154MHz	45.154MHz	5.644MHz	11.024/22.05/44.1/88.2/176.4 kHz
32.768MHz	32.768MHz	4.096MHz	8/16/32/64/128kHz

Table 18-5 Relationship of ACDC_ADC_CLK, D2A_ADC_CLK, D2A_ADC_SYNC and Sample Rates in Low Power Mode 1

ACDC_ADC_CLK	D2A_ADC_CLK	D2A_ADC_SYNC	Sample rates supported
24.576MHz	24.576MHz	3.072MHz	12/24/48/96kHz
22.577MHz	22.577MHz	2.822MHz	11.024/22.05/44.1/88.2kHz
16.384MHz	16.384MHz	2.048MHz	8/16/32/64kHz

Table 18-6 Relationship of ACDC_ADC_CLK, D2A_ADC_CLK, D2A_ADC_SYNC and Sample Rates in Low Power Mode 2

ACDC_CLK	D2A_ADC_CLK	D2A_ADC_SYNC	Sample rates supported
12.288MHz	1.536MHz	1.536MHz	12/24/48kHz
11.288MHz	1.411MHz	1.411MHz	11.024/22.05/44.1kHz
8.192MHz	1.024MHz	1.024MHz	8/16kHz

For Digital DAC, the relationship of ACDC_DAC_CLK, D2A_DAC_CLK, D2A_DAC_SYNC and sample rates is as follows where the D2A_DAC_CLK is 8 times of D2A_DAC_SYNC.

Table 18-7 Relationship of ACDC_DAC_CLK, D2A_DAC_CLK, D2A_DAC_SYNC and Sample Rates

ACDC_DAC_CLK	D2A_DAC_CLK	D2A_DAC_SYNC	Sample rates supported
49.152MHz	49.152MHz	6.144MHz	12/24/48/96/192kHz
45.154MHz	45.154MHz	5.644MHz	11.024/22.05/44.1/88.2/176.4 kHz
32.768MHz	32.768MHz	4.096MHz	8/16/32/64/128kHz

18.6.2 Operation Mode

The Digital Audio Codec supports two operation modes, asynchronous mode and synchronous mode by programming ACDCDIG_SYSCTRL0.sync_mode.
The asynchronous mode refers to that ACDC_ADC_CLK and ACDC_DAC_CLK are asynchronous to each other. The interface of Digital ADC is consists of D2A_ADC_CLK, D2A_ADC_SYNC and A2D_ADC_DATA, independently to the interface of Digital DAC which is consists of D2A_DAC_CLK, D2A_DAC_SYNC, D2A_DAC_DATA_L and D2A_DAC_DATA_R. There are no relationship between interface of Digital ADC and Digital DAC.
The synchronous mode refers to that ACDC_ADC_CLK and ACDC_DAC_CLK are fed into a mux to generate a common operation clock(named as ACDC_COM_CLK) for both Digital ADC and DAC. The operation clock of Digital ADC and DAC are from the same source. In this situation, the Digital Audio Codec use
D2A_ADC_CLK, D2A_ADC_SYNC , A2D_ADC_DATA, D2A_DAC_DATA_L and D2A_DAC_DATA_R to communicate with Analogy Audio Codec. D2A_ADC_CLK and D2A_ADC_SYNC are shared by Digital ADC and DAC.
When operates in synchronous mode, some restrictions should be taken into account. There are 3 applications(application mode 0, application mode 1 and application mode 2) that should be considered separately.
For application mode 0 and application mode 1, there are no restrictions on ACDC_COM_CLK. Just configure the frequency of ACDC_COM_CLK according to sample rate and make sure that the frequency of D2A_ADC_CLK is at least 8 times of D2A_ADC_SYNC.

In this situation, the divider ACDCDIG_ADCINT_DIV and ACDCDIG_DACINT_DIV must be equal to 0x07.
For application mode 2, there are some differences. For example, sample rate of digital ADC is 12kHz while sample rate of DAC is 128kHz. Configure ACDC_COM_CLK to the higher frequency such as 49.152MHz. The divider ACDCDIG_ADCINT_DIV must be set to 0x07 while ACDCDIG_DACINT_DIV must be set to 0x0b. On the contrary, if sample rate of digital ADC is 128kHz while sample rate of DAC is 12kHz. Configure ACDC_COM_CLK to the higher frequency such as 49.152MHz. The divider ACDCDIG_ADCINT_DIV must be set to 8'h0b while ACDCDIG_DACINT_DIV must be set to 0x07.

18.6.3 Software Application Notes

Steps to configure Digital ADC and DAC to start transfer at the same time are as follows.
1. Program CRU in the SOC system to achieve the frequency of ACDC_ADC_CLK and ACDC_DAC_CLK.
2. Program ACDCDIG_SYSCTRL0 to set sync_mode and clk_com_sel.
3. Program ACDCDIG_ADCINT_DIV to 0x07 or 0x0b according to application modes.
4. Program ACDCDIG_ADCCLKCTRL to 0x3e. It is preferred to set cic_ds_ratio=2'b00 to make Digital Audio Codec work in normal mode.
5. Program ACDCDIG_ADCSCLKTXINT_DIV.
6. Program ACDCDIG_I2S_CKR0 and ACDCDIG_I2S_CKR1 to set I2S TX related registers.
7. Program ACDCDIG_DACINT_DIV to 0x07 or 0x0b according to application modes.
8. Program ACDCDIG_DACCLKCTRL to 0x3c.
9. Program ACDCDIG_DACSCLKRXINT_DIV.
10. Program ACDCDIG_I2S_CKR0 to set I2S RX related registers.
11. Program ACDCDIG_SYSCTRL0. clk_com_sel to select ACDC_COM_CLK source if works in synchronous mode. Program ACDCDIG_SYSCTRL0.sync_sel to select D2A_ADC_SYNC source. Don't enable glbcke at this time.
12. Program I2C related registers to set properly device address, timeout value and so on. Then program ACDCDIG_I2C_CON0 to enable I2C at last.
13. Program ACDCDIG_I2S_CLR to clear I2S TX and RX logic.
14. Program ACDCDIG_I2S_TXCR0, ACDCDIG_I2S_TXCR1, ACDCDIG_I2S_RXCR0 and ACDCDIG_I2S_RXCR1.
15. Program I2S controller that is connected to Digital Audio Codec. Don't start I2S TX and RX at this time.
16. Program digital ADC related registers such as ACDCDIG_ADCHPFEN, ACDCDIG_ADCVUCTL, ACDCDIG_ADCCFG1 to achieve basic configuration.
17. Program DAC related registers such as ACDCDIG_DACHPF, ACDCDIG_DACVUCTL and ACDCDIG_DACCFG1 to achieve basic configuration.
18. Program ACDCDIG_I2S_XFER to start I2S TX and RX.
19. Program I2S controller outside Digital Audio Codec to start I2S TX and RX.
20. Program ACDCDIG_SYSCTRL0.glbcke to 1'b1 to enable output of D2A_ADC_SYNC if works in synchronous mode. Skip this step if ACDCDIG_SYSCTRL0.sync_mode is 1'b0.
21. Program ACDCDIG_ADCDIGEN and ACDCDIG_DACDIGEN to enable digital ADC channels and DAC channels. From now on, the Digital Audio Codec begins to work.
Steps to configure Digital Audio Codec to end transfer are as follows.
1. Program ACDCDIG_ADCDIGEN and ACDCDIG_DACDIGEN to disable digital ADC channels and DAC channels.
2. Program ACDCDIG_ADCCLKCTRL.adc_sync_ena and 1'b0 and ACDCDIG_DACCLKCTRL.dac_sync_ena to 1'b0 to disable the D2A_ADC_SYNC and D2A_ADC_SYNC respectively. Wait ACDCDIG_ADCCLKCTRL.adc_sync_status and ACDCDIG_DACCLKCTRL.dac_sync_status until both of them read back to be 1'b0.
3. Program ACDCDIG_SYSCTRL0.glbcke to 1'b0 if works in synchronous mode.

Chapter 19 Audio PWM

19.1 Overview

The Audio PWM provides an easy and cheap solution for audio playback in low quality. It acts as a digital-to-analog converter (DAC), which converts the digital audio PCM data to the analog PWM signals.
The Audio PWM supports the following features:

- Support one 32bit AHB slave interface for the configuration and data access.
- Support one 32x32bits TX FIFO
- The FIFO can be written by Cortex-A7, MCU and DMAC
- Support 16bits~32bits source data width
- Support 8bits~11bits output sampling data width
- Support two maskable output channels, left and right, and they can be swapped
- Support the interpolation rate up to 15
- Support linear interpolation, and the interpolation rate must be 1/3/7/15 in this mode
- Support one interrupt output for FIFO almost full, overrun and empty

19.2 Block Diagram

The following figure shows the block diagram of Audio PWM.

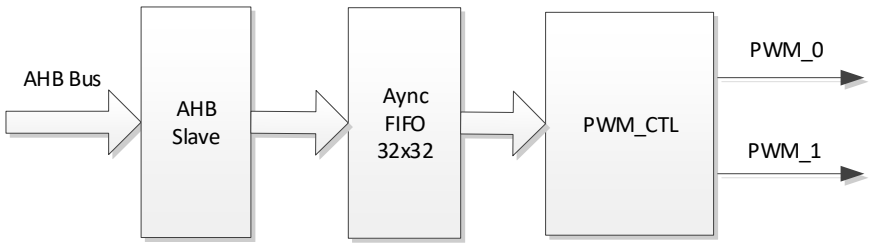


Fig. 19-1 Block Diagram of Audio PWM

19.3 Function Description

When the source data width is 16 bits, all the 32 bits of the FIFO data is valid in normal cases. The lower 16 bits are used for the channel 0 and the upper 16 bits are used for the channel 1. And when the source data width is greater than 16 bits, the word-wide data is used alternately for the channel 0 and the channel 1.
The Audio PWM samples two halfwords/words of the FIFO data for the two output channels, and the interpolation calculation also needs the next two halfwords/words. Therefore, the four sampling data values which stored in the four data buffers are needed at the beginning of the transfer. And it is required that the next two sampling data values are prefetched from the FIFO later. For this reason, when the transfer is finish by the long stop operation and the FIFO is empty, there are still two or four unprocessed words left in the buffers.
The Pulse Width Modulation (PWM) is a form of signal modulation where data is represented by the ratio of the high level time to the period, which is known as the duty cycle. In this chip, the PWM period is defined as (2^N-1) clock cycles, where N is the output sampling data width of the Audio PWM, which determines the duty cycle resolution and the required frequency of the system main clock. The duty cycle of 0% represents the minimum signed digit while the maximum signed digit is represented by the duty cycle of 100%.
The following table shows the duty cycle resolution of the Audio PWM.

Table 19-1 Duty Cycle Resolution of the Audio PWM

Output Width	Duty Cycle Resolution	Duty Cycle Range	Represented Signed Digit
8	256	0/255 ~ 255/255	-128 ~ 127

Output Width	Duty Cycle Resolution	Duty Cycle Range	Represented Signed Digit
9	512	0/511 ~ 511/511	-256 ~ 255
10	1024	0/1023 ~ 1023/1023	-512 ~ 511
11	2048	0/2047 ~ 2047/2047	-1024 ~ 1023

19.4 Register Description

This section describes the control/status registers of the design. Software should read and write these registers using 32-bits accesses.

19.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
AUDIOPWM_VERSION	0x0000	W	0x01000000	Version Number Register
AUDIOPWM_XFER	0x0004	W	0x00000000	Transfer Control Register
AUDIOPWM_SRC_CFG	0x0008	W	0x00000000	Source Data Configuration Register
AUDIOPWM_PWM_CFG	0x0010	W	0x00000000	PWM Configuration Register
AUDIOPWM_PWM_ST	0x0014	W	0x00000000	PWM Status Register
AUDIOPWM_PWM_BUF_01	0x0018	W	0x00000000	PWM Processed Data Buffer 0 & 1 Register
AUDIOPWM_PWM_BUF_23	0x001C	W	0x00000000	PWM Processed Data Buffer 2 & 3 Register
AUDIOPWM_FIFO_CFG	0x0020	W	0x00000000	FIFO Configuration Register
AUDIOPWM_FIFO_LVL	0x0024	W	0x00000000	FIFO Level Register
AUDIOPWM_FIFO_INT_EN	0x0028	W	0x00000000	FIFO Interrupt Enable Register
AUDIOPWM_FIFO_INT_ST	0x002C	W	0x00000000	FIFO Interrupt Status Register
AUDIOPWM_FIFO_ENTRY	0x0080	W	0x00000000	FIFO Data Entry Register

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

19.4.2 Detail Registers Description

AUDIOPWM_VERSION

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:0	RO	0x01000000	version Version number: 1.0.0.0.

AUDIOPWM_XFER

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:18	RO	0x0000	reserved
17:16	WO	0x0	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:2	RO	0x0000	reserved
1	R/WSC	0x0	lstop Long stop transfer control. 1'b0: Nothing 1'b1: Long stop operation If a 1 is written into this bit field, the transfer will stop and this bit will be auto cleared to 0 when read out all FIFO data. At this time, there are still 2 or 4 unsent words left in the buffers.

Bit	Attr	Reset Value	Description
0	R/W SC	0x0	start Transfer start or stop. 1'b0: Transfer stop 1'b1: Transfer start

AUDIOPWM_SRC_CFG

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:23	RO	0x000	reserved
22:16	WO	0x00	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:7	RO	0x000	reserved
6	RW	0x0	half_en Half mode enable. 1'b0: All the 32 bits of data is valid (The lower 16 bits for channel 0 and the upper 16 bits for channel 1) 1'b1: Only the lower 16 bits of data is valid and the upper 16 bits of data is invalid This bit field is significant only when 16 bits source data width is selected.
5	RW	0x0	align Source data alignment. 1'b0: Right aligned 1'b1: Left aligned This bit field is significant when 16 bits source data width is not selected.
4:0	RW	0x00	width Source data width. 5'd0~5'd14: Reserved 5'd15: 16 bits 5'd16: 17 bits 5'd17: 18 bits 5'd18: 19 bits 5'd19: 20 bits 5'd20: 21 bits 5'd21: 22 bits 5'd22: 23 bits 5'd23: 24 bits 5'd24: 25 bits 5'd25: 26 bits 5'd26: 27 bits 5'd27: 28 bits 5'd28: 29 bits 5'd29: 30 bits 5'd30: 31 bits 5'd31: 32 bits

AUDIOPWM_PWM_CFG

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:26	RO	0x00	reserved

Bit	Attr	Reset Value	Description
25:16	WO	0x000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:10	RO	0x00	reserved
9:8	RW	0x0	sample_width Output sampling data width. 2'b00: 8 bits 2'b01: 9 bits 2'b10: 10 bits 2'b11: 11 bits
7	RW	0x0	right_dis Right channel output disable. 1'b0: Enable 1'b1: Disable
6	RW	0x0	left_dis Left channel output disable. 1'b0: Enable 1'b1: Disable
5	RW	0x0	out_swap Output swap. 1'b0: Audio PWM channel 0 for left channel and Audio PWM channel 1 for right channel (Default output IO map) 1'b1: Audio PWM channel 1 for left channel and Audio PWM channel 0 for right channel (Swap output IO map)
4	RW	0x0	linear_interp_en Linear interpolation mode enable. 1'b0: Disable 1'b1: Enable Note that linear interpolation only support interpolation rate = 1/3/7/15.
3:0	RW	0x0	interp_rate Interpolation rate. 4'd0: Interpolate 0 point 4'd1: Interpolate 1 point 4'd2: Interpolate 2 points 4'd3: Interpolate 3 points 4'd4: Interpolate 4 points 4'd5: Interpolate 5 points 4'd6: Interpolate 6 points 4'd7: Interpolate 7 points 4'd8: Interpolate 8 points 4'd9: Interpolate 9 points 4'd10: Interpolate 10 points 4'd11: Interpolate 11 points 4'd12: Interpolate 12 points 4'd13: Interpolate 13 points 4'd14: Interpolate 14 points 4'd15: Interpolate 15 points

AUDIOPWM_PWM_ST

Address: Operational Base + offset (0x0014)

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

Bit	Attr	Reset Value	Description
1	RO	0x0	pwm_busy Audio PWM status. 1'b0: Audio PWM is idle 1'b1: Audio PWM is busy
0	RO	0x0	fifo_busy TX FIFO status. 1'b0: FIFO is idle 1'b1: FIFO is busy

AUDIOPWM_PWM_BUF_01

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:27	RO	0x00	reserved
26:16	RO	0x000	pwm_buf_1 Current data value of channel 1.
15:11	RO	0x00	reserved
10:0	RO	0x000	pwm_buf_0 Current data value of channel 0.

AUDIOPWM_PWM_BUF_23

Address: Operational Base + offset (0x001C)

Bit	Attr	Reset Value	Description
31:27	RO	0x00	reserved
26:16	RO	0x000	pwm_buf_3 Next data value of channel 1.
15:11	RO	0x00	reserved
10:0	RO	0x000	pwm_buf_2 Next data value of channel 0.

AUDIOPWM_FIFO_CFG

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	WO	0x000	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	almost_full_watermark TX FIFO almost full watermark. When the number of valid FIFO data entries is less than or equal to the value in this bit field, the almost full interrupt is triggered.
7	RW	0x0	dma_en DMA TX request enable. 1'b0: Disable 1'b1: Enable
6:5	RO	0x0	reserved
4:0	RW	0x00	dma_watermark DMA watermark. This bit field controls the level at which a DMA request is made by the transmit logic. The watermark level = watermark + 1; that is, dma_tx_req is generated when the number of valid FIFO data entries is greater than the value in this field.

AUDIOPWM_FIFO_LVL

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:6	RO	0x0000000	reserved
5:0	RO	0x00	fifo_space2full TX FIFO space2full level. Indicates the number of valid data entries in the TX FIFO.

AUDIOPWM_FIFO_INT_EN

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:3	RO	0x00000000	reserved
2	RW	0x0	empty_int_en TX FIFO empty interrupt enable. 1'b0: Disable 1'b1: Enable
1	RW	0x0	overrun_int_en TX FIFO overrun interrupt enable. 1'b0: Disable 1'b1: Enable
0	RW	0x0	full_int_en TX FIFO almost full interrupt enable. 1'b0: Disable 1'b1: Enable

AUDIOPWM_FIFO_INT_ST

Address: Operational Base + offset (0x002C)

Bit	Attr	Reset Value	Description
31:3	RO	0x00000000	reserved
2	RW	0x0	empty_int_st TX FIFO empty interrupt status. 1'b0: Nothing 1'b1: FIFO empty
1	RW	0x0	overrun_int_st TX FIFO overrun interrupt status. 1'b0: Nothing 1'b1: FIFO overrun
0	RO	0x0	full_int_st TX FIFO almost full interrupt status. 1'b0: Nothing 1'b1: FIFO almost full

AUDIOPWM_FIFO_ENTRY

Address: Operational Base + offset (0x0080)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	fifo_data_entry External DMAC or application writes data to this address.

19.5 Interface Description

Table 19-2 Audio PWM Interface Description

Module Pin	Direction	Pad Name	IOMUX Setting
AUDPWM_L	O	I2S0_SDO2_SDI2_M0/PDM_SDI2_M0/AUDPWM_L_M0/I2C4_SCL_M1/AUDDSM_RN/GPIO4_A0_d	GRF_GPIO4A_IOMUX_L[2:0]=3'b011

Module Pin	Direction	Pad Name	IOMUX Setting
		I2S0_LRCK_TX_M0/ACODEC_DAC_SYNC/AUDPWM_L_M1/AUDDSM_LN/GPIO3_D3_d	GRF_GPIO3D_IOMUX_L[14:12]=3'b100
AUDPWM_R	O	I2S0_SDO3_SDI1_M0/PDM_SDI1_M0/AUDPWM_R_M0/I2C4_SDA_M1/AUDDSM_RP/GPIO4_A1_d	GRF_GPIO4A_IOMUX_L[6:4]=3'b011
		I2S0_SDO0_M0/ACODEC_DAC_DATAR/AUDPWM_R_M1/AUDDSM_LP/GPIO3_D5_d	GRF_GPIO3D_IOMUX_H[6:4]=3'b100

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

19.6 Application Notes

- When the AUDIOPWM_FIFO_CFG.dma_en is set to 1 and the number of the valid FIFO data entries is greater than AUDIOPWM_FIFO_CFG.dma_watermark, the Audio PWM will generate a DMA request to the external DMAC. And then the required word-aligned data will be written to the FIFO if the DMAC has been correctly configured.
- Start the data transfer and output the processed PWM signals to the IO by configuring AUDIOPWM_XFER.start to 1. If this bit is set to 0, the processing will terminate immediately.
- Generally, you can also stop the processing when all the FIFO data has been read out by configuring AUDIOPWM_XFER.lstop to 1. In this way, the long stop bit will be auto cleared to 0 when the FIFO is empty. You should wait for the FIFO and the control logic of the Audio PWM to be idle by polling AUDIOPWM_PWM_ST.
- If the FIFO is empty but the processing is still running, the output level will always be low.
- You can enable the FIFO interrupt by configuring AUDIOPWM_FIFO_INT_EN and query the interrupt status from AUDIOPWM_FIFO_INT_ST. Writing 1 to the interrupt status bit of overrun or empty will clear it to 0, but the bit of almost full cannot be cleared by this means. Note that the FIFO overrun interrupt cannot occur in application.
- When the linear interpolation mode is enabled, you must interpolate 1/3/7/15 points by configuring AUDIOPWM_PWM_CFG.interp_rate.
- Although the left and right channels are masked, the whole processing logic of the Audio PWM is still running.
- The required clock frequency = the audio sample rate * the duty cycle resolution * (the interpolation rate + 1). For example, when the audio sample rate is 16KHz, the output width is 11bits, and interpolate 1 point, the frequency of sclk_audpwm = 16K * (2^11) * (1+1) = 65.536MHz. You should configuring the CRU to generate the accurate divided clock.

Chapter 20 Pulse Density Modulation Interface Controller

20.1 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 4 paths. Each path is composed of two digital microphone channels. It can be used with four stereo or eight mono microphones. Each path is enabled or disabled independently
- Support 16~24 bit sample resolution
- Support sample rate:
- 8kHz,16kHz,32kHz,64kHz,128kHz,11.025kHz,22.05kHz,44.1kHz,88.2kHz,176.4kHz,12kHz,24kHz,48kHz,96kHz,192kHz
- Support two 16-bit audio data store together in one 32-bit wide location
- Support 16 to 31 bit audio data left or right justified in 32-bit wide FIFO
- Support programmable left and right channel exchange

20.2 Block Diagram

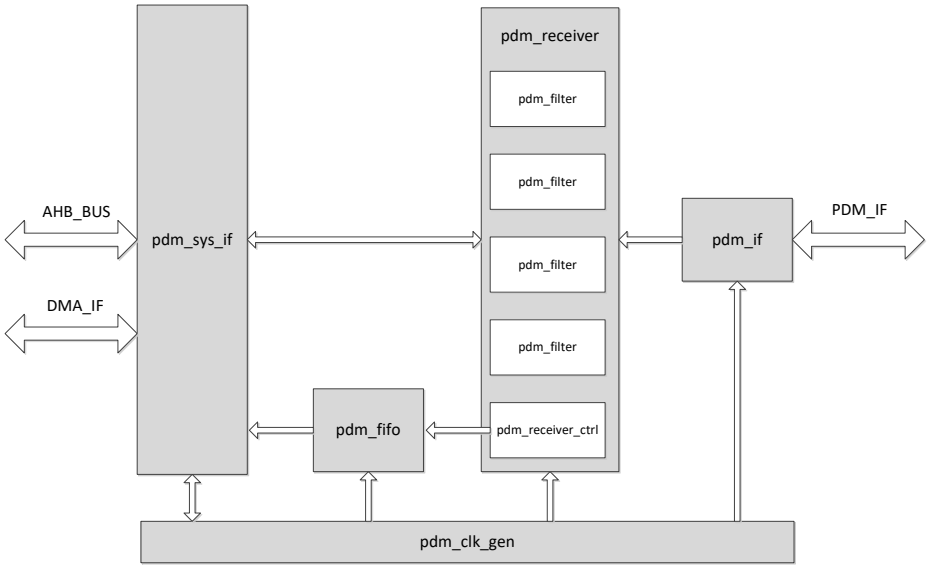


Fig.20-1 PDM Block Diagram

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.

20.3 Function Description

20.3.1 AHB Interface

There is an AHB slave interface in PDM. It is responsible for accessing registers.

20.3.2 PDM Interface

The PDM interface is a 5-wire interface. The PDM module can support up to four external stereo and eight digital microphones.
Following shows two cases of usage of the PDM, but all configurations are possible with stereo and mono digital microphones.

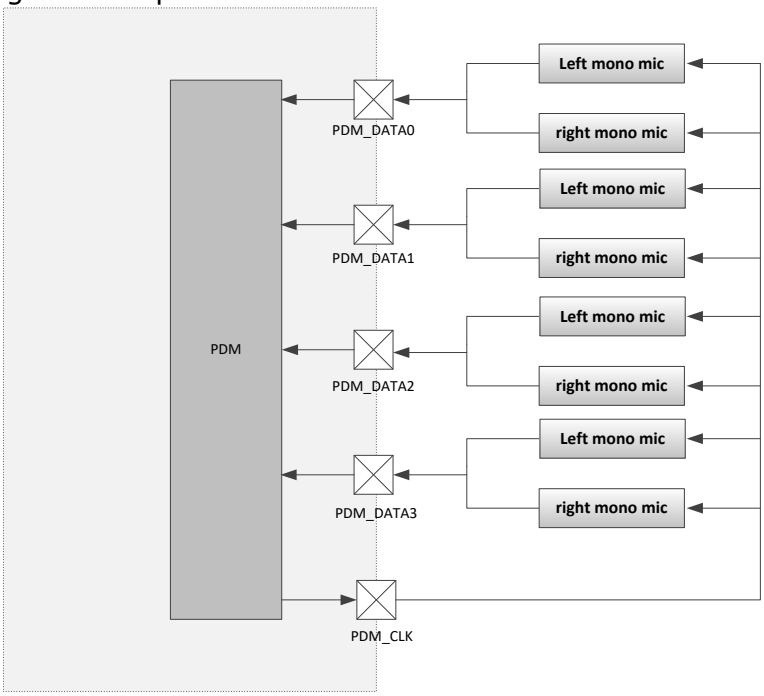


Fig.20-2 PDM with Eight Mono MIC

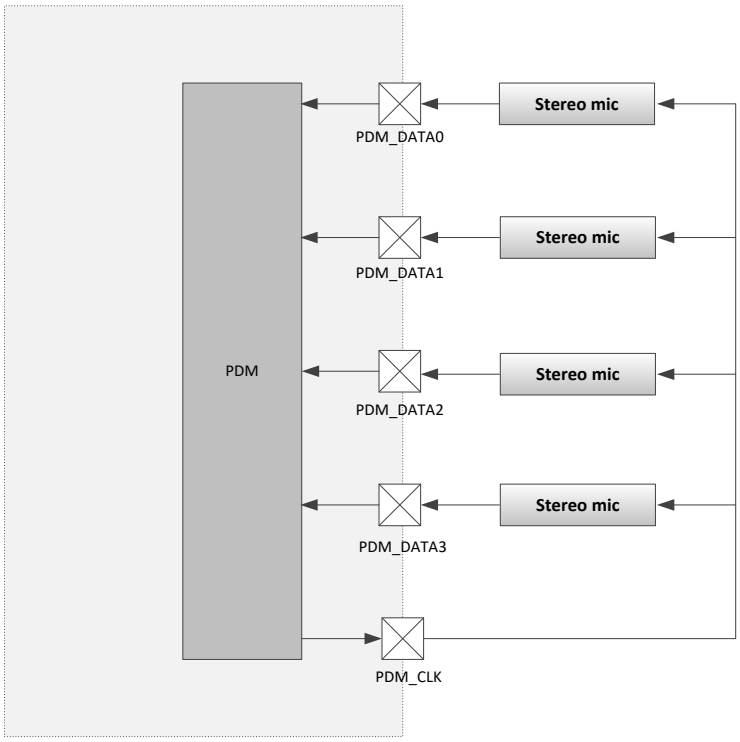


Fig.20-3 PDM with Four Stereo MIC

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 tri-state 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).

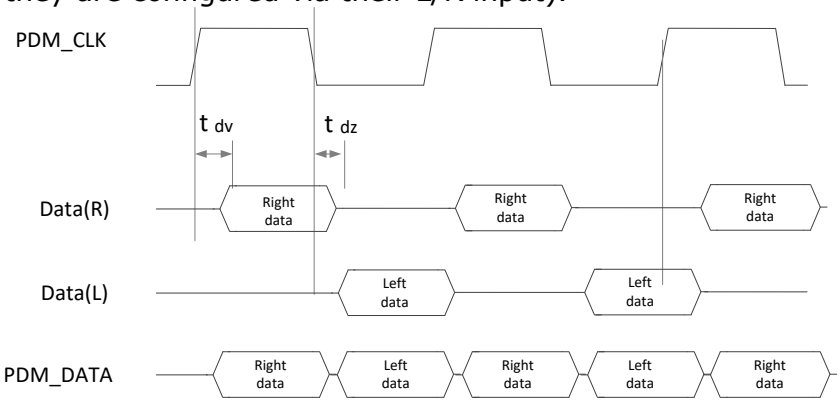


Fig.20-4 PDM interface diagram with external MIC

20.3.3 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.

20.3.4 Frequency Configuration

MCLK is the source clock signal. PDM_CLK 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).

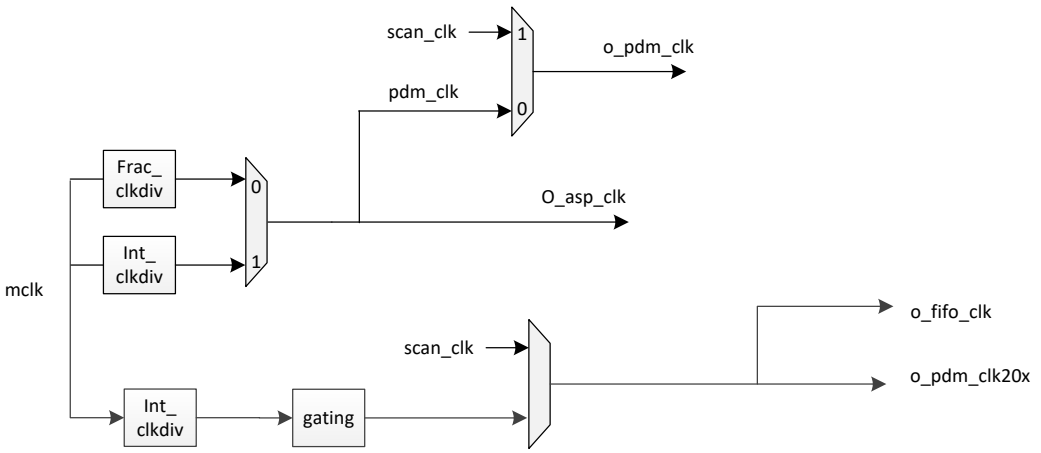


Fig.20-5 PDM Clock Structure

Table 20-1 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

20.4 Register Description

20.4.1 Internal Address Mapping

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

20.4.2 Registers Summary

Name	Offset	Size	Reset Value	Description
PDM_SYSCONFIG	0x0000	W	0x00000000	PDM System Configure Register
PDM_CTRL0	0x0004	W	0x78000377	PDM Control Register 0
PDM_CTRL1	0x0008	W	0x0bb8ea60	PDM Control Register 1
PDM_CLK_CTRL	0x000c	W	0x0000e401	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
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_DATA1R_REG	0x003c	W	0x00000000	PDM Path1 Right Channel Data Register
PDM_DATA1L_REG	0x0040	W	0x00000000	PDM Path1 Left Channel Data Register
PDM_DATA2R_REG	0x0044	W	0x00000000	PDM Path2 Right Channel Data Register
PDM_DATA2L_REG	0x0048	W	0x00000000	PDM Path2 Left Channel Data Register
PDM_DATA3R_REG	0x004c	W	0x00000000	PDM Path3 Right Channel Data Register

Name	Offset	Size	Reset Value	Description
PDM_DATA3L_REG	0x0050	W	0x00000000	PDM Path3 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:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

20.4.3 Detail Registers Description

PDM_SYSCONFIG

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:3	RO	0x00000000	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 1'b1: Clear RX logic Write 1'b0: No action Read 1'b1: Clear ongoing Read 1'b0: 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	RW	0x1	path3_en Path 3 enable 1'b1: Enable 1'b0: Disable
29	RW	0x1	path2_en Path 2 enable 1'b1: Enable 1'b0: Disable
28	RW	0x1	path1_en Path 1 enable 1'b1: Enable 1'b0: Disable
27	RW	0x1	path0_en Path 0 enable 1'b1: Enable 1'b0: Disable

Bit	Attr	Reset Value	Description
26	RW	0x0	hwt_en 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	0x03	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
4:0	RW	0x17	data_vld_width Can be written only when SYSCONFIG[2] is 0. Valid Data width 0~14: Reserved 15: 16bit 16: 17bit 17: 18bit 18: 19bit ..... n: (n+1)bit 23: 24bit

PDM_CTRL1

Address: Operational Base + offset (0x0008)

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

PDM_CLK_CTRL

Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15:14	RW	0x3	rx_path_select3 RX Path Select 2'b00: Path3 data from PDM data0 2'b01: Path3 data from PDM data1 2'b10: Path3 data from PDM data2 2'b11: Path3 data from PDM data3
13:12	RW	0x2	rx_path_select2 RX Path Select 2'b00: Path2 data from PDM data0 2'b01: Path2 data from PDM data1 2'b10: Path2 data from PDM data2 2'b11: Path2 data from PDM data3
11:10	RW	0x1	rx_path_select1 RX Path Select 2'b00: Path1 data from PDM data0 2'b01: Path1 data from PDM data1 2'b10: Path1 data from PDM data2 2'b11: Path1 data from PDM data3
9:8	RW	0x0	rx_path_select0 RX Path Select 2'b00: Path0 data from PDM data0 2'b01: Path0 data from PDM data1 2'b10: Path0 data from PDM data2 2'b11: Path0 data from PDM data3
7:6	RO	0x0	reserved
5	RW	0x0	pdm_clk_en PDM clk enable, working at PDM mode. Can be written only when SYSCONFIG[2] is 0. 1'b0: PDM clk disable 1'b1: PDM clk enable
4	RW	0x0	div_type_sel Divider type select signal Can be written only when SYSCONFIG[2] is 0. 1'b0: Fraction divider 1'b1: Integer divider
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	0x01	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	0x0000000	reserved

Bit	Attr	Reset Value	Description
3	RW	0x0	hpfle High-pass filter enable for left channel 1'b0: High pass filter for right channel is disabled. 1'b1: High pass filter for right channel is enabled.
2	RW	0x0	hpfre 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	0x00000	reserved
14:8	RW	0x00	rft Receive FIFO threshold When the number of receive FIFO entries is more than or equal to this threshold plus 1, the receive FIFO threshold interrupt is triggered.
7:0	RO	0x00	rfl 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	0x000000	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	0x00000000	reserved
1	RW	0x0	rxoie RX overflow interrupt enable 1'b0: Disable 1'b1: Enable

Bit	Attr	Reset Value	Description
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	0x00000000	reserved
1	RW	0x0	rxoic RX overflow interrupt clear (high active and auto cleared).

PDM_INT_ST

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved
1	RO	0x0	rxoi RX overflow interrupt 1'b0: Inactive 1'b1: Active
0	RO	0x0	rxfi RX full interrupt 1'b0: Inactive 1'b1: Active

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_DATA1R_REG

Address: Operational Base + offset (0x003c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	data1r Data of the path 1 right channel

PDM_DATA1L_REG

Address: Operational Base + offset (0x0040)

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

PDM_DATA2R_REG

Address: Operational Base + offset (0x0044)

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

PDM_DATA2L_REG

Address: Operational Base + offset (0x0048)

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

PDM_DATA3R_REG

Address: Operational Base + offset (0x004c)

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

PDM_DATA3L_REG

Address: Operational Base + offset (0x0050)

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

PDM_DATA_VALID

Address: Operational Base + offset (0x0054)

Bit	Attr	Reset Value	Description
31:4	RO	0x0000000	reserved
3	WO	0x0	path0_vld 1'b0: DATA0R_REG, DATA0L_REG value is invalid. 1'b1: DATA0R_REG, DATA0L_REG value is valid.
2	WO	0x0	path1_vld 1'b0: DATA1R_REG, DATA1L_REG value is invalid. 1'b1: DAT1R_REG, DATA1L_REG value is valid.
1	WO	0x0	path2_vld 1'b0: DATA2R_REG, DATA2L_REG value is invalid. 1'b1: DATA2R_REG, DATA2L_REG value is valid.
0	WO	0x0	path3_vld 1'b0: DATA3R_REG, DATA3L_REG value is invalid. 1'b1: DATA3R_REG, DATA3L_REG value is valid.

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.

20.5 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 20-2 Group 0 PDM IO Interface Description

Module Pin	Direction	Pin Name	IOMUX Setting
o_pdm_clk	O	I2S0_LRCK_RX_M0/PDM_CLk0_M0/ACODEC_ADC_SYNC/GPIO3_D4_d	GRF_GPIO3D_IOMUX_H[2:0]=3'b010
o_pdm_clk	O	I2S0_SCLK_RX_M0/PDM_CLk1_M0/ACODEC_ADC_CLK/GPIO3_D1_d	GRF_GPIO3D_IOMUX_L[6:4]=3'b010
i_pdm_data0	I	I2S0_SDI0_M0/PDM_SDI0_M0/ACODEC_DAC_DATAL/GPIO3_D6_d	GRF_GPIO3D_IOMUX_H[10:8]=3'b010
i_pdm_data1	I	I2S0_SDO3_SDI1_M0/PDM_SDI1_M0/AUDPWM_R_M0/I2C4_SDA_M1/AUDDSM_RP/GPIO4_A1_d	GRF_GPIO4A_IOMUX_L[6:4]=3'b010
i_pdm_data2	I	I2S0_SDO2_SDI2_M0/PDM_SDI2_M0/AUDPWM_L_M0/I2C4_SCL_M1/AUDDSM_RN/GPIO4_A0_d	GRF_GPIO4A_IOMUX_L[2:0]=3'b010
i_pdm_data3	I	I2S0_SDO1_SDI3_M0/PDM_SDI3_M0/ACODEC_ADC_DATA/GPIO3_D7_d	GRF_GPIO3D_IOMUX_H[14:12]=3'b010

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

Table 20-3 Group 1 PDM IO Interface Description

Module Pin	Direction	Pin Name	IOMUX Setting
o_pdm_clk	O	CIF_D12_M0/RGMII_CLK_M0/PDM_CLk0_M1/SPI1_CLK_M0/GPIO3_C0_d	GRF_GPIO3C_IOMUX_L[2:0]=3'b011
o_pdm_clk	O	CIF_D15_M0/RGMII_MDIO_M0/PDM_CLk1_M1/GPIO3_C3_d	GRF_GPIO3C_IOMUX_L[14:12]=3'b011
i_pdm_data0	I	CIF_D13_M0/RGMII_RXDV_M0/PDM_SDI0_M1/GPIO3_C1_d	GRF_GPIO3C_IOMUX_L[6:4]=3'b011
i_pdm_data1	I	CIF_D14_M0/RGMII_RXER_M0/PDM_SDI1_M1/GPIO3_C2_d	GRF_GPIO3C_IOMUX_L[10:8]=3'b011
i_pdm_data2	I	CIF_D10_M0/RGMII_RXD0_M0/PDM_SDI2_M1/SPI1_MOSI_M0/GPIO3_B6_d	GRF_GPIO3B_IOMUX_H[10:8]=3'b011
i_pdm_data3	I	CIF_D11_M0/RGMII_RXD1_M0/PDM_SDI3_M1/SPI1_MISO_M0/GPIO3_B7_d	GRF_GPIO3B_IOMUX_H[14:12]=3'b011

By configuring GRF_IOFUNC_SEL0[12] to 1'b0, the input bit stream from group 0 IO interface will be fed into PDM. Otherwise the group 1 IO interface will be selected.

20.6 Application Notes

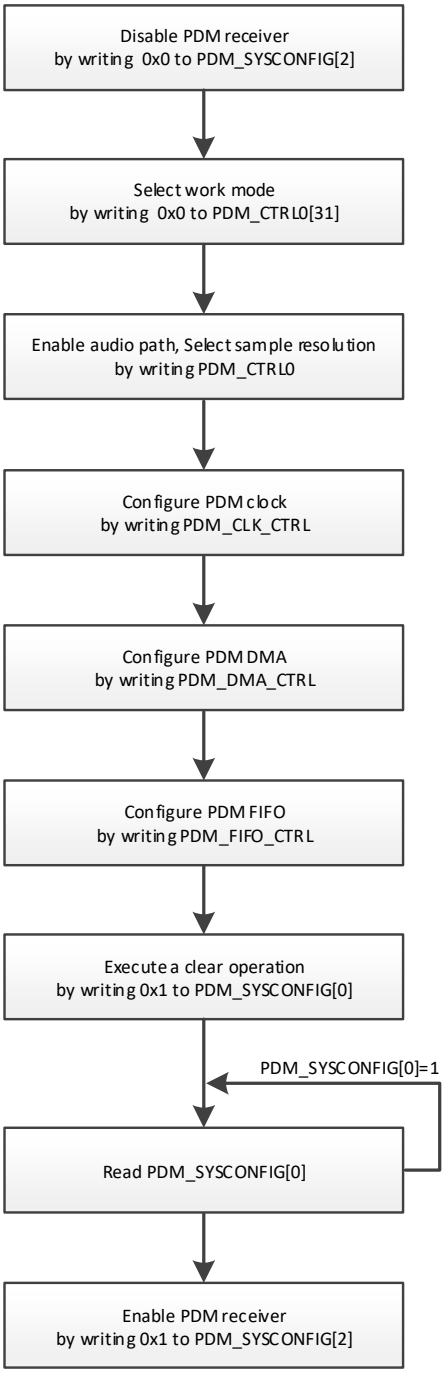


Fig.20-6 PDM Operation Flow

Chapter 21 CAN

21.1 Overview

CAN (Controller Area Network) bus is a robust vehicle bus standard designed to allow micro controllers and devices to communicate with each other in applications without a host computer. It is a message-based protocol, designed originally for multiplex electrical wiring within automobiles to save on copper, but is also used in many other contexts.
CAN controller supports the following features:

- Support CAN 2.0B protocol
- Support 32-bit APB bus
- Support transmit or receive standard frame
- Support transmit or receive extended frame
- Support transmit or receive data frame, remote frame, overload frame, error frame and frame interval
- Support transmit or receive error count
- Support ID mask
- Support bit error, bit stuffing error, form error, ack error and crc error
- Support 7 types of interrupt and all interrupt can be masked
- Support CAN controller status query
- Support capture the bit position of arbitration section where the arbitration was lost
- Support error code check
- Support self test mode
- Support single sample and three sample configurable
- Support SJW(reSynchronization Jump Width) configurable
- Support BRP(system prescaler coefficient) configurable
- Support bit timing configurable
- Support loop-back mode for self-test operation
- Support silent mode for debug
- Support receive self-transmit data mode(rxstx_mode)
- Support TX data and RX data order select
- Support RX data cover mode
- Support auto retransmission mode
- Support auto bus on after bus-off state
- Support space_rx_mode

21.2 Block Diagram

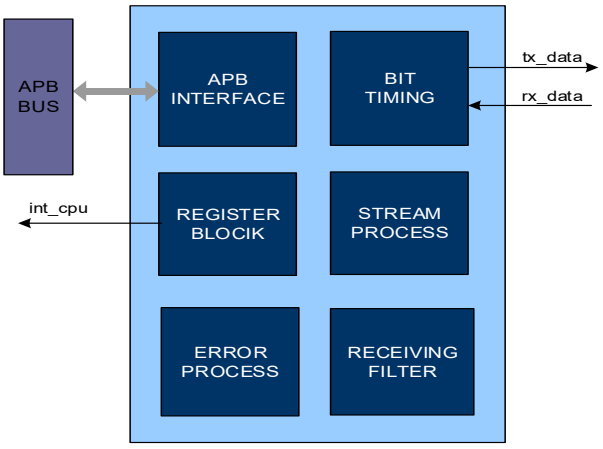


Fig. 21-1 CAN Controller Block Diagram

21.3 Function Description

21.3.1 RECEIVING FILTER

The receiving filter performs filtering using the acceptance ID register and the ID mask register. The receiving filter uses one-time filtering. After sampling all the bits of the ID, it is compared with the ID register. It is not a comparison every time a bit is sampled. And don't check the bit in the ID mask register that is 1. After all the ID bits have passed the filtering, the controller considers the frame data as the desired ID, thereby performing the next operation.

21.3.2 BIT TIMING

21.3.2.1 Bit Timing Logic

The bit timing logic controls the sampling point position through the buffer bits in the timing register to ensure the accuracy of the data sampling. The bit timing logic receives the clock frequency division signal for identification, sets the bus timing parameters, establishes synchronization parameters, and adjusts the bus transmission rate. At the same time, the bus is monitored and the message to be sent is transmitted to the bus at the set timing.

According to the provisions of the CAN protocol: the CAN bus is always at a high level when no message is sent, and the continuous recessive bit is monitored on the bus. At this time, the bus is in an idle state, and the arbitration priority is low, ready to receive data at any time. When the bus detects a transition from a recessive bit to a dominant bit, it is proved that the frame start bit starts transmitting, and the bus performs a hard synchronization in the bit start sync segment. Then, in the process of receiving the message, once the edge of the transition close to the sampling point is detected and the edge of the edge and the synchronization segment have a phase error lower than the synchronization width (SJW is taken when the value exceeds SJW), the controller executes once resynchronization, resynchronization can be performed multiple times during one data transmission.

21.3.2.2 Bit Timing Definition

The transmission time of each bit is divided into 4 parts:

$$t_{nbt} = t_{sync\_seg} + t_{prop\_seg} + t_{phase\_seg1} + t_{phase\_seg2}$$

tsync_seg, tprop_seg, tphase_seg1 and tphase_seg2 are integer multiples of the unit time(tsclk), and this unit time is a specific multiple of the system clock, which is determined by the division factor BRP in the bus timing register: 1. The counter is obtained by the BRP operation; 2. The system clock clk is counted by the counter; 3. When the limit is reached, a clock with a period of tsclk is generated.

In the system design, considering the cyclical occurrence of each time period, a state machine with three states is used in the bit timing design. The three states correspond to the sync segment(sync_seg) and the phase buffer segment 1 (phase_seg1=phase_seg1+prog_seg) and the phase buffer segment 2 (phase_seg2), respectively. Where PHASE_SEG1 range: 1~16; PHASE_SEG2 range: 1~8; BRP range: 1~64.

$$t_{sync\_seg} = 1 \times t_{sclk}$$

$$t_{prop\_seg} + t_{phase\_seg1} = t_{sclk} \times (8 \times TSEG1.3 + 4 \times TSEG1.2 + 2 \times TSEG1.1 + TSEG1.0 + 1)$$

$$t_{phase\_seg2} = t_{sclk} \times (4 \times TSEG2.2 + 2 \times TSEG2.1 + TSEG1.0 + 1)$$

Define a counter to count tsclk. When the count reaches the TSEG1, TSEG2 defined in the bus timing register and the length of the synchronization segment defined in the design, the system will generate the corresponding transition conditions: go_seg1, go_seg2, go_sync.By judging these transition conditions, the control state machine cycles through the above three states.

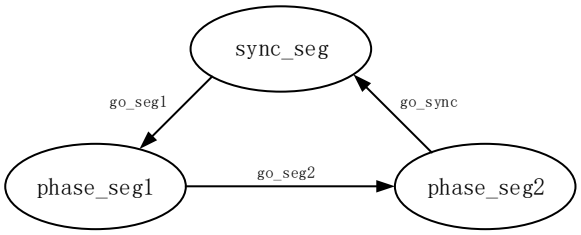


Fig. 21-2 Bit timing FSM

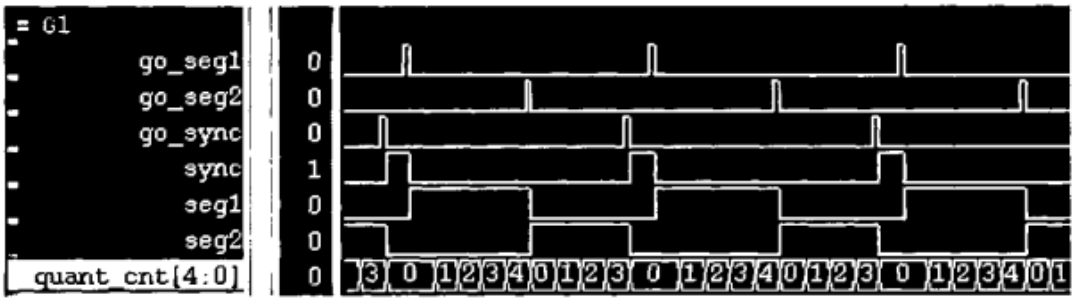
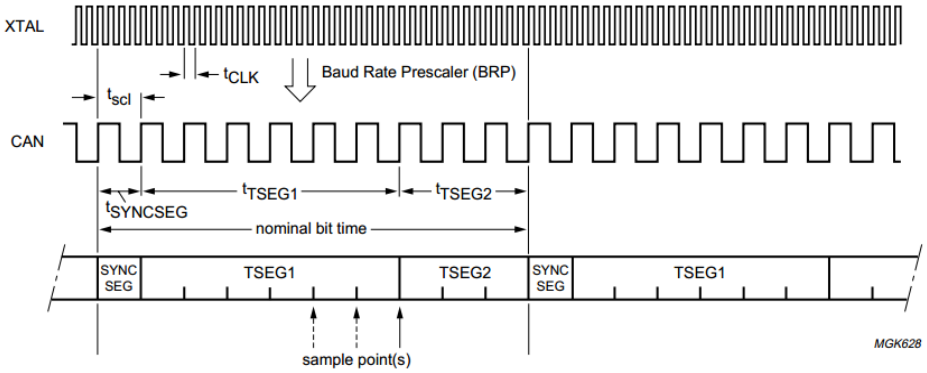


Fig. 21-3 Bit timing waveform diagram

21.3.2.3 Sampling Point and Sending Point

Sampling Point: According to the agreement, the sample point sample_point should be between phase_seg1 and phase_seg2. It is designed to be in the position of phase_seg2 synchronization. The sampling pulse width is defined as one system clock cycle. Considering the accuracy of sampling, you can use the method of taking three samples to take the mean. The interval between each sample point is one tsclk.



Possible values are BRP = 000001, TSEG1 = 0101 and TSEG2 = 010.

Fig. 21-4 Three sampling diagram

Sending Point: The location of the transmission point(tx_point) should be at the beginning of each bit time according to the protocol, and it is designed to synchronize it with the go_sync signal. In addition, if it is in sync or resynchronize, tx_point will be valid immediately.

21.3.2.4 Bit Synchronization

With the synchronization method, one or more nodes that satisfy the synchronization condition align their synchronization segments with the transmission data on the bus at a specific time. Synchronization occurs on the 1-to-0 transition edge in order to control the distance from the transition edge to the sample point. Two synchronization methods are defined in the CAN bus communication protocol: hard synchronization and resynchronization.

Hard_sync: All nodes must be synchronized to the leading edge of the starting frame of the node that first started transmitting the message. At the beginning of each frame of data, a synchronization action is taken between the nodes.
Hard synchronization implementation method: determine whether the bus state meets the frame start condition specified in the protocol, and the node is not in the state of the data to be transmitted. At this time, a hard synchronization flag signal hard_sync is generated, and the pulse width is a system clock; The hard_sync entry is added to the control condition of the go_seg1 signal, and go_seg1 is valid immediately when the hard_sync signal is '1'. Therefore, once the hard synchronization condition is met, the system enters the phase_seg1 segment, thereby achieving synchronization.

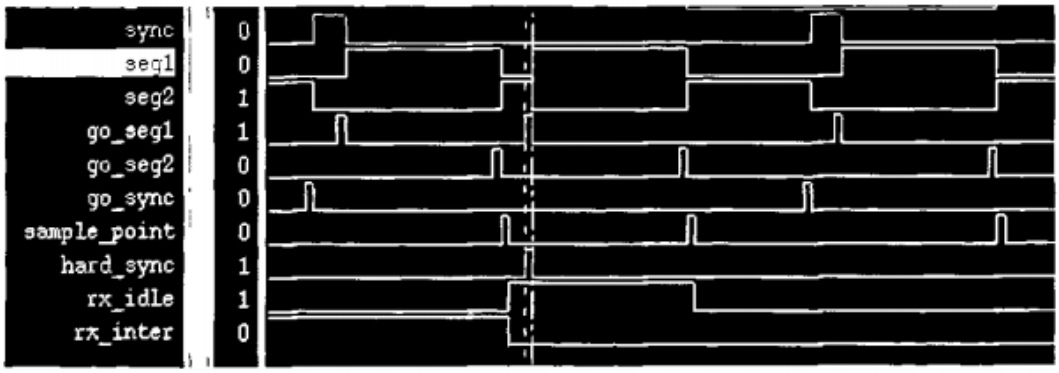


Fig. 21-5 Hard_sync waveform diagram

Resynchronization: In addition to generating hard synchronization at the beginning of each frame of data, the CAN bus communication protocol also specifies the transmission of data for each frame. When the timing coordination between the nodes is not ideal, it will be resynchronized, so that the cooperation between the nodes is in a good state.
For the receiver, the bus changes it received should occur in the sync segment. Once the receiver's status violates this rule, the receiver will perform a resynchronization.
Two situations that require resynchronization:

1. The jump of the bus value '1' to '0' occurs between the sync segment and the sample point, which is a delay jump.
2. The jump of the bus value '1' to '0' occurs between the sample point and the sync segment, which is an early jump.

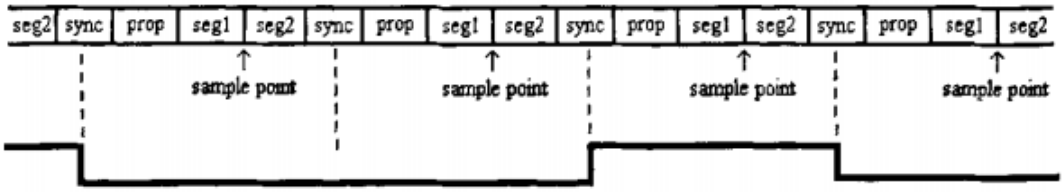


Fig. 21-6 Resynchronization

Resynchronization method:

1. Condition for resynchronization: the receiver is in a mode of receiving data (not inter-frame space or bus idle), and a resynchronization flag signal ‘resync’ is generated when a jump of the bus value '1' to '0' occurs.
2. According to the position of the resynchronization flag signal, it is determined that the abnormal jump is a delay jump or an early jump.
3. Counting the delay or the length of the advance by the counter.
4. In this bit, phase_seg1 is added with a delay time or phase_seg2 is subtracted from the advance time to obtain a new phase buffer time for the bit, thereby achieving synchronization.

21.3.3 STREAM PROCESS

21.3.3.1 Data Buffering

The data cache includes two parts: a transmit buffer and a receive buffer. The data to be sent on the CAN bus is loaded into the memory area. This memory area is called the "transmission buffer"; The data received from the CAN bus is also stored in the memory area, which is called the "receiver buffer". The size of the buffer is 13 bytes, including 1 byte frame information, 2 to 4 byte identifiers (standard or extended frame), and 8 bytes of data.

After the CPU configures the transmission of the data information (frame information, identification code, data), the request bit is enabled. The data buffer sends the data to be sent to the bit stream processor, and splices into a parallel complete data to be transmitted (including the CRC check code) according to the current frame information.

The received message is placed in the receiving buffer. After receiving, the receiving completion interrupt will be generated. After receiving the interrupt, the CPU will read the data in the receiving buffer, and set CMD[1] to clear the receiving buffer, and STATE[0] will be 0. If the CPU does not read in time, STATE[1] will remain at 1. If a new message is received at this time, overflow will occur.

21.3.3.2 Receive Data

According to the bus protocol, taking the data frame of the extended frame format as an example, the receiving data state machine is as follows:

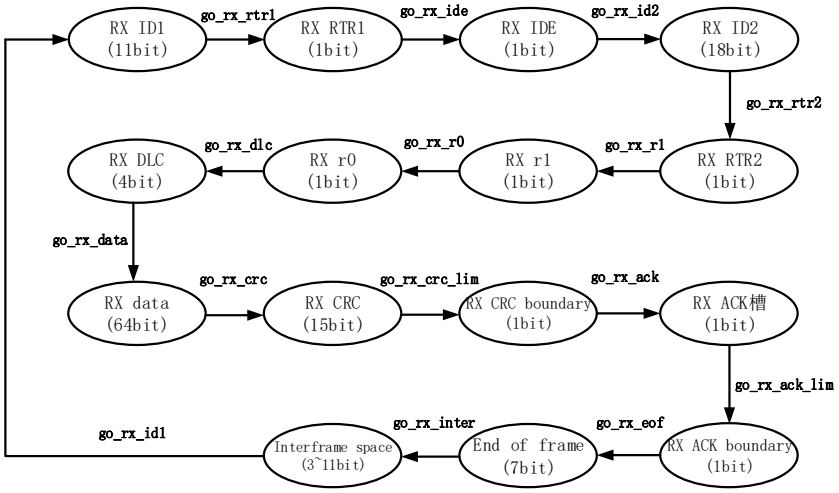


Fig. 21-7 Receive data state diagram

Each state of the state machine corresponds to which "field" of the frame data the receiver is in when receiving data, and only one of all state flags is valid at the same time. The method of implementation is as follows:

Define a bit counter(bit_cnt), add one to each sample point (provided by the bit timing module). At the same time, according to the current state of the state machine, it is determined in this state that the bit counter needs to count to what value (determined by the length of each field defined in the protocol) to generate a jump signal to the next state transition. Whenever a hopping signal is generated, the counter is cleared and ready to begin counting for the next state.

The receiving state machine ensures that the bus controller can automatically recognize the current receiving state when receiving data. After that, the data of each data field needs to be stored in the corresponding register according to different states, and then the next operation is performed.

21.3.3.3 Transmit Data

Parallel data is serially transmitted to the bus according to a specific timing. The sending point is the tx_point. The transmit counter is self-added in the valid data portion of the data frame, and is cleared when one frame of data is transmitted or the bus is in an error state.

The implementation is as follows:

1. Combine all the data to be sent into a completely parallel data chain according to the currently set frame type.

2. using tx_pointer as a data link pointer, obtaining a serial data signal (tx_bit) to be transmitted.

3. Determine what type of data to send according to the current state, and determine the data tx_next to be sent at the next transmission time.

4. Synchronize the data transmission signal tx with the tx_next through the transmission point tx_point to obtain a true transmission signal tx.

21.3.3.4 Bit Stuffing

Bit stuffing is a function set to prevent burst errors. Add a bit of inversion data when the same level lasts 5 bits.

Sending unit: The data between the SOF and CRC segments when transmitting data frames and remote frames. If the same level lasts for 5 bits, the next bit (bit 6) is inserted with the level of the 1st bit and the first 5 bits.

Receiving unit: Data between SOF and CRC segments when receiving data frames and remote frames. If the same level lasts for 5 digits, the next bit (bit 6) needs to be deleted and received. If the 6th bit is the same level as the first 5 bits, it will be treated as an error and an error frame will be sent.

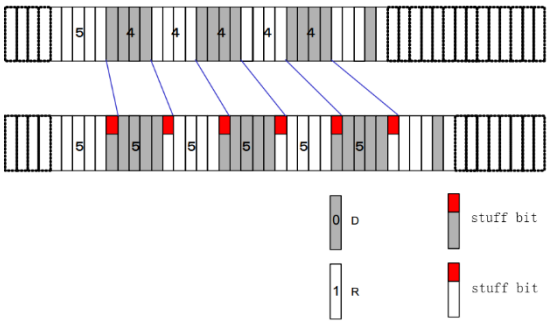


Fig. 21-8 Bit Stuffing

21.3.4 ERROR PROCESS

There are five types of errors. Multiple errors can occur at the same time.

21.3.4.1 BIT ERROR

A unit that is sending a bit on the bus also monitors the bus. A BIT ERROR has to be detected at that bit time, when the bit value that is monitored is different form the bit value that is sent. An exception is the sending of a 'recessive' bit during the stuffed bit stream of the ARBITRATION FIELD of during the ACK SLOT. Then no BIT ERROR occurs when a 'dominant' bit is monitored. A TRANSMITTER sending a PASSIVE ERROR FLAG and detecting a 'dominant' bit does not interpret this as a BIT ERROR.

21.3.4.2 BIT STUFF ERROR

A STUFF ERROR has to be detected at the bit time of the 6th consecutive equal bit level in a message field that should be coded by the method of bit stuffing.

21.3.4.3 FORM ERROR

A FORM ERROR has to be detected when a fixed-form bit field contains one or more illegal bits.

21.3.4.4 ACK ERROR

An ACK ERROR has to be detected by a transmitter whenever it does not monitor a 'dominant' bit during the ACK SLOT.

21.3.4.5 CRC ERROR

The CRC sequence consists of the result of the CRC calculation by the transmitter. The receivers calculate the CRC in the same way as the transmitter. A CRC ERROR has to be detected, if the calculated result is not the same as that received in the CRC sequence.

21.4 Register Description

21.4.1 Internal Address Mapping

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

21.4.2 Registers Summary

Name	Offset	Size	Reset Value	Description
CAN_MODE	0x0000	W	0x00000000	CAN controller working mode configure register
CAN_CMD	0x0004	W	0x00000000	CAN command register
CAN_STATE	0x0008	W	0x00000000	CAN state register
CAN_INT	0x000c	W	0x00000000	Interrupt state register
CAN_INT_MASK	0x0010	W	0x00000000	Interrupt enable registers
CAN_BITTIMING	0x0018	W	0x00000000	Bit timing configure register
CAN_ARBITFAIL	0x0028	W	0x00000000	Arbit fail code register
CAN_ERROR_CODE	0x002c	W	0x00000000	Error code register
CAN_RXERRORCNT	0x0034	W	0x00000000	Receive error counter
CAN_TXERRORCNT	0x0038	W	0x00000000	Transmit error counter
CAN_IDCODE	0x003c	W	0x00000000	CAN controller's identifier
CAN_IDMASK	0x0040	W	0x00000000	Identification code bit mask register
CAN_TXFRAMEINFO	0x0050	W	0x00000000	TX frame information configuration register
CAN_TXID	0x0054	W	0x00000000	CAN controller transmit ID
CAN_TXDATA1	0x0058	W	0x00000000	CAN controller transmit DATA1
CAN_TXDATA2	0x005c	W	0x00000000	CAN controller transmit DATA2
CAN_RXFRAMEINFO	0x0060	W	0x00000000	RX frame information register.This register needs to be read(clear) before receiving the next frame.
CAN_RXID	0x0064	W	0x00000000	CAN controller receive ID.This register needs to be read(clear) before receiving the next frame.
CAN_RXDATA1	0x0068	W	0x00000000	Receive Data Reg1.This register needs to be read(clear) before receiving the next frame.
CAN_RXDATA2	0x006c	W	0x00000000	Receive Data Reg2.This register needs to be read(clear) before receiving the next frame.
CAN_RTL_VERSION	0x0070	W	0x00000021	CAN RTL version

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

21.4.3 Detail Registers Description

CAN_MODE

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:13	RO	0x00000	reserved
12	RW	0x0	space_rx_mode Interframe Spaceing RX Mode. 1'b0 : Enable. 1'b1 : Disable.
11	RW	0x0	auto_bus_on Auto Bus On Enable. 1'b0 : After the RKCAN has entered bus_off state, the software can start a bus_off_recovery sequence by resetting the work_mode to 0. 1'b1 : Automatic reset TEC/REC to bus_on after 128 occurrence of 11 consecutive recessive bits have been monitored on the bus.

Bit	Attr	Reset Value	Description
10	RW	0x0	auto_retx_mode Auto Retransmission Mode. 1'b0 : Disable. RKCAN return to idle state after transmit DATA/RTR frame failed, and set tx_req to 0. 1'b1 : Enable. The RKCAN automatically retransmit frames which have lost arbitration or have been disturbed by errors during transmission.
9	RW	0x0	ovld_mode Overload Mode. 1'b0: overload lite mode. 1'b1: overload extended mode. RKCAN
8	RW	0x0	cover_mode Rx Data Cover Mode. 1'b0 : RKCAN can't receive new frame before receive registers cleared. 1'b1 : RKCAN can receive new frame before receive registers cleared, and new rxdata will cover old rxdata.
7	RW	0x0	rxsort_mode RX Data Sort Mode. 1'b0: The data received first is placed in the lower address. 1'b1: The data received first moves to the upper address, and the data received later is placed at the lower address.
6	RW	0x0	txorder_mode TX Data Order Mode. 1'b0: tx_data1[7:0] --> tx_data1[15:8] --> tx_data1[23:16] --> tx_data1[31:24] -->tx_data2[7:0] --> tx_data2[15:8] --> tx_data2[23:16] --> tx_data2[31:24]. 1'b1: tx_data2[31:0] --> tx_data1[31:0].
5	RW	0x0	rxstx_mode Receive Self Transmit data mode. 1'b0: Disable; 1'b1: Enable. When the RKCAN sends data, it can also receive the data sent by itself.
4	RW	0x0	lback_mode Loopback Mode. 1'b0: Disable. 1'b1: Enable.
3	RW	0x0	silent_mode Silent Mode. 1'b0: Disable. 1'b1: Enable.
2	RW	0x0	self_test Self check ACK slot when TX frame. 1'b0: Normal 1'b1: Self test mode. When in self test mode, the receiver does not need to return an ACK signal when receiving data. Therefore, no ack_error will occur in self test mode. When in 'lback_mode' or 'silent_mode', it need to enable 'self_test' mode when sending frame.
1	RO	0x0	reserved
0	RW	0x0	work_mode Work Mode. 1'b0: Reset mode, CAN stop transmit and registers can be configured. 1'b1: Work mode, CAN enter the working mode, RX/TX data.

CAN_CMD

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	WO	0x0	tx_req Transmit request enable 1'b0: Disable; 1'b1: Enable. When 'tx_req' is enable, the RKCAN is in the transmit mode and cannot receive frames from other CANs. When the RKCAN transmission frame complete, 'tx_req' is automatically cleared to 0. Also, when RKCAN is set 'work_mode' to 'reset_mode', tx_req' is cleared to 0.

CAN_STATE

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:6	RO	0x0000000	reserved
5	RO	0x0	bus_off_state 1'b0: None 1'b1: Bus off state: When the error counter is incremented to 255, the CAN controller will enter the bus off state, generate an error warning interrupt and enter reset mode, waiting for the CPU to restart. (rx/tx_err_cnt >=32'd255)
4	RO	0x0	error_warning_state 1'b0: None 1'b1: Error state: At least one error counter has reached the error warning threshold (rx/tx_err_cnt >=32'd96)
3	RO	0x0	tx_period 1'b0: Not in transmitting state 1'b1: CAN controller is in the transmitting state
2	RO	0x0	rx_period 1'b0: Not in receiving state 1'b1: CAN controller is in the receiving state
1	RO	0x0	tx_buffer_full Transmit buffer full flag bit 1'b0: Not full 1'b1: TX buffer is full. There is data in buffer waiting to be sent or being sent.
0	RO	0x0	rx_buffer_full Receive buffer full flag bit 1'b0: Not full 1'b1: RX buffer is full. A complete message is stored in the buffer.

CAN_INT

Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:7	RO	0x0000000	reserved
6	WO	0x0	error_int 1'b0: None 1'b1: CAN bus error interrupt. This interrupt is generated when a bus error is detected. Error details can refer to the ERROR_CODE register. Write 1 then clear.

Bit	Attr	Reset Value	Description
5	WO	0x0	tx_arbit_fail_int 1'b0: None 1'b1: Arbitration loss interrupt. This interrupt is generated when the loss of arbitration is turned into a receiver. Write 1 then clear.
4	WO	0x0	passive_error_int 1'b0: None 1'b1: Passive error interrupt. This interrupt is generated when the controller enters an error passive state (at least one error counter reaches 127) or returns from the error passive state to the error active state. Write 1 then clear.
3	WO	0x0	rx_buffer_overflow_int Receive buffer overflow interrupt 1'b0: None 1'b1: This interrupt is generated when the data in the receiving buffer is not read and a new message is received. Write 1 then clear.
2	WO	0x0	error_warning_int Error warning interrupt 1'b0: None 1'b1: error_warning_int. This interrupt is generated when the error_warning_state bits change. Write 1 then clear.
1	WO	0x0	tx_finish_int Transmit finish interrupt 1'b0: None 1'b1: tx_finish_int. CAN controller sends the message and the buffer is empty. Write 1 then clear.
0	WO	0x0	rx_finish_int Receive finish interrupt 1'b0: None 1'b1: RX finish int. CAN controller has received the message and the rx_buffer is full. Write 1 then clear.

CAN_INT_MASK

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:7	RO	0x0000000	reserved
6	RW	0x0	error_int 1'b0: Unmasked 1'b1: Masked
5	RW	0x0	tx_arbit_fail_int 1'b0: Unmasked 1'b1: Masked
4	RW	0x0	passive_error_int 1'b0: Unmasked 1'b1: Masked
3	RW	0x0	rx_buffer_overflow_int 1'b0: Unmasked 1'b1: Masked
2	RW	0x0	error_warning_int_mask 1'b0: Unmasked 1'b1: Masked
1	RW	0x0	tx_finish_int_mask 1'b0: Unmasked 1'b1: Masked

Bit	Attr	Reset Value	Description
0	RW	0x0	rx_finish_int_mask 1'b0: Unmasked 1'b1: Masked

CAN_BITTIMING

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:17	RO	0x0000	reserved
16	RW	0x0	sample_mode CAN controller sampling mode configuration register. 1'b0: Single sample mode 11'b1: Three sample mode
15:14	RW	0x0	sjw SJW: reSynchronization Jump Width Each unit has a synchronization error due to a clock frequency deviation or a transmission delay. SJW is to compensate for the maximum value of this error.
13:8	RW	0x00	brp brp: system prescaler coefficient Tsclk = 2 x Tclk x (brp + 1).
7	RO	0x0	reserved
6:4	RW	0x0	tseg2 Phase buffer segment 2 Tphase_seg2 = Tsclk x (tseg2 + 1)
3:0	RW	0x0	tseg1 Phase buffer segment 1 Tphase_seg1 = Tsclk x (tseg1 + 1)

CAN_ARBITFAIL

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:7	RO	0x0000000	reserved
6:0	RO	0x00	arbit_fail_code This register indicates the bit position of arbitration section where the arbitration was lost.

CAN_ERROR_CODE

Address: Operational Base + offset (0x002c)

Bit	Attr	Reset Value	Description
31:25	RO	0x00	reserved
24:22	RO	0x0	error_type Error type: 3'b000: BIT ERROR 3'b001: BIT STUFF ERROR 3'b010: FORM ERROR 3'b011: ACK ERROR 3'b100: CRC ERROR
21	RO	0x0	error_direction 1'b0: TX error 1'b1: RX error

Bit	Attr	Reset Value	Description
20:14	RW	0x00	tx_error_position Indicate transmit error position. 7'b000_0000 : TRANSMIT_IDLE. 7'b000_0010 : TRANSMIT_SOF_DATA. 7'b000_0100 : TRANSMIT_CRC. 7'b000_1000 : TRANSMIT_ACK_ROF. 7'b001_0000 : TRANSMIT_ACK. 7'b010_0000 : TRANSMIT_ERROR. 7'b100_0000 : TRANSMIT_OVERLOAD.
13:0	RO	0x0000	rx_error_position Indicate receive error position 14'b00000000000000: RECEIVE_IDLE 14'b00000000000010: RECEIVE_ID1 14'b00000000000100: RECEIVE_RTR1~IDLE 14'b00000000001000: RECEIVE_ID2 14'b00000000010000: RECEIVE_RTR2~R1R0 14'b00000000100000: RECEIVE_R0 14'b00000001000000: RECEIVE_DLC 14'b00000010000000: RECEIVE_DATA 14'b00000100000000: RECEIVE_CRC 14'b00001000000000: RECEIVE_CRC_LIM 14'b00010000000000: RECEIVE_ACK 14'b00100000000000: RECEIVE_ACK_LIM 14'b01000000000000: RECEIVE_EOF 14'b10000000000000: RECEIVE_SPACE

CAN_RXERRORCNT

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RO	0x00	rx_err_cnt Receive Error Counter(REC). Actual state of the Receive Error Counter. Value between 0 and 127.

CAN_TXERRORCNT

Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:9	RO	0x000000	reserved
8:0	RO	0x000	tx_err_cnt Transmit Error Counter(TEC). Actual state of the Transmit Error Counter. Value between 0 and 255.

CAN_IDCODE

Address: Operational Base + offset (0x003c)

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28:0	RW	0x00000000	id_code CAN controller ID code: Standard frame ID: id_code[10:0]; Extended frame ID: id_code[28:0].

CAN_IDMASK

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28:0	RW	0x00000000	id_mask [0]: Unmasked [1]: Masked

CAN_TXFRAMEINFO

Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7	RW	0x0	txframe_format 1'b0: Standard frame 1'b1: Extended frame
6	RW	0x0	tx_rtr 1'b0: Data frame 1'b1: Remote frame
5:4	RO	0x0	reserved
3:0	RW	0x0	txdata_length Transmit data length configure register.(unit:byte) 4'b0000: 0byte 4'b0001: 1byte 4'b0010: 2byte 4'b0011: 3byte 4'b0100: 4byte 4'b0101: 5byte 4'b0110: 6byte 4'b0111: 7byte 4'b1000: 8byte others: reserved

CAN_TXID

Address: Operational Base + offset (0x0054)

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28:0	RW	0x00000000	tx_id can_tx_id[28:0]

CAN_TXDATA1

Address: Operational Base + offset (0x0058)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	tx_data1 tx_data1

CAN_TXDATA2

Address: Operational Base + offset (0x005c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	tx_data2 tx_data2

CAN_RXFRAMEINFO

Address: Operational Base + offset (0x0060)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7	WO	0x0	rxframe_format 1'b0: Standard frame 1'b1: Extended frame

Bit	Attr	Reset Value	Description
6	WO	0x0	rx_rtr 1'b0: Data frame 1'b1: Remote frame
5:4	RO	0x0	reserved
3:0	WO	0x0	rxdata_length Receive data length register.(unit:byte) 4'b0000: 0byte 4'b0001: 1byte 4'b0010: 2byte 4'b0011: 3byte 4'b0100: 4byte 4'b0101: 5byte 4'b0110: 6byte 4'b0111: 7byte 4'b1000: 8byte others: reserved

CAN_RXID

Address: Operational Base + offset (0x0064)

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28:0	WO	0x00000000	rx_id rx_id[28:0]

CAN_RXDATA1

Address: Operational Base + offset (0x0068)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	rx_data1 rx_data1[31:0]

CAN_RXDATA2

Address: Operational Base + offset (0x006c)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	rx_data2 rx_data2[31:0]

CAN_RTL_VERSION

Address: Operational Base + offset (0x0070)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000021	version CAN rtl version = 32'h21

21.5 Application Notes

21.5.1 Controller Initialization Flow

The controller must configure the registers after power-up or hardware reset. During the operation of the controller, a software reset request may be sent and reconfigured (re-initialized) as shown below. After the initialization is completed, the controller enters the working mode, sends the frame to be sent to the buffer, and then sets the "send request" flag of the command register to start transmitting.

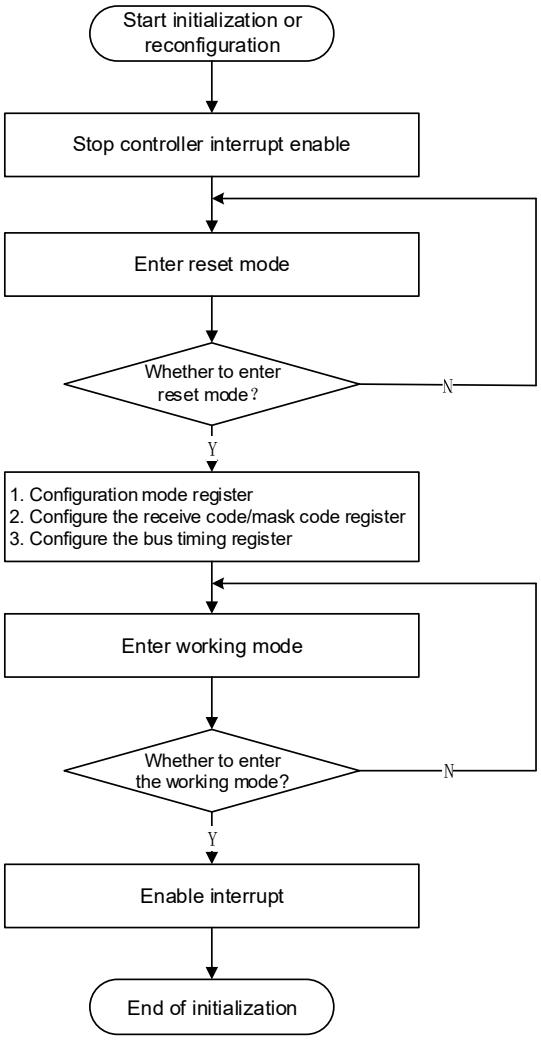


Fig. 21-9 Initialization Flow

21.5.2 Loop-back Mode

The controller must configure the registers after power-up or hardware reset. During the operation of the controller, a software reset request may be sent and reconfigured (re-initialized) as shown below.

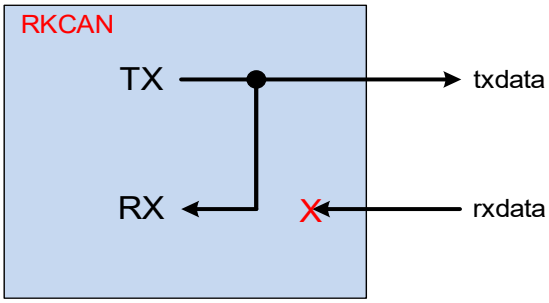


Fig. 21-10 Loop-back Mode

21.5.3 Silent Mode

The controller must configure the registers after power-up or hardware reset. During the operation of the controller, a software reset request may be sent and reconfigured (re-initialized) as shown below.

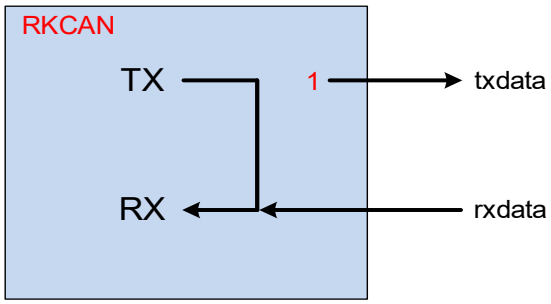


Fig. 21-11 Silent Mode

21.5.4 RXSTX Mode

- rxstx_mode=0, RKCAN cannot receive the data sent by itself
- rxstx_mode=1, RKCAN can receive the data sent by itself, and when the ID of the frame is the same as the ID_CODE of RKCAN itself, RKCAN will store the frame information into CAN_RXDATA1/2

21.5.5 TXORDER Mode

The TXORDER Mode is used to determine the order in which data is sent.

- txorder_mode=0, Data transmission order is "tx_data1[7:0] --> tx_data1[15:8] --> tx_data1[23:16] --> tx_data1[31:24] -->tx_data2[7:0] --> tx_data2[15:8] --> tx_data2[23:16] --> tx_data2[31:24]"
- txorder_mode=1, Data transmission order is "tx_data2[31:0] --> tx_data1[31:0]"

21.5.6 RXSORT Mode

The TXSORT Mode is used to determine how the received data is stored.

- rxsort_mode=0: The data received first is placed in the lower address
- rxsort_mode=1: The data received first moves to the upper address, and the data received later is placed at the lower address

21.5.7 Cover Mode

- cover_mode=1'b0 : RKCAN can't receive new frame before receive registers cleared
- cover_mode=1'b1 : RKCAN can receive new frame before receive registers cleared, and new rxdata will cover old rxdata

21.5.8 Overload Mode

According to the CAN protocol, a CAN transceiver will generate an OVERLOAD frame in three cases:

CASE1: When the CAN is used as a receiver, if it is necessary to delay the next frame.
CASE2: Detection of a dominant bit at the first and second bit of Intermission.
CASE3: If an CAN node samples a dominant bit at the eight bit (the last bit) of an ERROR DELIMITER or OVERLOAD DELIMITER.

- overload_mode=0: Overload Lite Mode: RKCAN only generates overload frames in the CASE2
- overload_mode=1: Overload Extended Mode: RKCAN can generates overload frames in all three cases

21.5.9 Auto_retx Mode

- auto_retx_mode=0 : Disable. RKCAN return to idle state after transmit DATA/RTR frame failed, and set tx_req to 0
- auto_retx_mode=1 : Enable. The RKCAN automatically retransmit frames which have lost arbitration or have been disturbed by errors during transmission

21.5.10 Auto Bus On

- auto_bus_on=0 : After the RKCAN has entered bus_off state, the software can start a bus_off_recovery sequence by resetting the work_mode to 0
- auto_bus_on=1 : Automatic reset TEC/REC to bus_on after 128 occurrence of 11 consecutive recessive bits have been monitored on the bus

Chapter 22 Pulse Width Modulation (PWM)

22.1 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.

22.2 Block Diagram

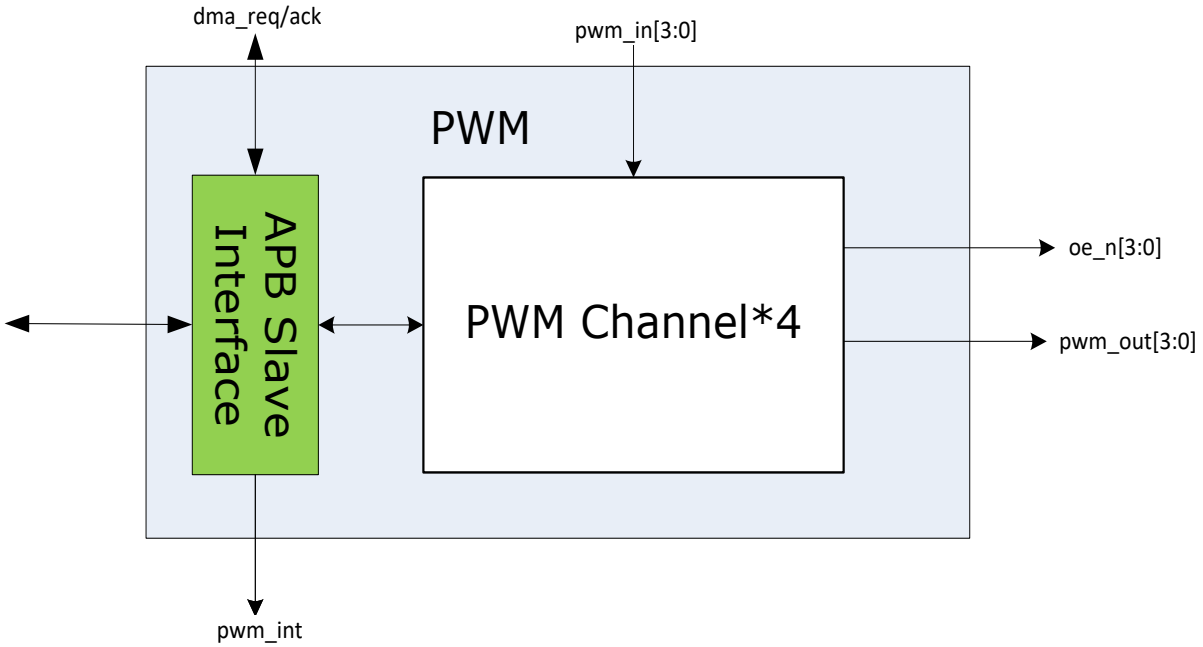


Fig. 22-1 PWM Block Diagram

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.

22.3 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.

22.3.1 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.

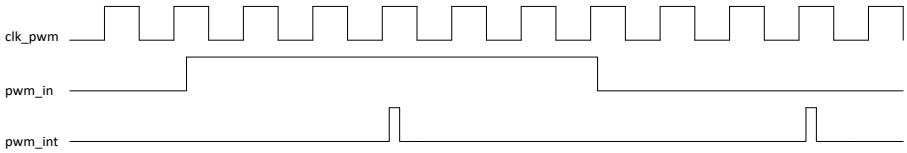


Fig. 22-2 PWM Capture Mode

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, user can poll the status to judge whether a power key access.

22.3.2 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.

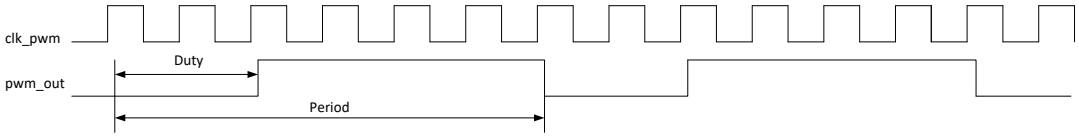


Fig. 22-3 PWM Continuous Left-aligned Output Mode

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.

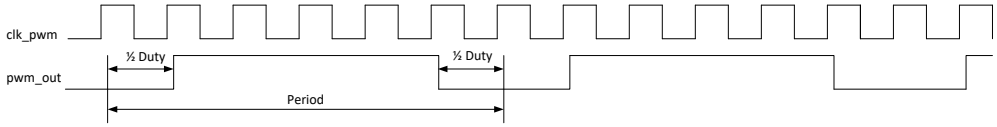


Fig. 22-4 PWM Continuous Center-aligned Output Mode

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).

22.3.3 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.

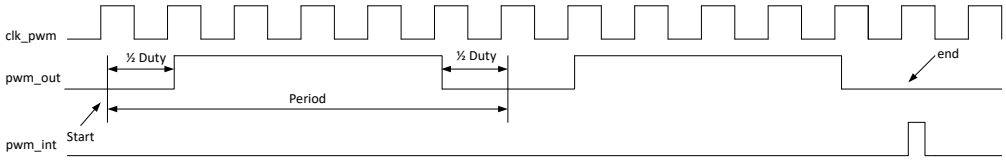


Fig. 22-5 PWM One-shot Center-aligned Output Mode

22.4 Register Description

22.4.1 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

Name	Offset	Size	Reset Value	Description
PWM_PWM1_CNT	0x0010	W	0x00000000	PWM Channel 1 Counter Register
PWM_PWM1_PERIOD_HPR	0x0014	W	0x00000000	PWM Channel 1 Period Register/High Polarity Capture Register
PWM_PWM1_DUTY_LPR	0x0018	W	0x00000000	PWM Channel 1 Duty Register/Low Polarity Capture Register
PWM_PWM1_CTRL	0x001C	W	0x00000000	PWM Channel 1 Control Register
PWM_PWM2_CNT	0x0020	W	0x00000000	PWM Channel 2 Counter Register
PWM_PWM2_PERIOD_HPR	0x0024	W	0x00000000	PWM Channel 2 Period Register/High Polarity Capture Register
PWM_PWM2_DUTY_LPR	0x0028	W	0x00000000	PWM Channel 2 Duty Register/Low Polarity Capture Register
PWM_PWM2_CTRL	0x002C	W	0x00000000	PWM Channel 2 Control Register
PWM_PWM3_CNT	0x0030	W	0x00000000	PWM Channel 3 Counter Register
PWM_PWM3_PERIOD_HPR	0x0034	W	0x00000000	PWM Channel 3 Period Register/High Polarity Capture Register
PWM_PWM3_DUTY_LPR	0x0038	W	0x00000000	PWM Channel 3 Duty Register/Low Polarity Capture Register
PWM_PWM3_CTRL	0x003C	W	0x00000000	PWM Channel 3 Control Register
PWM_INTSTS	0x0040	W	0x00000000	Interrupt Status Register
PWM_INT_EN	0x0044	W	0x00000000	Interrupt Enable Register
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

Name	Offset	Size	Reset Value	Description
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
PWM_PWRMATCH_VALUE_9	0x00BC	W	0x00000000	Power Key Match Value 9 Register
PWM_PWM3_PWRCAPTURE_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, **DW**-Double WORD (64 bits) access*

22.4.2 Detail Registers Description

PWM_PWM0_CNT

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	cnt The 32-bit indicates current value of PWM Channel 0 counter. The counter runs at the rate of PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM0_PERIOD_HPR

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	period_hpr If PWM is operated at the continuous mode or one-shot mode, this value defines the period of the output waveform. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the bit [31:1] is taken into account and bit [0] always considered as 0. If PWM is operated at the capture mode, this value indicates the effective high polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM0_DUTY_LPR

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	duty_lpr If PWM is operated at the continuous mode or one-shot mode, this value defines the duty cycle of the output waveform. The PWM starts the output waveform with duty cycle. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the [31:1] is taken into account. If PWM is operated at the capture mode, this value indicates the effective low polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM0_CTRL

Address: Operational Base + offset (0x000C)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	rpt This field defines the repeated effective periods of output waveform in one-shot mode. The value N means N+1 repeated effective periods.
23:16	RW	0x00	scale This field defines the scale factor applied to prescaled clock. The value N means the clock is divided by 2*N. If N is 0, it means that the clock is divided by 512(2*256).
15	RO	0x0	reserved
14:12	RW	0x0	prescale This field defines the prescale factor applied to input clock. The value N means that the input clock is divided by 2^N.
11	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 prescaled 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 PWM0_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_PWM1_CNT

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	cnt The 32-bit indicates current value of PWM Channel 1 counter. The counter runs at the rate of PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM1_PERIOD_HPR

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	period_hpr If PWM is operated at the continuous mode or one-shot mode, this value defines the period of the output waveform. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the bit [31:1] is taken into account and bit [0] always considered as 0. If PWM is operated at the capture mode, this value indicates the effective high polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM1_DUTY_LPR

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	duty_lpr If PWM is operated at the continuous mode or one-shot mode, this value defines the duty cycle of the output waveform. The PWM starts the output waveform with duty cycle. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the [31:1] is taken into account. If PWM is operated at the capture mode, this value indicates the effective low polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM1_CTRL

Address: Operational Base + offset (0x001C)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	rpt This field defines the repeated effective periods of output waveform in one-shot mode. The value N means N+1 repeated effective periods.
23:16	RW	0x00	scale This field defines the scale factor applied to prescaled clock. The value N means the clock is divided by 2*N. If N is 0, it means that the clock is divided by 512(2*256).
15	RO	0x0	reserved
14:12	RW	0x0	prescale This field defines the prescale factor applied to input clock. The value N means that the input clock is divided by 2^N.
11	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 prescaled 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

Bit	Attr	Reset Value	Description
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 PWM1_CTRL.rpt. 2'b01: Continuous mode. PWM produces the waveform continuously. 2'b10: Capture mode. PWM measures the cycles of high/low polarity of input waveform. 2'b11: Reserved
0	RW	0x0	pwm_en 1'b0: Disabled 1'b1: Enabled If the PWM is worked in the one-shot mode, this bit will be cleared at the end of operation.

PWM_PWM2_CNT

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	cnt The 32-bit indicates current value of PWM Channel 2 counter. The counter runs at the rate of PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM2_PERIOD_HPR

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	period_hpr If PWM is operated at the continuous mode or one-shot mode, this value defines the period of the output waveform. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the bit [31:1] is taken into account and bit [0] always considered as 0. If PWM is operated at the capture mode, this value indicates the effective high polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM2_DUTY_LPR

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	duty_lpr If PWM is operated at the continuous mode or one-shot mode, this value defines the duty cycle of the output waveform. The PWM starts the output waveform with duty cycle. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the [31:1] is taken into account. If PWM is operated at the capture mode, this value indicates the effective low polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM2_CTRL

Address: Operational Base + offset (0x002C)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	rpt This field defines the repeated effective periods of output waveform in one-shot mode. The value N means N+1 repeated effective periods.
23:16	RW	0x00	scale This 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 prescaled 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.

Bit	Attr	Reset Value	Description
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 PWM2_CTRL.rpt. 2'b01: Continuous mode. PWM produces the waveform continuously. 2'b10: Capture mode. PWM measures the cycles of high/low polarity of input waveform. 2'b11: Reserved
0	RW	0x0	pwm_en 1'b0: Disabled 1'b1: Enabled If the PWM is worked in the one-shot mode, this bit will be cleared at the end of operation.

PWM_PWM3_CNT

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	cnt The 32-bit indicates current value of PWM Channel 3 counter. The counter runs at the rate of PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM3_PERIOD_HPR

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	period_hpr If PWM is operated at the continuous mode or one-shot mode, this value defines the period of the output waveform. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the bit [31:1] is taken into account and bit [0] always considered as 0. If PWM is operated at the capture mode, this value indicates the effective high polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM3_DUTY_LPR

Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	duty_lpr If PWM is operated at the continuous mode or one-shot mode, this value defines the duty cycle of the output waveform. The PWM starts the output waveform with duty cycle. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the [31:1] is taken into account. If PWM is operated at the capture mode, this value indicates the effective low polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM3_CTRL

Address: Operational Base + offset (0x003C)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	rpt This field defines the repeated effective periods of output waveform in one-shot mode. The value N means N+1 repeated effective periods.
23:16	RW	0x00	scale This field defines the scale factor applied to prescaled clock. The value N means the clock is divided by 2*N. If N is 0, it means that the clock is divided by 512(2*256).
15	RO	0x0	reserved
14:12	RW	0x0	prescale This field defines the prescale factor applied to input clock. The value N means that the input clock is divided by 2^N.
11	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 prescaled 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 PWM3_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	0x00000	reserved
11	RO	0x0	CH3_Pol This bit is used in capture mode in order to identify the transition of the input waveform when interrupt is generated. When bit is 1, please refer to PWM3_PERIOD_HPR to know the effective high cycle of Channel 3 input waveform. Otherwise, please refer to PWM3_PERIOD_LPR to know the effective low cycle of Channel 3 input waveform. Write 1 to CH3_IntSts will clear this bit.
10	RO	0x0	CH2_Pol This bit is used in capture mode in order to identify the transition of the input waveform when interrupt is generated. When bit is 1, please refer to PWM2_PERIOD_HPR to know the effective high cycle of Channel 2 input waveform. Otherwise, please refer to PWM2_PERIOD_LPR to know the effective low cycle of Channel 2 input waveform. Write 1 to CH2_IntSts will clear this bit.
9	RO	0x0	CH1_Pol This bit is used in capture mode in order to identify the transition of the input waveform when interrupt is generated. When bit is 1, please refer to PWM1_PERIOD_HPR to know the effective high cycle of Channel 1 input waveform. Otherwise, please refer to PWM1_PERIOD_LPR to know the effective low cycle of Channel 1 input waveform. Write 1 to CH1_IntSts will clear this bit.
8	RO	0x0	CH0_Pol This bit is used in capture mode in order to identify the transition of the input waveform when interrupt is generated. When bit is 1, please refer to PWM0_PERIOD_HPR to know the effective high cycle of Channel 0 input waveform. Otherwise, please refer to PWM0_PERIOD_LPR to know the effective low cycle of Channel 0 input waveform. Write 1 to CH0_IntSts will clear this bit.
7	W1C	0x0	CH3_pwr_IntSts 1'b0: Channel 3 power key Interrupt not generated 1'b1: Channel 3 power key Interrupt generated
6:4	RO	0x0	reserved
3	W1C	0x0	CH3_IntSts 1'b0: Channel 3 Interrupt not generated 1'b1: Channel 3 Interrupt generated

Bit	Attr	Reset Value	Description
2	W1C	0x0	CH2_IntSts 1'b0: Channel 2 Interrupt not generated 1'b1: Channel 2 Interrupt generated
1	W1C	0x0	CH1_IntSts 1'b0: Channel 1 Interrupt not generated 1'b1: Channel 1 Interrupt generated
0	W1C	0x0	CH0_IntSts 1'b0: Channel 0 Interrupt not generated 1'b1: Channel 0 Interrupt generated

PWM_INT_EN

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7	RW	0x0	CH3_pwr_Int_en 1'b0: Channel 3 power key Interrupt disabled 1'b1: Channel 3 power key Interrupt enabled
6:4	RO	0x0	reserved
3	RW	0x0	CH3_Int_en 1'b0: Channel 3 Interrupt disabled 1'b1: Channel 3 Interrupt enabled
2	RW	0x0	CH2_Int_en 1'b0: Channel 2 Interrupt disabled 1'b1: Channel 2 Interrupt enabled
1	RW	0x0	CH1_Int_en 1'b0: Channel 1 Interrupt disabled 1'b1: Channel 1 Interrupt enabled
0	RW	0x0	CH0_Int_en 1'b0: Channel 0 Interrupt disabled 1'b1: Channel 0 Interrupt enabled

PWM_FIFO_CTRL

Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:14	RO	0x00000	reserved
13:12	RW	0x0	dma_ch_sel 2'b00: Select PWM0 2'b01: Select PWM1 2'b10: Select PWM2 2'b11: Select PWM3
11	RO	0x0	reserved

Bit	Attr	Reset Value	Description
10	RW	0x0	dma_ch_sel_en 1'b0: Disabled. Select the channel PWM3 to FIFO mode and DMA mode. 1'b1: Enabled. Use dma_ch_sel to select the channel to FIFO mode and DMA mode.
9	RW	0x0	timeout_en FIFO timeout enable.
8	RW	0x0	dma_mode_en 1'b1: Enabled 1'b0: Disabled
7	RO	0x0	reserved
6:4	RW	0x0	almost_full_watermark Almost full Watermark level.
3	RW	0x0	watermark_int_en Watermark full interrupt.
2	RW	0x0	overflow_int_en When high, an interrupt asserts when the FIFO overflow.
1	RW	0x0	full_int_en When high, an interrupt asserts when the FIFO is full.
0	RW	0x0	fifo_mode_sel When high, PWM FIFO mode is activated.

PWM_FIFO_INTSTS

Address: Operational Base + offset (0x0054)

Bit	Attr	Reset Value	Description
31:5	RO	0x0000000	reserved
4	RO	0x1	fifo_empty_status This bit indicates the FIFO is empty.
3	W1C	0x0	timieout_intsts Timeout interrupt.
2	W1C	0x0	fifo_watermark_full_intsts This bit indicates the FIFO is Watermark full.
1	W1C	0x0	fifo_overflow_intsts This bit indicates the FIFO is overflow.
0	W1C	0x0	fifo_full_intsts This bit indicates the FIFO is full.

PWM_FIFO_TOUTTHR

Address: Operational Base + offset (0x0058)

Bit	Attr	Reset Value	Description
31:20	RO	0x000	reserved
19:0	RW	0x00000	timeout_threshold FIFO timeout value (Unit: pwm clock).

PWM_VERSION_ID

Address: Operational Base + offset (0x005C)

Bit	Attr	Reset Value	Description
31:24	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	0x13	minor_version Minor version
15:0	RW	0x11B6	svn_version SVN version

PWM_FIFO

Address: Operational Base + offset (0x0060)

Bit	Attr	Reset Value	Description
31	RO	0x0	pol This bit indicates the polarity of the lower 31-bit counter. 1'b0: Low 1'b1: High
30:0	RO	0x00000000	cycle_cnt This 31-bit counter indicates the effective cycles of high/low waveform.

PWM_PWRMATCH_CTRL

Address: Operational Base + offset (0x0080)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15	RW	0x0	CH3_pwrkey_int_ctrl 1'b0: Assert interrupt after key capture with power key match 1'b1: Assert interrupt after key capture without power key match
14:12	RO	0x0	reserved
11	RW	0x0	CH3_pwrkey_capture_ctrl 1'b0: Capture the value after interrupt 1'b1: Capture the value directly
10:8	RO	0x0	reserved
7	RW	0x0	CH3_pwrkey_polarity 1'b0: PWM in polarity is positive 1'b1: PWM in polarity is negative
6:4	RO	0x0	reserved
3	RW	0x0	CH3_pwrkey_enable 1'b0: Disabled 1'b1: Enabled
2:0	RO	0x0	reserved

PWM_PWRMATCH_LPRE

Address: Operational Base + offset (0x0084)

Bit	Attr	Reset Value	Description
31:16	RW	0x238c	cnt_max The maximum counter value.
15:0	RW	0x22c4	cnt_min The minimum counter value.

PWM_PWRMATCH_HPRE

Address: Operational Base + offset (0x0088)

Bit	Attr	Reset Value	Description
31:16	RW	0x11f8	cnt_max The maximum counter value.
15:0	RW	0x1130	cnt_min The minimum counter value.

PWM_PWRMATCH_LD

Address: Operational Base + offset (0x008C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0294	cnt_max The maximum counter value.
15:0	RW	0x01cc	cnt_min The minimum counter value.

PWM_PWRMATCH_HD_ZERO

Address: Operational Base + offset (0x0090)

Bit	Attr	Reset Value	Description
31:16	RW	0x0294	cnt_max The maximum counter value.
15:0	RW	0x01cc	cnt_min The minimum counter value.

PWM_PWRMATCH_HD_ONE

Address: Operational Base + offset (0x0094)

Bit	Attr	Reset Value	Description
31:16	RW	0x06fe	cnt_max The maximum counter value.
15:0	RW	0x0636	cnt_min The minimum counter value.

PWM_PWRMATCH_VALUE0

Address: Operational Base + offset (0x0098)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value Power key match value 0.

PWM_PWRMATCH_VALUE1

Address: Operational Base + offset (0x009C)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value Power key match value 1.

PWM_PWRMATCH_VALUE2

Address: Operational Base + offset (0x00A0)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value Power key match value 2.

PWM_PWRMATCH_VALUE3

Address: Operational Base + offset (0x00A4)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value Power key match value 3.

PWM_PWRMATCH_VALUE4

Address: Operational Base + offset (0x00A8)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value Power key match value 4.

PWM_PWRMATCH_VALUE5

Address: Operational Base + offset (0x00AC)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value Power key match value 5.

PWM_PWRMATCH_VALUE6

Address: Operational Base + offset (0x00B0)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value Power key match value 6.

PWM_PWRMATCH_VALUE7

Address: Operational Base + offset (0x00B4)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value Power key match value 7.

PWM_PWRMATCH_VALUE8

Address: Operational Base + offset (0x00B8)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value Power key match value 8.

PWM_PWRMATCH_VALUE9

Address: Operational Base + offset (0x00BC)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value 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	0x0000	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
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

22.5 Interface Description

Table 22-1 PWM Interface Description

Module Pin	Direction	Pin Name	IOMUX Setting
PWM0CH0	I/O	GPIO0_B6/UART1_TX_M0/PWM0_M0	PMUGRF_GPIO0B_IOMUX_H[10:8]=3'b011
		GPIO2_B3/LCDC_D7/I2S2_MCLK_M1/CIF_D3_M1/UART5_CTSN_M1/PWM0_M1/SPI0_CS1N_M2/I2C5_SDA_M0	GRF_GPIO2B_IOMUX_L[14:12]=3'b101
PWM0CH1	I/O	GPIO0_B7/UART1_RX_M0/PWM1_M0	PMUGRF_GPIO0B_IOMUX_H[14:12]=3'b011
		GPIO2_B2/LCDC_D6/I2S2_LRCK_M1/UART5_RTSN_M1/PWM1_M1/SPI0_CLK_M2	GRF_GPIO2B_IOMUX_L[10:8]=3'b101
PWM0CH2	I/O	GPIO0_C0/SDMMC0_PWR/UART1_RTSN_M0/PWM2_M0	PMUGRF_GPIO0C_IOMUX_L[2:0]=3'b011
		GPIO2_B1/LCDC_D5/I2S2_SCLK_M1/UART5_RX_M1/PWM2_M1/SPI0_MISO_M2	GRF_GPIO2B_IOMUX_L[6:4]=3'b101
PWM0CH3	I/O	GPIO0_C1/PMU_DEBUG/UART1_CTSN_M0/PWM3_IR_M0	PMUGRF_GPIO0C_IOMUX_L[6:4]=3'b011
		GPIO2_B0/LCDC_D4/I2S2_SDI_M1/UART5_TX_M1/PWM3_IR_M1/SPI0_MOSI_M2	GRF_GPIO2B_IOMUX_L[2:0]=3'b101
PWM1CH0	I/O	GPIO0_C2/I2C2_SCL/PWM4_M0	PMUGRF_GPIO0C_IOMUX_L[10:8]=3'b011
		GPIO2_A7/LCDC_D3/I2S2_SDO_M1/UART4_RX_M1/PWM4_M1/SPI0_CS0N_M2	GRF_GPIO2A_IOMUX_H[14:12]=3'b101
PWM1CH1	I/O	GPIO0_C3/I2C2_SDA/PWM5_M0	PMUGRF_GPIO0C_IOMUX_L[14:12]=3'b011
		GPIO2_A6/LCDC_D2/RGMII_COL_M1/CIF_D2_M1/UART4_TX_M1/PWM5_M1	GRF_GPIO2A_IOMUX_H[10:8]=3'b101
PWM1CH2	I/O	GPIO0_B2/PMIC_SLEEP/TSADC_SHUT_M1/PWM6_M0	PMUGRF_GPIO0B_IOMUX_L[10:8]=3'b011
		GPIO2_D4/LCDC_DEN/PWM6_M1/SPI1_CS0N_M2/I2C3_SCL_M1	GRF_GPIO2D_IOMUX_H[2:0]=3'b101
PWM1CH3	I/O	GPIO0_B1/PMIC_INT/PWM7_IR_M0	PMUGRF_GPIO0B_IOMUX_L[6:4]=3'b011
		GPIO3_A0/CAN_RXD_M0/UART3_TX_M2/PWM7_IR_M1/SPI1_CS1N_M2/I2C4_SCL_M0	GRF_GPIO3A_IOMUX_L[2:0]=3'b101
PWM2CH0	I/O	GPIO3_A4/CIF_D0_M0/I2S0_SCLK_TX_M1/UART4_TX_M0/I2C3_SCL_M0/PWM8_M0	GRF_GPIO3A_IOMUX_H[2:0]=3'b110
		GPIO2_D7/LCDC_CLK/UART3_CTSN_M2/PWM8_M1/SPI1_MISO_M2	GRF_GPIO2D_IOMUX_H[14:12]=3'b101

Module Pin	Direction	Pin Name	IOMUX Setting
PWM2CH1	I/O	GPIO3_A5/CIF_D1_M0/RGMII_CRS_M0/I2S0_LRCK_TX_M1/UART4_RX_M0/I2C3_SDA_M0/PWM9_M0	GRF_GPIO3A_IOMUX_H [6:4]=3'b110
		GPIO2_D6/LCDC_VSYNC/UART3_RTSN_M2/PWM9_M1/SPI1_MOSI_M2	GRF_GPIO2D_IOMUX_H [10:8]=3'b101
PWM2CH2	I/O	GPIO3_A6/CIF_D2_M0/RGMII_COL_M0/I2S0_SDO0_M1/UART5_TX_M0/CAN_RXD_M1/PWM10_M0	GRF_GPIO3A_IOMUX_H [10:8]=3'b110
		GPIO2_D5/LCDC_HSYNC/PWM10_M1/SPI1_CLK_M2/I2C3_SDA_M1	GRF_GPIO2D_IOMUX_H [6:4]=3'b101
PWM2CH3	I/O	GPIO3_A7/CIF_D3_M0/RGMII_RXD2_M0/I2S0_SDI0_M1/UART5_RX_M0/CAN_TXD_M1/PWM11_IR_M0	GRF_GPIO3A_IOMUX_H [14:12]=3'b110
		GPIO3_A1/CAN_TXD_M0/UART3_RX_M2/PWM11_IR_M1/I2C4_SDA_M0	GRF_GPIO3A_IOMUX_L [6:4]=3'b101

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

22.6 Application Notes

22.6.1 PWM Capture Mode Standard Usage Flow

1. Set PWM_PWMx_CTRL.pwm_en to '0' to disable the PWM channel.
2. Choose the prescale factor and the scale factor for clk_pwm by programming PWM_PWMx_CTRL.prescale and PWM_PWMx_CTRL.scale, and select the clock needed by setting PWM_PWMx_CTRL.clk_sel and PWM_PWMx_CTRL.clk_src_sel.
3. Configure the channel to work in the capture mode.
4. Enable the PWM_INT_EN.chx_int_en to enable the interrupt generation.
5. Set PWM_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.

22.6.2 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.

22.6.3 PWM Power key Capture Mode Standard Usage Flow

1. Set PWM_PWM3_CTRL.pwm_en to '0' to disable the PWM channel.
2. Choose the prescale factor and the scale factor for clk_pwm by programming PWM_PWM3_CTRL.prescale and PWM_PWM3_CTRL.scale, and select the clock needed by setting PWM_PWM3_CTRL.clk_sel. The clock should be 1 MHz after division.
3. Configure the channel to work in the capture mode.
4. Enable the PWM_INT_EN.CH3_int_pwr to enable the interrupt generation.
5. Set the PWM_PWRMATCH_VALUE0~9 registers for the 10 power key match value.
6. Set max_cnt and min_cnt of follow register: PWM_PWRMATCH_LPRE, PWM_PWRMATCH_HPRE, PWM_PWRMATCH_LD, PWM_PWRMATCH_HD_ZERO, PWM_PWRMATCH_HD_ONE. It doesn't need to set these registers when the default value can meet the requirement.
7.Set PWM_PWRMATCH_CTRL.CH3_pwrkey_polarity for the polarity of power key signal, the default value is 0. Enable the PWM_PWRMATCH_CTRL.CH3_pwrkey_enable.
8. Set PWM_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. Poll INTSTS. CH3_pwr_IntSts ==1, and refer to PWM_PWM3_PWRCAPTURE_VALUE to know the power key capture value.
11. Write '0' to PWM_PWM3_CTRL.pwm_en to disable the channel.

22.6.4 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.

22.6.5 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.

22.6.6 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 23 I2C Interface

23.1 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 6 independent I2C: I2C0/1/2/3/4/5, with 6 pairs of IOs
- Item Compatible with I2C-bus
- AMBA APB slave interface
- Supports master mode of I2C bus
- Software programmable clock frequency and transfer rate up to 400Kbit/sec
- Supports 7 bits and 10 bits addressing modes
- Interrupt or polling driven multiple bytes data transfer
- Clock stretching and wait state generation
- Filter out glitch on SCL and SDA

23.2 Block Diagram

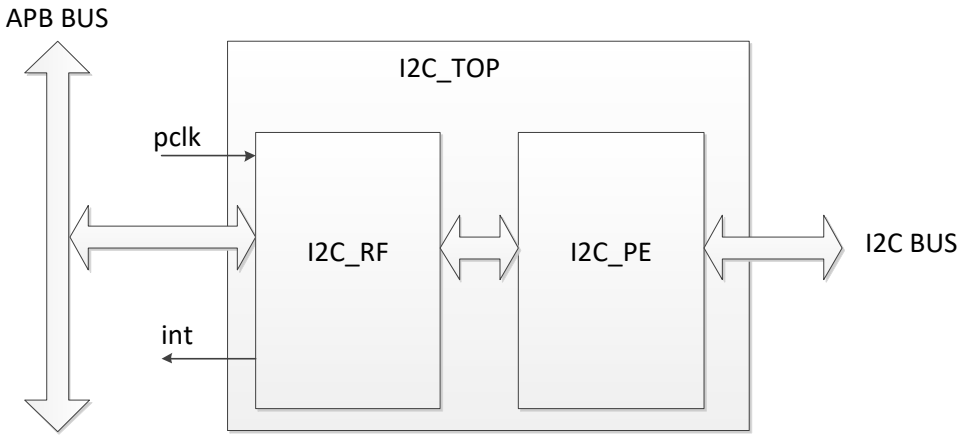


Fig. 23-1 I2C Architecture

23.2.1 I2C_RF

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

23.2.2 I2C_PE

I2C_PE module implements the I2C master operation for transmit data to and receive data from other I2C devices. The I2C master controller operates synchronously with the pclk.

23.2.3 I2C_TOP

I2C_TOP module is the top module of the I2C controller.

23.3 Function Description

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

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

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

23.3.1 Initialization

The I2C controller is based on AMBA APB bus architecture and usually is part of a SOC. So before I2C operates, some system setting and configuration must be conformed, which includes:

- I2C interrupt connection type: CPU interrupt scheme should be considered. If the I2C interrupt is connected to extra Interrupt Controller module, we need decide the INTC vector.
- I2C Clock Rate: The I2C controller uses the APB clock as the working clock so the APB clock will determine the I2C bus clock. The correct register setting is subject to the system requirement.

23.3.2 Master Mode Programming

- SCL Clock

When the I2C controller is programmed in Master mode, the SCL frequency is determined by I2C_CLKDIV register. The SCL frequency is calculated by the following formula:

SCL Divisor = 8*(CLKDIVL + 1 + CLKDIVH + 1); clk_i2c = 100MHz~200MHz

SCL = clk_i2c/ SCLK Divisor

- Data Receiver Register Access

When the I2C controller received MRXCNT bytes data， CPU can get the data through register RXDATA0 ~ RXDATA7. The controller can receive up to 32 bytes' data in one transaction.

When MRXCNT register is written, the I2C controller will start to drive SCL to receive data.

- Transmit Transmitter Register

Data to transmit are written to TXDATA0~7 by CPU. The controller can transmit up to 32 bytes' data in one transaction. The lower byte will be transmitted first.

When MTXCNT register is written, the I2C controller will start to transmit data.

- Start Command

Write 1 to I2C_CON[3], the controller will send I2C start command.

- Stop Command

Write 1 to I2C_CON[4], the controller will send I2C stop command

- I2C Operation mode

There are four i2c operation modes.

- When I2C_CON[2:1] is 2'b00, the controller transmit all valid data in TXDATA0~TXDATA7 byte by byte. The controller will transmit lower byte first.
- When I2C_CON[2:1] is 2'b01,the controller will transmit device address in MRXADDR first (Write/Read bit = 0) and then transmit device register address in MRXRADDR. After that, the controller will assert restart signal and resend MRXADDR (Write/Read bit = 1). At last, the controller enter receive mode.
- When I2C_CON[2:1] is 2'b10, the controller is in receive mode, it will trigger clock to read MRXCNT byte data.
- When I2C_CON[2:1] is 2'b11, the controller will transmit device address in MRXADDR first (Write/Read bit = 1) and then transmit device register address in MRXRADDR . After that, the controller will assert restart signal and resend MRXADDR (Write/Read bit = 1). At last, the controller enter receive mode.

- Read/Write Command

- When I2C_OPMODE(I2C_CON[2:1]) is 2'b01 or 2'b11, the Read/Write command bit is decided by controller itself.
- In RX only mode (I2C_CON[2:1] is 2'b10), the Read/Write command bit is decided by MRXADDR[0].
- In TX only mode (I2C_CON[[2:1] is 2'b00), the Read/Write command bit is decided by TXDATA[0].

- Master Interrupt Condition

There are 7 interrupt bits in I2C_ISR register related to master mode.

- Byte transmitted finish interrupt (Bit 0): The bit is asserted when Master completed

transmitting a byte.

- Byte received finish interrupt (Bit 1): The bit is asserted when Master completed receiving a byte.
- MTXCNT bytes data transmitted finish interrupt (Bit 2): The bit is asserted when Master completed transmitting MTXCNT bytes.
- MRXCNT bytes data received finish interrupt (Bit 3): The bit is asserted when Master completed receiving MRXCNT bytes.
- Start interrupt (Bit 4): The bit is asserted when Master finished asserting start command to I2C bus.
- Stop interrupt (Bit 5): The bit is asserted when Master finished asserting stop command to I2C bus.
- NAK received interrupt (Bit 6): The bit is asserted when Master received a NAK handshake.
    - Last byte acknowledge control
- If I2C_CON[5] is 1, the I2C controller will transmit NAK handshake to slave when the last byte received in RX only mode.
- If I2C_CON[5] is 0, the I2C controller will transmit ACK handshake to slave when the last byte received in RX only mode.
    - How to handle NAK handshake received
- If I2C_CON[6] is 1, the I2C controller will stop all transactions when NAK handshake received. And the software should take responsibility to handle the problem.
- If I2C_CON[6] is 0, the I2C controller will ignore all NAK handshake received.
    - I2C controller data transfer waveform
- Bit transferring
    - ◆ Data Validity

The SDA line must be stable during the high period of SCL, and the data on SDA line can only be changed when SCL is in low state.

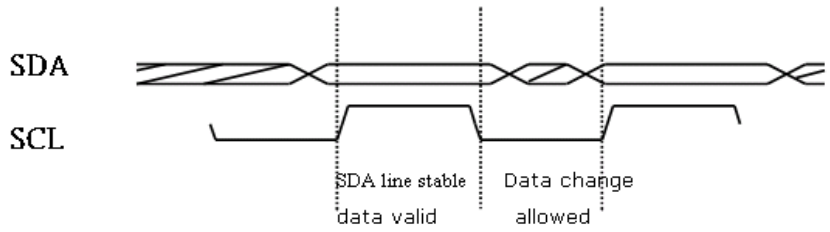


Fig. 23-2 I2C DATA Validity

- ◆ 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.

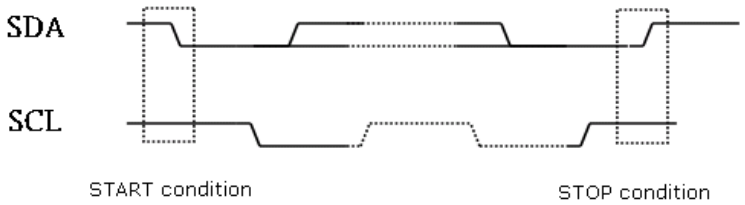


Fig. 23-3 I2C Start and Stop Conditions

- ◆ 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".

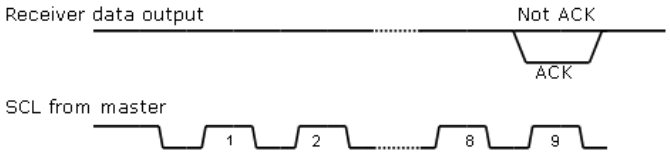


Fig. 23-4 I2C Acknowledge

- Byte transfer

The master own I2C bus might initiate multiple 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.

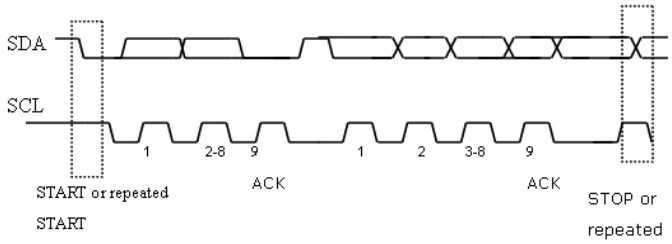


Fig. 23-5 I2C Byte Transfer

23.4 Register Description

23.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
RKI2C_CON	0x0000	W	0x00030000	Control Register
RKI2C_CLKDIV	0x0004	W	0x00000001	Clock Divider Register
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 TX Count Register
RKI2C_MRXCNT	0x0014	W	0x00000000	Master RX Count Register
RKI2C_IEN	0x0018	W	0x00000000	Interrupt Enable Register
RKI2C_IPD	0x001C	W	0x00000000	Interrupt Pending Register
RKI2C_FCNT	0x0020	W	0x00000000	Finished Count Register
RKI2C_SCL_OE_DB	0x0024	W	0x00000020	Slave Hold Debounce 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

Name	Offset	Size	Reset Value	Description
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, **DW**-Double WORD (64 bits) access*

23.4.2 Detail Register Description

RKI2C_CON

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:16	RW	0x0003	version RKI2C version information.
15:14	RW	0x0	stop_setup Stop setup config. TSU; sto = (stop_setup + 1) * T(SCL_HIGH) + Tclk_i2c
13:12	RW	0x0	start_setup Start setup config. TSU; sta = (start_setup + 1) * T(SCL_HIGH) + Tclk_i2c THD; sta = (start_setup + 2) * T(SCL_HIGH) - Tclk_i2c
11:8	RW	0x0	data_upd_st SDA update point config. Used to config sda change state when scl is low, and used to adjust setup/hold time. 4'bn: Thold = (n + 1) * Tclk_i2c Note: 0 ≤ n ≤ 5
7	RO	0x0	reserved
6	RW	0x0	act2nak Operation when NAK handshake is received. 1'b0: Ignored 1'b1: Stop transaction
5	RW	0x0	ack Last byte acknowledge control in master receive mode. 1'b0: ACK 1'b1: NAK
4	RW	0x0	stop Stop enable. When this bit is written to 1, I2C will generate stop signal.
3	RW	0x0	start Start enable. When this bit is written to 1, I2C will generate start signal.
2:1	RW	0x0	i2c_mode I2C mode select. 2'b00: Transmit only 2'b01: Transmit address (device + register address) --> restart --> transmit address --> receive only 2'b10: Receive only 2'b11: Transmit address (device + register address, write/read bit is 1) --> restart --> transmit address (device address) --> receive data
0	RW	0x0	i2c_en I2C module enable. 1'b0: Not enable 1'b1: Enable

RKI2C_CLKDIV

Address: Operational Base + offset (0x0004)

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

RKI2C_MRXADDR

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:27	RO	0x00	reserved
26	RW	0x0	addhvld Address high byte valid. 1'b0: Invalid 1'b1: Valid
25	RW	0x0	addmvld Address middle byte valid. 1'b0: Invalid 1'b1: Valid
24	RW	0x0	addlvld Address low byte valid. 1'b0: Invalid 1'b1: Valid
23:0	RW	0x000000	saddr Master address register. The lowest bit indicate write or read.

RKI2C_MRXRADDR

Address: Operational Base + offset (0x000C)

Bit	Attr	Reset Value	Description
31:27	RO	0x00	reserved
26	RW	0x0	sraddhvld Address high byte valid. 1'b0: Invalid 1'b1: Valid
25	RW	0x0	sraddmvld Address middle byte valid. 1'b0: Invalid 1'b1: Valid
24	RW	0x0	sraddlvld Address low byte valid. 1'b0: Invalid 1'b1: Valid
23:0	RW	0x000000	sraddr Slave register address accessed.

RKI2C_MTXCNT

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:6	RO	0x0000000	reserved
5:0	RW	0x00	mtxcnt Master transmit count. Specify the total bytes to be transmitted (0~32).

RKI2C_MRXCNT

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:6	RO	0x0000000	reserved
5:0	RW	0x00	mrxcnt Master RX count. Specify the total bytes to be received(0~32).

RKI2C_IEN

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7	RW	0x0	slavehdsclen Slave hold scl interrupt enable. 1'b0: Disable 1'b1: Enable
6	RW	0x0	nakrcvien NAK handshake received interrupt enable. 1'b0: Disable 1'b1: Enable
5	RW	0x0	stopien Stop operation finished interrupt enable. 1'b0: Disable 1'b1: Enable
4	RW	0x0	startien Start operation finished interrupt enable. 1'b0: Disable 1'b1: Enable
3	RW	0x0	mbrfien MRXCNT data received finished interrupt enable. 1'b0: Disable 1'b1: Enable
2	RW	0x0	mbtfien MTXCNT data transfer finished interrupt enable. 1'b0: Disable 1'b1: Enable
1	RW	0x0	brfien Byte RX finished interrupt enable. 1'b0: Disable 1'b1: Enable
0	RW	0x0	btfien Byte TX finished interrupt enable. 1'b0: Disable 1'b1: Enable

RKI2C_IPD

Address: Operational Base + offset (0x001C)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7	RW	0x0	slavehdsclipd Slave hold scl interrupt pending bit. 1'b0: No interrupt available 1'b1: Slave hold scl interrupt appear, write 1 to clear

Bit	Attr	Reset Value	Description
6	RW	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	RW	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	RW	0x0	startipd interrupt pending bit. 1'b0: No interrupt available 1'b1: Start operation finished interrupt appear, write 1 to clear
3	RW	0x0	mbrfipd MRXCNT data received finished interrupt pending bit. 1'b0: No interrupt available 1'b1: MRXCNT data received finished interrupt appear, write 1 to clear
2	RW	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	RW	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	RW	0x0	btfipd Byte TX finished interrupt pending bit. 1'b0: No interrupt available 1'b1: Byte TX finished interrupt appear, write 1 to clear

RKI2C_FCNT

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:6	RO	0x0000000	reserved
5:0	RW	0x00	fcnt The count of data which has been transmitted or received.

RKI2C_SCL_OE_DB

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x20	scl_oe_db Slave hold scl debounce. Cycles for debounce (unit: Tclk_i2c).

RKI2C_TXDATA0

Address: Operational Base + offset (0x0100)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata0 Data0 to be transmitted.

RKI2C_TXDATA1

Address: Operational Base + offset (0x0104)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata1 Data1 to be transmitted.

RKI2C_TXDATA2

Address: Operational Base + offset (0x0108)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata2 Data2 to be transmitted.

RKI2C_TXDATA3

Address: Operational Base + offset (0x010C)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata3 Data3 to be transmitted.

RKI2C_TXDATA4

Address: Operational Base + offset (0x0110)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata4 Data4 to be transmitted.

RKI2C_TXDATA5

Address: Operational Base + offset (0x0114)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata5 Data5 to be transmitted.

RKI2C_TXDATA6

Address: Operational Base + offset (0x0118)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata6 Data6 to be transmitted.

RKI2C_TXDATA7

Address: Operational Base + offset (0x011C)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata7 Data7 to be transmitted.

RKI2C_RXDATA0

Address: Operational Base + offset (0x0200)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxdata0 Data0 received.

RKI2C_RXDATA1

Address: Operational Base + offset (0x0204)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxdata1 Data1 received.

RKI2C_RXDATA2

Address: Operational Base + offset (0x0208)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxdata2 Data2 received.

RKI2C_RXDATA3

Address: Operational Base + offset (0x020C)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxdata3 Data3 received.

RKI2C_RXDATA4

Address: Operational Base + offset (0x0210)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxdata4 Data4 received.

RKI2C_RXDATA5

Address: Operational Base + offset (0x0214)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxdata5 Data5 received.

RKI2C_RXDATA6

Address: Operational Base + offset (0x0218)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxdata6 Data6 received.

RKI2C_RXDATA7

Address: Operational Base + offset (0x021C)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxdata7 Data7 received.

RKI2C_ST

Address: Operational Base + offset (0x0220)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved
1	RW	0x0	scl_st Scl status. 1'b0: Scl status low 1'b0: Scl status high
0	RW	0x0	sda_st Sda status. 1'b0: Sda status low 1'b0: Sda status high

RKI2C_DBGCTRL

Address: Operational Base + offset (0x0224)

Bit	Attr	Reset Value	Description
31:15	RO	0x00000	reserved
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)

Bit	Attr	Reset Value	Description
13	RW	0x0	nak_release_scl 1'b0: Hold scl as low when recieved nack 1'b1: Release scl as high when recieved nack
12	RW	0x0	flt_en SCL edge 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.

23.5 Interface Description

Table 23-1 I2C Interface Description

Module Pin	Direction	Pin Name	IOMUX Setting
IOMUX0 I2C0 Interface			
I2C_SDA0	I/O	I2C0_SDA/GPIO0_B5_u	PMUGRF_GPIO0B_IOMUX_SEL_H[3:0]=4'h1
I2C_SCL0	I/O	I2C0_SCL/GPIO0_B4_u	PMUGRF_GPIO0B_IOMUX_SEL_H[7:4]=4'h1
IOMUX0 I2C1 Interface			
I2C_SDA1	I/O	I2C1_SDA/UART4_RTSN_M2/GPIO1_D2_u	PMUGRF_GPIO1D_IOMUX_SEL_L[11:8]=4'h1
I2C_SCL1	I/O	I2C1_SCL/UART4_CTSN_M2/GPIO1_D3_u	PMUGRF_GPIO1D_IOMUX_SEL_L[15:12]=4'h1
IOMUX0 I2C2 Interface			
I2C_SDA2	I/O	I2C2_SDA/PWM5_M0/GPIO0_C3_d	PMUGRF_GPIO0C_IOMUX_SEL_L[15:12]=4'h1
I2C_SCL2	I/O	I2C2_SCL/PWM4_M0/GPIO0_C2_d	PMUGRF_GPIO0C_IOMUX_SEL_L[11:8]=4'h1
IOMUX0 I2C3 Interface			
I2C_SDA3	I/O	CIF_D1_M0/RGMII_CRS_M0/I2S0_LRCK_TX_M1/UART4_RX_M0/I2C3_SDA_M0/PWM9_M0/GPIO3_A5_d	GRF_GPIO3A_IOMUX_SEL_H[7:4]=4'h5
I2C_SCL3	I/O	CIF_D0_M0/I2S0_SCLK_TX_M1/UART4_TX_M0/I2C3_SCL_M0/PWM8_M0/GPIO3_A4_d	GRF_GPIO3A_IOMUX_SEL_H[3:0]=4'h5
IOMUX0 I2C4 Interface			
I2C_SDA4	I/O	CAN_TXD_M0/UART3_RX_M2/PWM11_IR_M1/I2C4_SDA_M0/GPIO3_A1_u	GRF_GPIO3A_IOMUX_SEL_L[7:4]=4'h7
I2C_SCL4	I/O	CAN_RXD_M0/UART3_TX_M2/PWM7_IR_M1/SPI1_CS1n_M2/I2C4_SCL_M0/GPIO3_A0_u	GRF_GPIO3A_IOMUX_SEL_L[3:0]=4'h7
IOMUX0 I2C5 Interface			
I2C_SDA5	I/O	LCDC_D7/I2S2_MCLK_M1/CIF_D3_M1/UART5_CTSN_M1/SPI0_CS1n_M2/PWM0_M1/I2C5_SDA_M0/ GPIO2_B3_d	GRF_GPIO2B_IOMUX_SEL_L[15:12]=4'h7
I2C_SCL5	I/O	LCDC_D1/RGMII_CRS_M1/CIF_D1_M1/UART4_CTSN_M1/I2C5_SCL_M0/GPIO2_A5_d	GRF_GPIO2A_IOMUX_SEL_H[7:4]=4'h7
IOMUX1 I2C3 Interface			
I2C_SDA3	I/O	LCDC_HSYNC/PWM10_M1/ SPI1_CLK_M2/I2C3_SDA_M1/GPIO2_D5_d	GRF_GPIO2D_IOMUX_SEL_H[3:0]=4'h7
I2C_SCL3	I/O	LCDC_DEN/PWM6_M1/SPI1_CS0n_M2/I2C3_SCL_M1/GPIO2_D4_d	GRF_GPIO2D_IOMUX_SEL_H[7:4]=4'h7
IOMUX1 I2C4 Interface			
I2C_SDA4	I/O	I2S0_SDO3_SDI1_M0/PDM_SDI1_M0/AUDPWM_R_M0/I2C4_SDA_M1/AUDDSM_RP/GPIO4_A1_d	GRF_GPIO4A_IOMUX_SEL_L[7:4]=4'h4
I2C_SCL4	I/O	I2S0_SDO2_SDI2_M0/PDM_SDI2_M0/AUDPWM_L_M0/I2C4_SCL_M1/AUDDSM_RN/GPIO4_A0_d	GRF_GPIO4A_IOMUX_SEL_L[3:0]=4'h4
IOMUX1 I2C5 Interface			

I2C_SDA5	I/O	CIF_D5_M0/RGMII_TXD2_M0/I2S0_SCLK_RX_M1/UART5_CTSN_M0/I2C5_SDA_M1/GPIO3_B1_d	GRF_GPIO3B_IOMUX_SEL_L[7:4]=4'h5
I2C_SCL5	I/O	CIF_D4_M0/RGMII_RXD3_M0/I2S0_MCLK_M1/UART5_RTSN_M0/I2C5_SCL_M1/GPIO3_B0_d	GRF_GPIO3B_IOMUX_SEL_L[3:0]=4'h5
IOMUX2 I2C3 Interface			
I2C_SDA3	I/O	SPI0_MISO_M1/I2S1_LRCK_M1/I2C3_SDA_M2/GPIO1_D7_d	GRF_GPIO1D_IOMUX_SEL_H[15:12]=4'h3
I2C_SCL3	I/O	SPI0_MOSI_M1/I2S1_SCLK_M1/I2C3_SCL_M2/GPIO1_D6_d	GRF_GPIO1D_IOMUX_SEL_H[11:8]=4'h3
IOMUX2 I2C5 Interface			
I2C_SDA5	I/O	SDMMC1_PWR/I2C5_SDA_M2/UART1_RX_M1/GPIO1_D1_d	GRF_GPIO1D_IOMUX_SEL_L[7:4]=4'h4
I2C_SCL5	I/O	I2S2_MCLK_M0/SDMMC1_DET/SPI1_CS1n_M1/I2C5_SCL_M2/UART1_TX_M1/GPIO1_D0_d	GRF_GPIO1D_IOMUX_SEL_L[3:0]=4'h4

23.6 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)

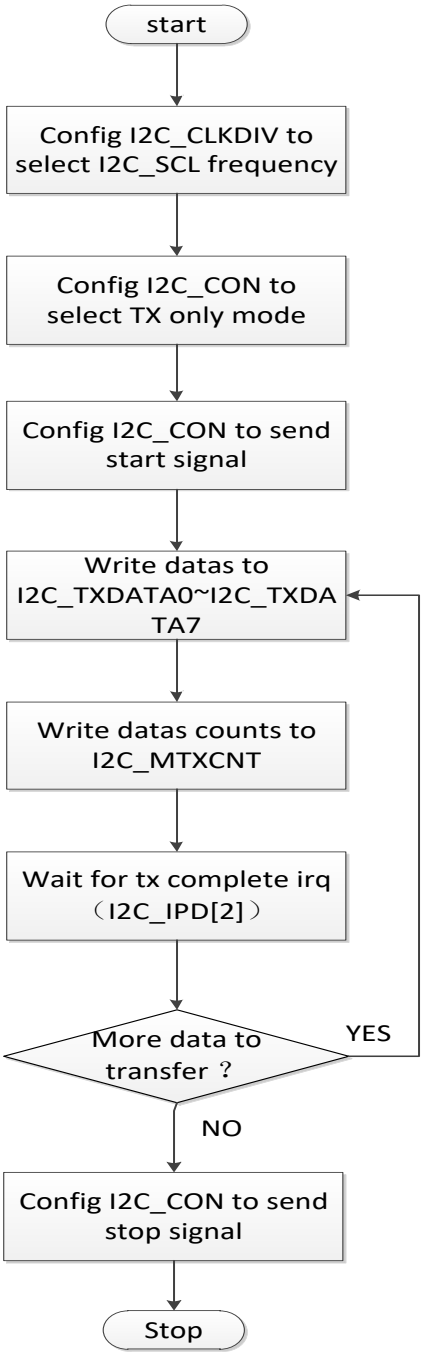


Fig. 23-6 I2C Flow Chat for Transmit Only Mode

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

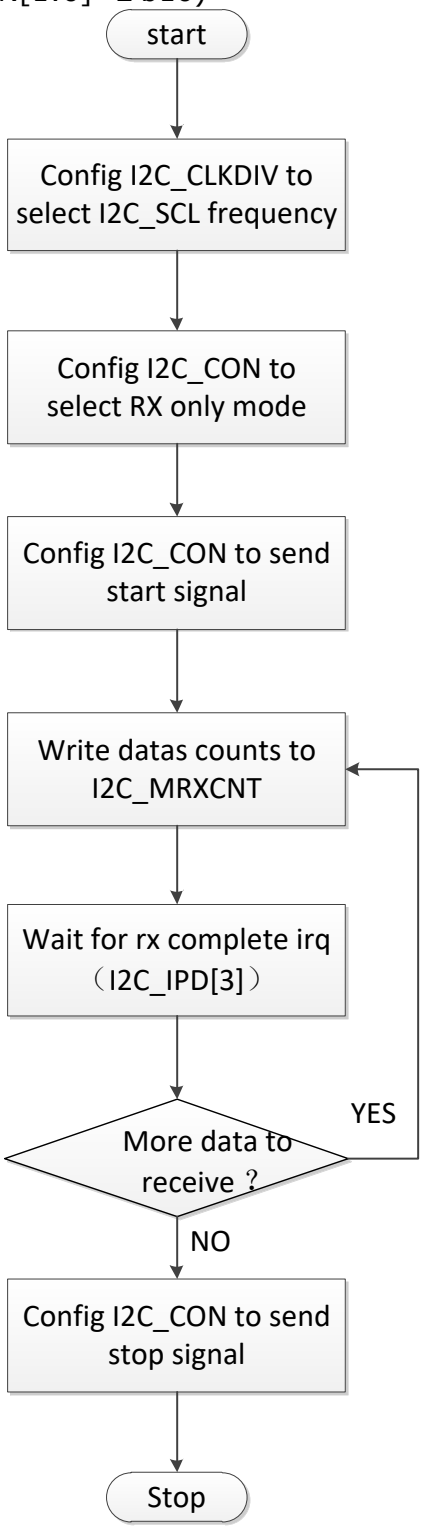


Fig. 23-7 I2C Flow Chat for Receive Only Mode

- Mix mode (I2C_CON[1:0]=2'b01 or I2C_CON[1:0]=2'b11)

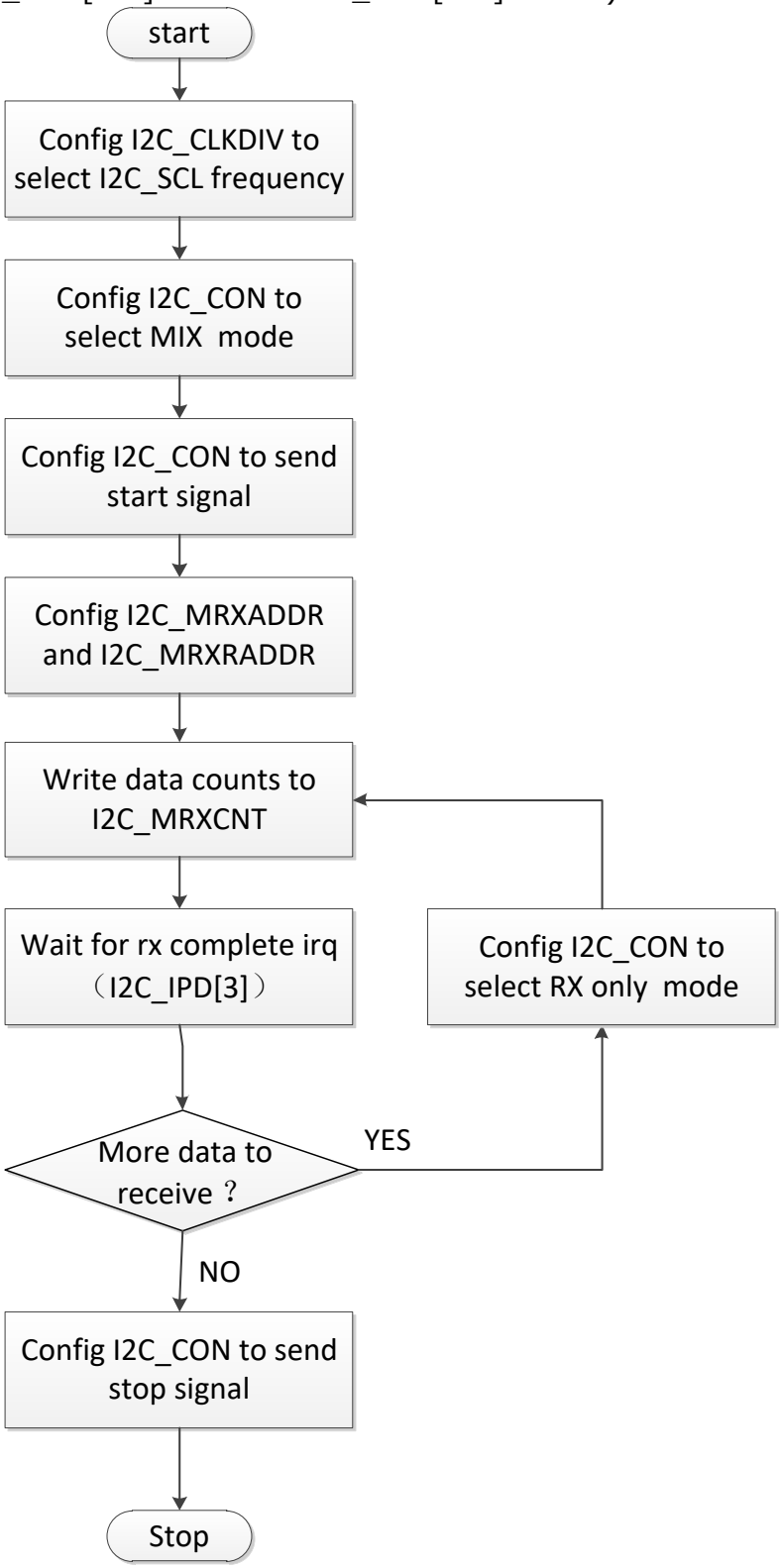


Fig. 23-8 I2C Flow Chat for Mix Mode

Chapter 24 UART

24.1 Overview

The Universal Asynchronous Receiver/Transmitter (UART) is used for serial communication with a peripheral, modem (data carrier equipment, DCE) or data set. Data is written from a master (CPU) over the APB bus to the UART and it is converted to serial form and transmitted to the destination device. Serial data is also received by the UART and stored for the master (CPU) to read back.
UART Controller supports the following features:

- Support 6 independent UART controller: UART0-UART5
- All contain two 64Bytes FIFOs for data receive and transmit
- All support auto flow-control except
- 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

24.2 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

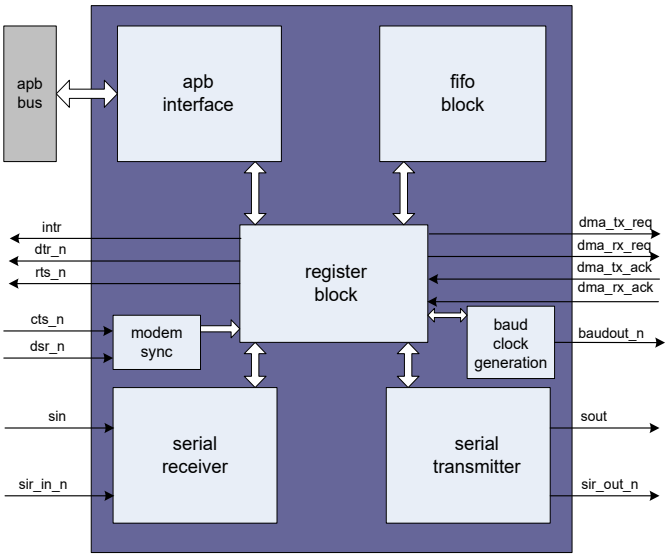


Fig. 23-9 UART Architecture

APB INTERFACE
The host processor accesses data, control, and status information on the UART through the APB interface. The UART supports APB data bus widths of 8, 16, and 32 bits.
Register block
Be responsible for the main UART functionality including control, status and interrupt generation.
Modem Synchronization block
Synchronizes the modem input signal.
FIFO block
Be responsible for FIFO control and storage (when using internal RAM) or signaling to control external RAM (when used).
Baud Clock Generator

Generates the transmitter and receiver baud clock along with the output reference clock signal (baudout_n).

Serial Transmitter

Converts the parallel data, written to the UART, into serial form and adds all additional bits, as specified by the control register, for transmission. 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.

24.3 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.

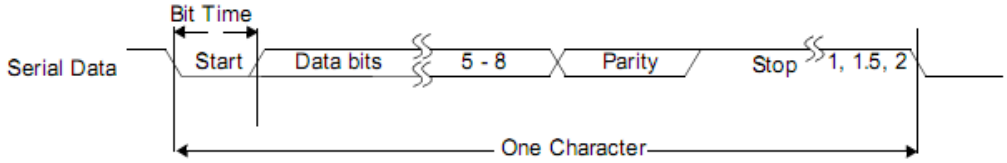


Fig. 23-10 UART Serial protocol

IrDA 1.0 SIR Protocol

The Infrared Data Association (IrDA) 1.0 Serial Infrared (SIR) mode supports bi-directional datacommunications with remote devices using infrared radiation as the transmission medium. IrDA 1.0 SIR mode specifies a maximum baud rate of 115.2 Kbaud.

Transmitting a single infrared pulse signals a logic zero, while a logic one is represented by not sending a pulse. The width of each pulse is 3/16ths of a normal serial bit time. Data transfers can only occur in half-duplex fashion when IrDA SIR mode is enabled.

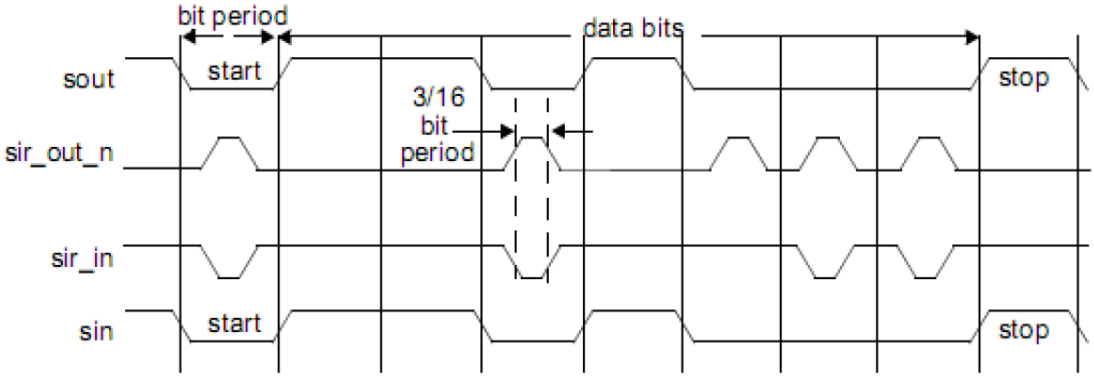


Fig. 23-11 IrDA 1.0

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.

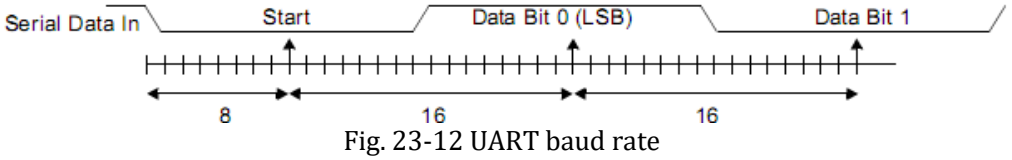


Fig. 23-12 UART baud rate

FIFO Support

1. NONE FIFO MODE

If FIFO support is not selected, then no FIFOs are implemented and only a single receive data byte and transmit data byte can be stored at a time in the RBR and THR.

2. FIFO MODE

The FIFO depth of UART0/UART1/UART2is 64bytes. The FIFO mode of all the UART is enabled by register FCR[0].

Interrupts

The following interrupt types can be enabled with the IER register.

- Receiver Error
- Receiver Data Available
- Character Timeout (in FIFO mode only)
- Transmitter Holding Register Empty at/below threshold (in Programmable THRE Interrupt mode)
- Modem Status

DMA Support

The UART supports DMA signaling with the use of two output signals (dma_tx_req_n and dma_rx_req_n) to indicate when data is ready to be read or when the transmit FIFO is empty.

The dma_tx_req_n signal is asserted under the following conditions:

- When the Transmitter Holding Register is empty in non-FIFO mode.
- When the transmitter FIFO is empty in FIFO mode with Programmable THRE interrupt mode disabled.
- When the transmitter FIFO is at, or below the programmed threshold with Programmable THRE interrupt mode enabled.

The dma_rx_req_n signal is asserted under the following conditions:

- When there is a single character available in the Receive Buffer Register in non-FIFO mode.
- When the Receiver FIFO is at or above the programmed trigger level in FIFO mode.

Auto Flow Control

The UART can be configured to have a 16750-compatible Auto RTS and Auto CTS serial data flow control mode available. If FIFOs are not implemented, then this mode cannot be selected. When Auto Flow Control mode has been selected, it can be enabled with the Modem Control Register (MCR[5]). Following figure shows a block diagram of the Auto Flow Control functionality.

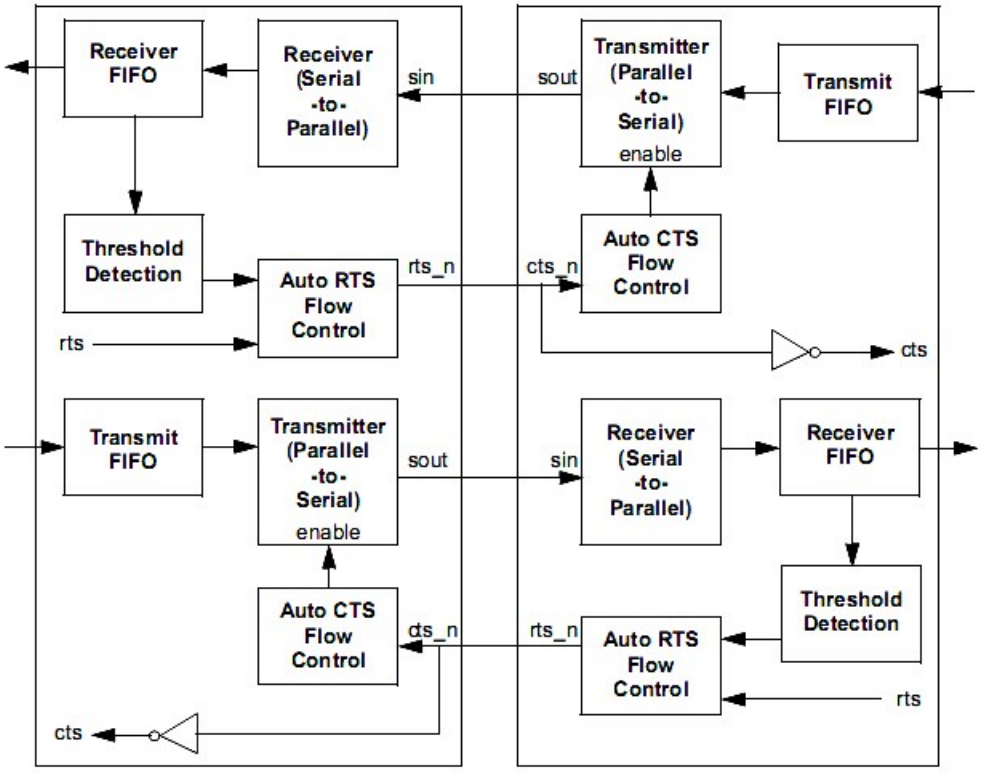


Fig. 23-13 UART Auto flow control block diagram

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)

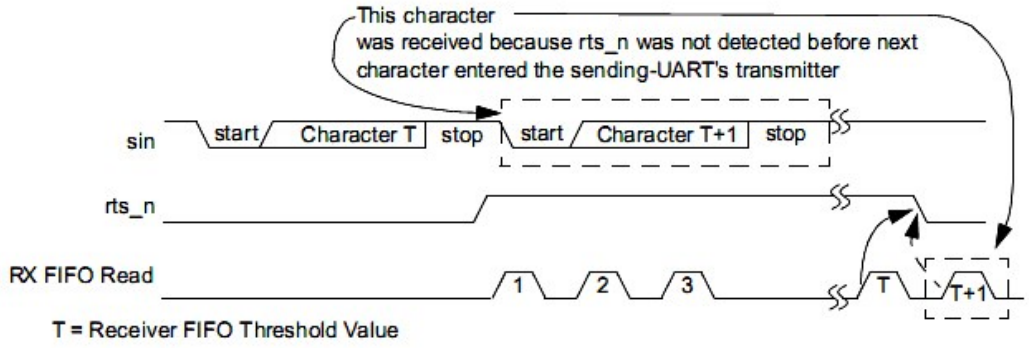


Fig. 23-14 UART AUTO RTS TIMING

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)

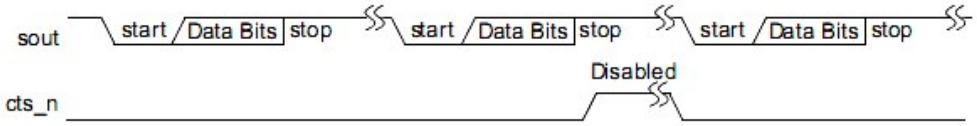


Fig. 23-15 UART AUTO CTS TIMING

24.4 Register Description

This section describes the control/status registers of the design. There are 3 UARTs in RK3228, and each one has its own base address.

24.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
UART_RBR	0x0000	W	0x00000000	Receive Buffer Register
UART_DLL	0x0000	W	0x00000000	Divisor Latch Low
UART_THR	0x0000	W	0x00000000	Transmit Buffer Register
UART_DLH	0x0004	W	0x00000000	Divisor Latch High
UART_IER	0x0004	W	0x00000000	Interrupt Enable Register
UART_FCR	0x0008	W	0x00000000	FIFO Enable
UART_IIR	0x0008	W	0x00000001	Interrupt Identity Register
UART_LCR	0x000c	W	0x00000000	Line Control Register
UART_MCR	0x0010	W	0x00000000	Modem Control Register
UART_LSR	0x0014	W	0x00000060	Line Status Register
UART_MSR	0x0018	W	0x00000000	Modem Status Register
UART_SCR	0x001c	W	0x00000000	Scratchpad Register
UART_SRBR	0x0030	W	0x00000000	Shadow Receive Buffer Register

Name	Offset	Size	Reset Value	Description
UART_STHR	0x0030	W	0x00000000	Shadow Transmit Holding Register
UART_FAR	0x0070	W	0x00000000	FIFO Access Register
UART_TFR	0x0074	W	0x00000000	Transmit FIFO Read
UART_RFW	0x0078	W	0x00000000	Receive FIFO write
UART_USR	0x007c	W	0x00000006	UART Status Register
UART_TFL	0x0080	W	0x00000000	Transmit FIFO level
UART_RFL	0x0084	W	0x00000000	Receive FIFO level
UART_SRR	0x0088	W	0x00000000	Software Reset Register
UART_SRTS	0x008c	W	0x00000000	Shadow Request to Send
UART_SBCR	0x0090	W	0x00000000	Shadow Break Control Register
UART_SDMAM	0x0094	W	0x00000000	Shadow DMA Mode
UART_SFE	0x0098	W	0x00000000	Shadow FIFO enable
UART_SRT	0x009c	W	0x00000000	Shadow RCVR Trigger
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	0x00000000	UART Component Version
UART_CTR	0x00fc	W	0x44570110	Component Type Register

24.4.2 Detail Register Description

UART_RBR

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RO	0x00	rbr Data byte received on the serial input port (sin) in UART mode, or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line Status Register (LCR) is set. If in non-FIFO mode (FIFO_MODE == NONE) or FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it is overwritten, resulting in an over-run error. If in FIFO mode (FIFO_MODE != NONE) and FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO is preserved, but any incoming data are lost and an over-run error occurs.

UART_DLL

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x00	dll Lower 8 bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. This register may only be accessed when the DLAB bit (LCR[7]) is set and the UART is not busy (USR[0] is zero). The output baud rate is equal to the serial clock (pclk if one clock design, sclk if two clock design (CLOCK_MODE == Enabled)) frequency divided by sixteen times the value of the baud rate divisor, as follows: baud rate = (serial clock freq) / (16 * divisor). Note that with the Divisor Latch Registers (DLL and DLH) set to zero, the baud clock is disabled and no serial communications occur. Also, once the DLL is set, at least 8 clock cycles of the slowest DW_apb_uart clock should be allowed to pass before transmitting or receiving data.

UART_THR

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	WO	0x00	thr Data to be transmitted on the serial output port (sout) in UART mode or the serial infrared output (sir_out_n) in infrared mode. Data should only be written to the THR when the THR Empty (THRE) bit (LSR[5]) is set. If in non-FIFO mode or FIFOs are disabled (FCR[0] = 0) and THRE is set, writing a single character to the THR clears the THRE. Any additional writes to the THR before the THRE is set again causes the THR data to be overwritten. If in FIFO mode and FIFOs are enabled (FCR[0] = 1) and THRE is set, 64 characters of data may be written to the THR before the FIFO is full.. Any attempt to write data when the FIFO is full results in the write data being lost.

UART_DLH

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved

Bit	Attr	Reset Value	Description
7:0	RW	0x00	dlh Upper 8-bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. This register may only be accessed when the DLAB bit (LCR[7]) is set and the UART is not busy (USR[0] is zero). The output baud rate is equal to the serial clock (pclk if one clock design, sclk if two clock design CLOCK_MODE == Enabled)) frequency divided by sixteen times the value of the baud rate divisor, as follows: baud rate = (serial clock freq) / (16 * divisor). Note that with the Divisor Latch Registers (DLL and DLH) set to zero, the baud clock is disabled and no serial communications occur. Also, once the DLH is set, at least 8 clock cycles of the slowest DW_apb_uart clock should be allowed to pass before transmitting or receiving data.

UART_IER

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7	RW	0x0	ptime Programmable THRE Interrupt Mode Enable that can be written to only when THRE_MODE_USER == Enabled, always readable. This is used to enable/disable the generation of THRE Interrupt. 1'b0: Disabled 1'b1: Enabled
6:4	RO	0x0	reserved
3	RW	0x0	edssi Enable Modem Status Interrupt. This is used to enable/disable the generation of Modem Status Interrupt. This is the fourth highest priority interrupt. 1'b0: Disabled 1'b1: Enabled
2	RW	0x0	elsi Enable Receiver Line Status Interrupt. This is used to enable/disable the generation of Receiver Line Status Interrupt. This is the highest priority interrupt. 1'b0: Disabled 1'b1: Enabled

Bit	Attr	Reset Value	Description
1	RW	0x0	etbei Enable Transmit Holding Register Empty Interrupt. This is used to enable/disable the generation of Transmitter Holding Register Empty Interrupt. This is the third highest priority interrupt. 1'b0: Disabled 1'b1: Enabled
0	RW	0x0	erbfi Enable Received Data Available Interrupt. This is used to enable/disable the generation of Received Data Available Interrupt and the Character Timeout Interrupt (if in FIFO mode and FIFOs enabled). These are the second highest priority interrupts. 1'b0: Disabled 1'b1: Enabled

UART_FCR

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:6	WO	0x0	rt at which the Received Data Available Interrupt is generated. In auto flow control mode it is used to determine when the rts_n signal is de-asserted. It also determines when the dma_rx_req_n signal is asserted in certain modes of operation. For details on DMA support, refer to "DMA Suppor". The following trigger levels are supported: 2'b00: 1 character in the FIFO 2'b01: FIFO 1/4 full 2'b10: FIFO 1/2 full 2'b11: FIFO 2 less than full
5:4	WO	0x0	tet TX Empty Trigger. Writes have no effect when THRE_MODE_USER == Disabled. This is used to select the empty threshold level at which the THRE Interrupts are generated when the mode is active. It also determines when the dma_tx_req_n signal is asserted when in certain modes of operation. For details on DMA support, refer to " DMA Support" . The following trigger levels are supported: 2'b00: FIFO empty 2'b01: 2 characters in the FIFO 2'b10: FIFO 1/4 full 2'b11: FIFO 1/2 full

Bit	Attr	Reset Value	Description
3	WO	0x0	dmam DMA Mode. This determines the DMA signalling mode used for the dma_tx_req_n and dma_rx_req_n output signals when additional DMA handshaking signals are not selected (DMA_EXTRA == No). For details on DMA support, refer to DMA Support. 1'b0: Mode 0 1'b1: Mode 1
2	WO	0x0	xfifor XMIT FIFO Reset. This resets the control portion of the transmit FIFO and treats the FIFO as empty. This also de-asserts the DMA TX request and single signals when additional DMA handshaking signals are selected (DMA_EXTRA == YES). Note that this bit is 'self-clearing'. It is not necessary to clear this bit.
1	WO	0x0	rfifor RCVR FIFO Reset. This resets the control portion of the receive FIFO and treats the FIFO as empty. This also de-asserts the DMA RX request and single signals when additional DMA handshaking signals are selected (DMA_EXTRA == YES). Note that this bit is 'self-clearing'. It is not necessary to clear this bit.
0	WO	0x0	fifoe FIFO Enable. This enables/disables the transmit (XMIT) and receive (RCVR) FIFOs. Whenever the value of this bit is changed both the XMIT and RCVR controller portion of FIFOs is reset.

UART_IIR

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:6	RO	0x0	fifose FIFOs Enabled. This is used to indicate whether the FIFOs are enabled or disabled. 2'b00: Disabled 2'b11: Enabled
5:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3:0	RO	0x1	iid Interrupt ID. This indicates the highest priority pending interrupt which can be one of the following types: 4'b0000: Modem status 4'b0001: No interrupt pending 4'b0010: THR empty 4'b0100: Received data available 4'b0110: Receiver line status 4'b0111: Busy detect 4'b1100: Character timeout The interrupt priorities are split into four levels that are detailed in Table X. Bit 3 indicates an interrupt can only occur when the FIFOs are enabled and used to distinguish a Character Timeout condition interrupt.

UART_LCR

Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7	RW	0x0	dlab Divisor Latch Access Bit.Writable only when UART is not busy (USR[0] is zero), always readable. This bit is used to enable reading and writing of the Divisor Latch register (DLL and DLH) to set the baud rate of the UART. This bit must be cleared after initial baud rate setup in order to access other registers.
6	RW	0x0	bc Break Control Bit.This is used to cause a break condition to be transmitted to the receiving device. If set to one the serial output is forced to the spacing (logic 0) state. When not in Loopback Mode, as determined by MCR[4], the sout line is forced low until the Break bit is cleared. If SIR_MODE == Enabled and active (MCR[6] set to one) the sir_out_n line is continuously pulsed. When in Loopback Mode, the break condition is internally looped back to the receiver and the sir_out_n line is forced low.
5	RO	0x0	reserved
4	RW	0x0	eps Even Parity Select. Writeable only when UART is not busy (USR[0] is zero), always readable. This is used to select between even and odd parity, when parity is enabled (PEN set to one). If set to one, an even number of logic 1s is transmitted or checked. If set to zero, an odd number of logic 1s is transmitted or checked.

Bit	Attr	Reset Value	Description
3	RW	0x0	pen Parity Enable. Writeable only when UART is not busy (USR[0] is zero), always readable. This bit is used to enable and disable parity generation and detection in transmitted and received serial character respectively. 1'b0: Parity disabled 1'b1: Parity enabled
2	RW	0x0	stop 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.
1:0	RW	0x0	dls Data Length Select. Writeable only when UART is not busy (USR[0] is zero), always readable. This is used to select the number of data bits per character that the peripheral transmits and receives. The number of bit that may be selected areas follows: 2'b00: 5 bits 2'b01: 6 bits 2'b10: 7 bits 2'b11: 8 bits

UART_MCR

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:7	RO	0x0000000	reserved
6	RW	0x0	sire SIR Mode Enable. Writeable only when SIR_MODE == Enabled, always readable. This is used to enable/disable the IrDA SIR Mode features as described in "IrDA 1.0 SIR Protocol". 1'b0: IrDA SIR Mode disabled 1'b1: IrDA SIR Mode enabled

Bit	Attr	Reset Value	Description
5	RW	0x0	afce Auto Flow Control Enable. Writeable only when AFCE_MODE == Enabled, always readable. When FIFOs are enabled and the Auto Flow Control Enable (AFCE) bit is set, Auto Flow Control features are enabled as described in "Auto Flow Control". 1'b0: Auto Flow Control Mode disabled 1'b1: Auto Flow Control Mode enabled
4	RW	0x0	lb LoopBack Bit. This is used to put the UART into a diagnostic mode for test purposes. If operating in UART mode (SIR_MODE != Enabled or not active, MCR[6] set to zero), data on the sout line is held high, while serial data output is looped back to the sin line, internally. In this mode all the interrupts are fully functional. Also, in loopback mode, the modem control inputs (dsr_n, cts_n, ri_n, dcd_n) are disconnected and the modem control outputs (dtr_n, rts_n, out1_n, out2_n) are looped back to the inputs, internally. If operating in infrared mode (SIR_MODE == Enabled AND active, MCR[6] set to one), data on the sir_out_n line is held low, while serial data output is inverted and looped back to the sir_in line.
3	RW	0x0	out2 OUT2. This is used to directly control the user-designated Output2 (out2_n) output. The value written to this location is inverted and driven out on out2_n, that is: 1'b0: Out2_n de-asserted (logic 1) 1'b1: Out2_n asserted (logic 0) Note that in Loopback mode (MCR[4] set to one), the out2_n output is held inactive high while the value of this location is internally looped back to an input.
2	RW	0x0	out1 OUT1. This is used to directly control the user-designated Output1 (out1_n) output. The value written to this location is inverted and driven out on out1_n, that is: 1'b0: Out1_n de-asserted (logic 1) 1'b1: Out1_n asserted (logic 0) Note that in Loopback mode (MCR[4] set to one), the out1_n output is held inactive high while the value of this location is internally looped back to an input.

Bit	Attr	Reset Value	Description
1	RW	0x0	rts Request to Send. This is used to directly control the Request to Send (rts_n) output. The Request To Send (rts_n) output is used to inform the modem or data set that the UART is ready to exchange data. When Auto RTS Flow Control is not enabled (MCR[5] set to zero), the rts_n signal is set low by programming MCR[1] (RTS) to a high.In Auto Flow Control, AFCE_MODE == Enabled and active (MCR[5] set to one) and FIFOs enable (FCR[0] set to one), the rts_n output is controlled in the same way, but is also gated with the receiver FIFO threshold trigger (rts_n is inactive high when above the threshold). The rts_n signal is de-asserted when MCR[1] is set low. Note that in Loopback mode (MCR[4] set to one), the rts_n output is held inactive high while the value of this location is internally looped back to an input.
0	RW	0x0	dtr Data Terminal Ready. This is used to directly control the Data Terminal Ready (dtr_n) output. The value written to this location is inverted and driven out on dtr_n, that is: 1'b0: dtr_n de-asserted (logic 1) 1'b1: dtr_n asserted (logic 0) The Data Terminal Ready output is used to inform the modem or data set that the UART is ready to establish communications. Note that in Loopback mode (MCR[4] set to one), the dtr_n output is held inactive

UART_LSR

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7	RO	0x0	rfe Receiver FIFO Error bit. This bit is only relevant when FIFO_MODE != NONE AND FIFOs are enabled (FCR[0] set to one). This is used to indicate if there is at least one parity error, framing error, or break indication in the FIFO. 1'b0: No error in RX FIFO 1'b1: Error in RX FIFO This bit is cleared when the LSR is read and the character with the error is at the top of the receiver FIFO and there are no subsequent errors in the FIFO.

Bit	Attr	Reset Value	Description
6	RO	0x1	temt Transmitter Empty bit. If in FIFO mode (FIFO_MODE != NONE) and FIFOs enabled (FCR[0] set to one), this bit is set whenever the Transmitter Shift Register and the FIFO are both empty. If in non-FIFO mode or FIFOs are disabled, this bit is set whenever the Transmitter Holding Register and the Transmitter Shift Register are both empty.
5	RO	0x1	thre Transmit Holding Register Empty bit. If THRE_MODE_USER == Disabled or THRE mode is disabled (IER[7] set to zero) and regardless of FIFO's being implemented/enabled or not, this bit indicates that the THR or TX FIFO is empty. This bit is set whenever data is transferred from the THR or TX FIFO to the transmitter shift register and no new data has been written to the THR or TX FIFO. This also causes a THRE Interrupt to occur, if the THRE Interrupt is enabled. If THRE_MODE_USER == Enabled AND FIFO_MODE != NONE and both modes are active (IER[7] set to one and FCR[0] set to one respectively), the functionality is switched to indicate the transmitter FIFO is full, and no longer controls THRE interrupts, which are then controlled by the FCR[5:4] threshold setting.
4	RO	0x0	bi Break Interrupt bit. This is used to indicate the detection of a break sequence on the serial input data. If in UART mode (SIR_MODE == Disabled), it is set whenever the serial input, sin, is held in a logic '0' state for longer than the sum of start time + data bits + parity + stop bits. If in infrared mode (SIR_MODE == Enabled), it is set whenever the serial input, sir_in, is continuously pulsed to logic '0' for longer than the sum of start time + data bits + parity + stop bits. A break condition on serial input causes one and only one character, consisting of all zeros, to be received by the UART. In the FIFO mode, the character associated with the break condition is carried through the FIFO and is revealed when the character is at the top of the FIFO. Reading the LSR clears the BI bit. In the non-FIFO mode, the BI indication occurs immediately and persists until the LSR is read.

Bit	Attr	Reset Value	Description
3	RO	0x0	fe Framing Error bit. This is used to indicate the occurrence of a framing error in the receiver. A framing error occurs when the receiver does not detect a valid STOP bit in the received data. In the FIFO mode, since the framing error is associated with a character received, it is revealed when the character with the framing error is at the top of the FIFO. When a framing error occurs, the UART tries to resynchronize. It does this by assuming that the error was due to the start bit of the next character and then continues receiving the other bit i.e. data, and/or parity and stop. It should be noted that the Framing Error (FE) bit (LSR[3]) is set if a break interrupt has occurred, as indicated by Break Interrupt (BI) bit (LSR[4]). 1'b0: No framing error 1'b1: Framing error Reading the LSR clears the FE bit.
2	RO	0x0	pe Parity Error bit. This is used to indicate the occurrence of a parity error in the receiver if the Parity Enable (PEN) bit (LCR[3]) is set. In the FIFO mode, since the parity error is associated with a character received, it is revealed when the character with the parity error arrives at the top of the FIFO. It should be noted that the Parity Error (PE) bit (LSR[2]) is set if a break interrupt has occurred, as indicated by Break Interrupt (BI) bit (LSR[4]). 1'b0: No parity error 1'b1: Parity error Reading the LSR clears the PE bit.
1	RO	0x0	oe Overrun error bit. This is used to indicate the occurrence of an overrun error. This occurs if a new data character was received before the previous data was read. In the non-FIFO mode, the OE bit is set when a new character arrives in the receiver before the previous character was read from the RBR. When this happens, the data in the RBR is overwritten. In the FIFO mode, an overrun error occurs when the FIFO is full and a new character arrives at the receiver. The data in the FIFO is retained and the data in the receive shift register is lost. 1'b0: No overrun error 1'b1: Overrun error Reading the LSR clears the OE bit.

Bit	Attr	Reset Value	Description
0	RO	0x0	dr Data Ready bit. This is used to indicate that the receiver contains at least one character in the RBR or the receiver FIFO. 1'b0: No data ready 1'b1: Data ready This bit is cleared when the RBR is read in non-FIFO mode, or when the receiver FIFO is empty, in FIFO mode.

UART_MSR

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7	RO	0x0	dcd Data Carrier Detect. This is used to indicate the current state of the modem control line dcd_n. This bit is the complement of dcd_n. When the Data Carrier Detect input (dcd_n) is asserted it is an indication that the carrier has been detected by the modem or data set. 1'b0: dcd_n input is de-asserted (logic 1) 1'b1: dcd_n input is asserted (logic 0) In Loopback Mode (MCR[4] set to one), DCD is the same as MCR[3] (Out2).
6	RO	0x0	ri Ring Indicator. This is used to indicate the current state of the modem control line ri_n. This bit is the complement of ri_n. When the Ring Indicator input (ri_n) is asserted it is an indication that a telephone ringing signal has been received by the modem or data set. 1'b0: ri_n input is de-asserted (logic 1) 1'b1: ri_n input is asserted (logic 0) In Loopback Mode (MCR[4] set to one), RI is the same as MCR[2] (Out1).
5	RO	0x0	dsr Data Set Ready. This is used to indicate the current state of the modem control line dsr_n. This bit is the complement of dsr_n. When the Data Set Ready input (dsr_n) is asserted it is an indication that the modem or data set is ready to establish communications with the DW_apb_uart. 1'b0: dsr_n input is de-asserted (logic 1) 1'b1: dsr_n input is asserted (logic 0) In Loopback Mode (MCR[4] set to one), DSR is the same as MCR[0] (DTR).

Bit	Attr	Reset Value	Description
4	RO	0x0	cts Clear to Send. This is used to indicate the current state of the modem control line cts_n. This bit is the complement of cts_n. When the Clear to Send input (cts_n) is asserted it is an indication that the modem or data set is ready to exchange data with the DW_apb_uart. 1'b0: cts_n input is de-asserted (logic 1) 1'b0: cts_n input is asserted (logic 0) In Loopback Mode (MCR[4] = 1), CTS is the same as MCR[1] (RTS).
3	RO	0x0	ddcd Delta Data Carrier Detect. This is used to indicate that the modem control line dcd_n has changed since the last time the MSR was read. 1'b0: No change on dcd_n since last read of MSR 1'b1: Change on dcd_n since last read of MSR Reading the MSR clears the DDCD bit. In Loopback Mode (MCR[4] = 1), DDCD reflects changes on MCR[3] (Out2). Note, if the DDCD bit is not set and the dcd_n signal is asserted (low) and a reset occurs (software or otherwise), then the DDCD bit is set when the reset is removed if the dcd_n signal remains asserted.
2	RO	0x0	teri Trailing Edge of Ring Indicator. This is used to indicate that a change on the input ri_n (from an active-low to an inactive-high state) has occurred since the last time the MSR was read. 1'b0: No change on ri_n since last read of MSR 1'b1: Change on ri_n since last read of MSR Reading the MSR clears the TERI bit. In Loopback Mode (MCR[4] = 1), TERI reflects when MCR[2] (Out1) has changed state from a high to a low.
1	RO	0x0	ddsr Delta Data Set Ready. This is used to indicate that the modem control line dsr_n has changed since the last time the MSR was read. 1'b0: No change on dsr_n since last read of MSR 1'b1: Change on dsr_n since last read of MSR Reading the MSR clears the DDSR bit. In Loopback Mode (MCR[4] = 1), DDSR reflects changes on MCR[0] (DTR). Note, if the DDSR bit is not set and the dsr_n signal is asserted (low) and a reset occurs (software or otherwise), then the DDSR bit is set when the reset is removed if the dsr_n signal remains asserted.

Bit	Attr	Reset Value	Description
0	RO	0x0	dcts Delta Clear to Send. This is used to indicate that the modem control line cts_n has changed since the last time the MSR was read. 1'b0: No change on ctsdsr_n since last read of MSR 1'b1: Change on ctsdsr_n since last read of MSR Reading the MSR clears the DCTS bit. In Loopback Mode (MCR[4] = 1), DCTS reflects changes on MCR[1] (RTS). Note, if the DCTS bit is not set and the cts_n signal is asserted (low) and a reset occurs (software or otherwise), then the DCTS bit is set when the reset is removed if the cts_n signal remains asserted.

UART_SCR

Address: Operational Base + offset (0x001c)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x00	scr Scratchpad register.This register is for programmers to use as a temporary storage space. It has no defined purpose in the DW_apb_uart.

UART_SRBR

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RO	0x00	srbr This is a shadow register for the RBR and has been allocated sixteen 32-bit locations (0x30-0x6c) so as to accommodate burst accesses from the master. This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If in non-FIFO mode (FIFO_MODE == NONE) or FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it is overwritten, resulting in an overrun error. If in FIFO mode (FIFO_MODE != NONE) and FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO are preserved, but any incoming data is lost. An overrun error also occurs.

UART_STHR

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	WO	0x0	sthr This is a shadow register for the THR and has been allocated sixteen 32-bit locations(0x30-0x6c) so as to accommodate burst accesses from the master. This register contains data to be transmitted on the serial output port (sout) in UART mode or the serial infrared output (sir_out_n) in infrared mode. Data should only be written to the THR when the THR Empty (THRE) bit (LSR[5]) is set. If in non-FIFO mode or FIFOs are disabled (FCR[0] set to zero) and THRE is set, writing a single character to the THR clears the THRE. Any additional writes to the THR before the THRE is set again causes the THR data to be overwritten. If in FIFO mode and FIFOs are enabled (FCR[0] set to one) and THRE is set, x number of characters of data may be written to the THR before the FIFO is full. The number x (default=16) is determined by the value of FIFO Depth that you set during configuration. Any attempt to write data when the FIFO is full results in the write data being lost.

UART_FAR

Address: Operational Base + offset (0x0070)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RW	0x0	far Writes have no effect when FIFO_ACCESS == No, always readable. 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 implemented or not enabled it allows the RBR to be written by the master and the THR to be read by the master. 1'b0: FIFO access mode disabled 1'b1: FIFO access mode enabled Note, that when the FIFO access mode is enabled/disabled, the control portion of the receive FIFO and transmit FIFO is reset and the FIFOs are treated as empty.

UART_TFR

Address: Operational Base + offset (0x0074)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved

Bit	Attr	Reset Value	Description
7:0	RO	0x00	tfr Transmit FIFO Read. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one). When FIFOs are implemented and enabled, reading this register gives the data at the top of the transmit FIFO. Each consecutive read pops the transmit FIFO and gives the next data value that is currently at the top of the FIFO. When FIFOs are not implemented or not enabled, reading this register gives the data in the THR.

UART_RFW

Address: Operational Base + offset (0x0078)

Bit	Attr	Reset Value	Description
31:10	RO	0x000000	reserved
9	WO	0x0	refe Receive FIFO Framing Error. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one). When FIFOs are implemented and enabled, this bit is used to write framing error detection information to the receive FIFO. When FIFOs are not implemented or not enabled, this bit is used to write framing error detection information to the RBR.
8	WO	0x0	repe Receive FIFO Parity Error. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one). When FIFOs are implemented and enabled, this bit is used to write parity error detection information to the receive FIFO. When FIFOs are not implemented or not enabled, this bit is used to write parity error detection information to the RBR.
7:0	WO	0x00	rfwd Receive FIFO Write Data. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one). When FIFOs are implemented and enabled, the data that is written to the RFWD is pushed into the receive FIFO. Each consecutive write pushes the new data to the next write location in the receive FIFO. When FIFOs are not implemented or not enabled, the data that is written to the RFWD is pushed into the RBR.

UART_USR

Address: Operational Base + offset (0x007c)

Bit	Attr	Reset Value	Description
31:5	RO	0x0000000	reserved

Bit	Attr	Reset Value	Description
4	RO	0x0	rfe Receive FIFO Full. This bit is only valid when FIFO_STAT == YES. This is used to indicate that the receive FIFO is completely full. 1'b0: Receive FIFO not full 1'b1: Receive FIFO Full This bit is cleared when the RX FIFO is no longer full.
3	RO	0x0	rfne Receive FIFO Not Empty. This bit is only valid when FIFO_STAT == YES. This is used to indicate that the receive FIFO contains one or more entries. 1'b0: Receive FIFO is empty 1'b1: Receive FIFO is not empty This bit is cleared when the RX FIFO is empty.
2	RO	0x1	tfe Transmit FIFO Empty. This bit is only valid when FIFO_STAT == YES. This is used to indicate that the transmit FIFO is completely empty. 1'b0: Transmit FIFO is not empty 1'b1: Transmit FIFO is empty This bit is cleared when the TX FIFO is no longer empty.
1	RO	0x1	tfnf Transmit FIFO Not Full. This bit is only valid when FIFO_STAT == YES. This is used to indicate that the transmit FIFO in not full. 1'b0: Transmit FIFO is full 1'b1: Transmit FIFO is not full This bit is cleared when the TX FIFO is full.
0	RO	0x0	busy UART Busy. This is indicates that a serial transfer is in progress, when cleared indicates that the DW_apb_uart is idle or inactive. 1'b0: DW_apb_uart is idle or inactive 1'b1: DW_apb_uart is busy (actively transferring data) Note that it is possible for the UART Busy bit to be cleared even though a new character may have been sent from another device. That is, if the DW_apb_uart has no data in THR and RBR and there is no transmission in progress and a start bit of a new character has just reached the DW_apb_uart. This is due to the fact that a valid start is not seen until the middle of the bit period and this duration is dependent on the baud divisor that has been programmed. If a second system clock has been implemented (CLOCK_MODE == Enabled), the assertion of this bit is also delayed by several cycles of the slower clock.

UART_TFL

Address: Operational Base + offset (0x0080)

Bit	Attr	Reset Value	Description
31:6	RO	0x0000000	reserved
5:0	RO	0x00	tfl Transmit FIFO Level. This is indicates the number of data entries in the transmit FIFO.

UART_RFL

Address: Operational Base + offset (0x0084)

Bit	Attr	Reset Value	Description
31:6	RO	0x0000000	reserved
5:0	RO	0x00	rfl Receive FIFO Level. This is indicates the number of data entries in the receive FIFO.

UART_SRR

Address: Operational Base + offset (0x0088)

Bit	Attr	Reset Value	Description
31:3	RO	0x00000000	reserved
2	WO	0x0	xfr XMIT FIFO Reset. This is a shadow register for the XMIT FIFO Reset bit (FCR[2]). This can be used to remove the burden on software having to store previously written FCR values (which are pretty static) just to reset the transmit FIFO. This resets the control portion of the transmit FIFO and treats the FIFO as empty. This also de-asserts the DMA TX request and single signals when additional DMA handshaking signals are selected (DMA_EXTRA == YES). Note that this bit is 'self-clearing'. It is not necessary to clear this bit.
1	WO	0x0	rfr RCVR FIFO Reset. This is a shadow register for the RCVR FIFO Reset bit (FCR[1]). This can be used to remove the burden on software having to store previously written FCR values (which are pretty static) just to reset the receive FIFO This resets the control portion of the receive FIFO and treats the FIFO as empty. This also de-asserts the DMA RX request and single signals when additional DMA handshaking signals are selected (DMA_EXTRA == YES). Note that this bit is 'self-clearing'. It is not necessary to clear this bit.
0	WO	0x0	ur UART Reset. This asynchronously resets the DW_apb_uart and synchronously removes the reset assertion. For a two clock implementation both pclk and sclk domains are reset.

UART_SRTS

Address: Operational Base + offset (0x008c)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RW	0x0	srts Shadow Request to Send. This is a shadow register for the RTS bit (MCR[1]), this can be used to remove the burden of having to performing a read-modify-write on the MCR. This is used to directly control the Request to Send (rts_n) output. The Request To Send (rts_n) output is used to inform the modem or data set that the DW_apb_uart is ready to exchange data. When Auto RTS Flow Control is not enabled (MCR[5] = 0), the rts_n signal is set low by programming MCR[1] (RTS) to a high. In Auto Flow Control, AFCE_MODE == Enabled and active (MCR[5] = 1) and FIFOs enable (FCR[0] = 1), the rts_n output is controlled in the same way, but is also gated with the receiver FIFO threshold trigger (rts_n is inactive high when above the threshold). Note that in Loopback mode (MCR[4] = 1), the rts_n output is held inactive-high while the value of this location is internally looped back to an input.

UART_SBCR

Address: Operational Base + offset (0x0090)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RW	0x0	sbcr Shadow Break Control Bit. This is a shadow register for the Break bit (LCR[6]), this can be used to remove the burden of having to performing a read modify write on the LCR. This is used to cause a break condition to be transmitted to the receiving device. If set to one the serial output is forced to the spacing (logic 0) state. When not in Loopback Mode, as determined by MCR[4], the sout line is forced low until the Break bit is cleared. If SIR_MODE == Enabled and active (MCR[6] = 1) the sir_out_n line is continuously pulsed. When in Loopback Mode, the break condition is internally looped back to the receiver.

UART_SDMAM

Address: Operational Base + offset (0x0094)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	sdmam Shadow DMA Mode. This is a shadow register for the DMA mode bit (FCR[3]). This can be used to remove the burden of having to store the previously written value to the FCR in memory and having to mask this value so that only the DMA Mode bit gets updated. This determines the DMA signalling mode used for the dma_tx_req_n and dma_rx_req_n output signals when additional DMA handshaking signals are not selected (DMA_EXTRA == NO). 1'b0: Mode 0 1'b1: Mode 1

UART_SFE

Address: Operational Base + offset (0x0098)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RW	0x0	sfe Shadow FIFO Enable. This is a shadow register for the FIFO enable bit (FCR[0]). This can be used to remove the burden of having to store the previously written value to the FCR in memory and having to mask this value so that only the FIFO enable bit gets updated.This enables/disables the transmit (XMIT) and receive (RCVR) FIFOs. If this bit is set to zero (disabled) after being enabled then both the XMIT and RCVR controller portion of FIFOs are reset.

UART_SRT

Address: Operational Base + offset (0x009c)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved
1:0	RW	0x0	srt Shadow RCVR Trigger. This is a shadow register for the RCVR trigger bits (FCR[7:6]). This can be used to remove the burden of having to store thepreviously written value to the FCR in memory and having to mask thisvalue so that only the RCVR trigger bit gets updated.This is used to select the trigger level in the receiver FIFO at which theReceived Data Available Interrupt is generated. It also determines when thedma_rx_req_n signal is asserted when DMA Mode (FCR[3]) = 1. The following trigger levels are supported: 2'b00: 1 character in the FIFO 2'b01: FIFO 1/4 full 2'b10: FIFO 1/2 full 2'b11: FIFO 2 less than full

UART_STET

Address: Operational Base + offset (0x00a0)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved
1:0	RW	0x0	stet Shadow TX Empty Trigger. This is a shadow register for the TX empty trigger bits (FCR[5:4]). This can be used to remove the burden of having to store the previously written value to the FCR in memory and having to mask this value so that only the TX empty trigger bit gets updated.This is used to select the empty threshold level at which the THRE Interrupts are generated when the mode is active. The following trigger levels are supported: 2'b00: FIFO empty 2'b01: 2 characters in the FIFO 2'b10: FIFO 1/4 full 2'b11: FIFO 1/2 full

UART_HTX

Address: Operational Base + offset (0x00a4)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RW	0x0	htx This register is use to halt transmissions for testing, so that the transmit FIFO can be filled by the master when FIFOs are implemented and enabled. 1'b0: Halt TX disabled 1'b1: Halt TX enabled Note, if FIFOs are implemented and not enabled, the setting of the halt TX register has no effect on operation.

UART_DMASA

Address: Operational Base + offset (0x00a8)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	WO	0x0	dmasa This register is use to perform a DMA software acknowledge if a transfer needs to be terminated due to an error condition. For example, if the DMA disables the channel, then the DW_apb_uart should clear its request. This causes the TX request, TX single, RX request and RX single signals to de-assert. Note that this bit is 'self-clearing'. It is not necessary to clear this bit.

UART_CPR

Address: Operational Base + offset (0x00f4)

Bit	Attr	Reset Value	Description
31:24	RO	0x00	reserved
23:16	RO	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: Ture
12	RO	0x0	uart_add_encoded_params 1'b0: False 1'b1: Ture
11	RO	0x0	shadow 1'b0: False 1'b1: Ture
10	RO	0x0	fifo_stat 1'b0: False 1'b1: Ture
9	RO	0x0	fifo_access 1'b0: False 1'b1: Ture
8	RO	0x0	new_feat 1'b0: False 1'b1: Ture
7	RO	0x0	sir_lp_mode 1'b0: False 1'b1: Ture
6	RO	0x0	sir_mode 1'b0: False 1'b1: Ture
5	RO	0x0	thre_mode 1'b0: False 1'b1: Ture
4	RO	0x0	afce_mode 1'b0: False 1'b1: Ture
3:2	RO	0x0	reserved
1:0	RO	0x0	apb_data_width 2'b00: 8 bits 2'b01: 16 bits 2'b10: 32 bits 2'b11: reserved

UART_UCV

Address: Operational Base + offset (0x00f8)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	ucv ASCII value for each number in the version.

UART_CTR

Address: Operational Base + offset (0x00fc)

Bit	Attr	Reset Value	Description
31:0	RO	0x44570110	id This register contains the peripherals identification code.

24.5 Interface Description

Table 23-2UART Interface Description

Module Pin	Direction	Pad Name	IOMUX Setting
UART0mux0 Interface			
uart0_sin	I	GPIO1_C2/UART0_RX	GRF_GPIO1C_IOMUX_SEL_L[11:8]=4'h1
uart0_sout	O	GPIO1_C3/UART0_TX	GRF_GPIO1C_IOMUX_SEL_L[15:12]=4'h1
uart0_cts_n	I	GPIO1_C1/UART0_CTSN	GRF_GPIO1C_IOMUX_SEL_L[7:4]=4'h1
uart0_rts_n	O	GPIO1_C0/UART0_RTSN	GRF_GPIO1C_IOMUX_SEL_L[3:0]=4'h1
UART1mux0 Interface			
uart1_sin	I	GPIO0_B7/UART1_RX_M0/PWM1_M0	PMUGRF_GPIO0B_IOMUX_SEL_H[15:12]=4'h2
uart1_sout	O	GPIO0_B6/UART1_TX_M0/PWM0_M0	PMUGRF_GPIO0B_IOMUX_SEL_H[11:8]=4'h2
uart1_cts_n	I	GPIO0_C1/PMU_DEBUG/UART1_CTSN_M0/PWM3_IR_M0	PMUGRF_GPIO0C_IOMUX_SEL_L[7:4]=4'h2
uart1_rts_n	O	GPIO0_C0/SDMMC0_PWR/UART1_RTSN_M0/PWM2_M0	PMUGRF_GPIO0C_IOMUX_SEL_L[3:0]=4'h2
UART1mux1 Interface			
uart1_sin	I	GPIO1_D1/SDMMC1_PWR/I2C5_SDA_M2/UART1_RX_M1	GRF_GPIO1D_IOMUX_SEL_L[7:4]=4'h5
uart1_sout	O	GPIO1_D0/I2S2_MCLK_M0/SDMMC1_DET/SPI1_CS1N_M1/I2C5_SCL_M2/UART1_TX_M1	GRF_GPIO1D_IOMUX_SEL_L[3:0]=4'h5
uart1_cts_n	I	GPIO1_C7/I2S2_LRCK_M0/SPI1_CS0N	GRF_GPIO1C_IOMUX

		_M1/UART1_CTSN_M1	_SEL_H[15:12]=4'h5
uart1_rts_n	O	GPIO1_C6/I2S2_SCLK_M0/SPI1_CLK_M1/PRELIGHT_TRIG_OUT/UART1_RTSN_M1	GRF_GPIO1C_IOMUX_SEL_H[11:8]=4'h5
UART2mux0 Interface			
uart2_sin	I	GPIO1_A4/SDMMC0_D0/TEST_CLK1_OUT/UART2_RX_M0	GRF_GPIO1A_IOMUX_SEL_H[3:0]=4'h3
uart2_sout	O	GPIO1_A5/SDMMC0_D1/TEST_CLK0_OUT/UART2_TX_M0/RISC-V_JTAG_TRSTn	GRF_GPIO1A_IOMUX_SEL_H[7:4]=4'h3
UART2mux1 Interface			
uart2_sin	I	GPIO3_A3/UART2_RX_M1/A7_JTAG_TMS_M1	GRF_GPIO3A_IOMUX_SEL_L[11:8]=4'h1
uart2_sout	O	GPIO3_A2/UART2_TX_M1/A7_JTAG_TCK_M1	GRF_GPIO3A_IOMUX_SEL_L[15:12]=4'h1
UART3mux0 Interface			
uart3_sin	I	GPIO3_C7/CIF_HSYNC_M0/RGMII_RXCLK_M0/UART3_RX_M0	GRF_GPIO3C_IOMUX_SEL_H[15:12]=4'h4
uart3_sout	O	GPIO3_C6/CIF_CLKOUT_M0/RGMII_TXCLK_M0/UART3_TX_M0	GRF_GPIO3C_IOMUX_SEL_H[11:8]=4'h4
uart3_cts_n	I	GPIO3_C5/CIF_CLKIN_M0/CLK_OUT_ETHERNET_M0/UART3_CTSN_M0	GRF_GPIO3C_IOMUX_SEL_H[7:4]=4'h4
uart3_rts_n	O	GPIO3_C4/CIF_VSYNC_M0/RGMII_MDC_M0/UART3_RTSN_M0	GRF_GPIO3C_IOMUX_SEL_H[3:0]=4'h4
UART3mux1 Interface			
uart3_sin	I	GPIO1_A6/SDMMC0_D2/UART3_RX_M1/A7_JTAG_TCK_M0/RISC-V_JTAG_TCK	GRF_GPIO1A_IOMUX_SEL_H[11:8]=4'h2
uart3_sout	O	GPIO1_A7/SDMMC0_D3/UART3_TX_M1/A7_JTAG_TMS_M0/RISC-V_JTAG_TMS	GRF_GPIO1A_IOMUX_SEL_H[15:12]=4'h2
uart3_cts_n	I	GPIO1_B1/SDMMC0_CMD/UART3_CTSN_M1/RISC-V_JTAG_TDI	GRF_GPIO1B_IOMUX_SEL_L[7:4]=4'h2
uart3_rts_n	O	GPIO1_B0/SDMMC0_CLK/UART3_RTSN_M1/RISC-V_JTAG_TDO	GRF_GPIO1B_IOMUX_SEL_L[3:0]=4'h2
UART3mux2 Interface			
uart3_sin	I	GPIO3_A1/CAN_TXD_M0/UART3_RX_M2/PWM11_IR_M1/I2C4_SDA_M0	GRF_GPIO3A_IOMUX_SEL_L[7:4]=4'h4
uart3_sout	O	GPIO3_A0/CAN_RXD_M0/UART3_TX_M2/PWM7_IR_M1/SPI1_CS1N_M2/I2C4_SCL_M0	GRF_GPIO3A_IOMUX_SEL_L[3:0]=4'h4
uart3_cts_n	I	GPIO2_D7/LCDC_CLK/UART3_CTSN_M2/PWM8_M1/SPI1_MISO_M2	GRF_GPIO2D_IOMUX_SEL_H[15:12]=4'h4
uart3_rts_n	O	GPIO2_D6/LCDC_VSYNC/UART3_RTSN_M2/PWM9_M1/SPI1_MOSI_M2	GRF_GPIO2D_IOMUX_SEL_H[11:8]=4'h4
UART4mux0 Interface			
uart4_sin	I	GPIO3_A5/CIF_D1_M0/RGMII_CRS_M0/I2S0_LRCK_TX_M1/UART4_RX_M0/I2	GRF_GPIO3A_IOMUX_SEL_H[7:4]=4'h4

		C3_SDA_M0/PWM9_M0	
uart4_sout	O	GPIO3_A4/CIF_D0_M0/I2S0_SCLK_TX_M1/UART4_TX_M0/I2C3_SCL_M0/PWM8_M0	GRF_GPIO3A_IOMUX_SEL_H[3:0]=4'h4
uart4_cts_n	I	GPIO3_B3/CIF_D7_M0/RGMII_TXD0_M0/I2S0_SDO1_SDI3_M1/UART4_CTSN_M0	GRF_GPIO3B_IOMUX_SEL_L[15:12]=4'h4
uart4_rts_n	O	GPIO3_B2/CIF_D6_M0/RGMII_TXD3_M0/I2S0_LRCK_RX_M1/UART4_RTSN_M0	GRF_GPIO3B_IOMUX_SEL_L[11:8]=4'h4
UART4mux1 Interface			
uart4_sin	I	GPIO2_A7/LCDC_D3/I2S2_SDO_M1/UART4_RX_M1/PWM4_M1/SPI0_CS0N_M2	GRF_GPIO2A_IOMUX_SEL_H[15:12]=4'h4
uart4_sout	O	GPIO2_A6/LCDC_D2/RGMII_COL_M1/CIF_D2_M1/UART4_TX_M1/PWM5_M1	GRF_GPIO2A_IOMUX_SEL_H[11:8]=4'h4
uart4_cts_n	I	GPIO2_A5/LCDC_D1/RGMII_CRS_M1/CIF_D1_M1/UART4_CTSN_M1/I2C5_SCL_M0	GRF_GPIO2A_IOMUX_SEL_H[7:4]=4'h4
uart4_rts_n	O	GPIO2_A4/LCDC_D0/RGMII_TXD3_M1/CIF_D0_M1/UART4_RTSN_M1	GRF_GPIO2A_IOMUX_SEL_H[3:0]=4'h4
UART4mux2 Interface			
uart4_sin	I	GPIO1_D4/UART4_RX_M2	GRF_GPIO1D_IOMUX_SEL_H[3:0]=4'h3
uart4_sout	O	GPIO1_D5/SPI0_CS1n_M1/I2S1_MCLK_M1/UART4_TX_M2	GRF_GPIO1D_IOMUX_SEL_H[7:4]=4'h3
uart4_cts_n	I	GPIO1_D3/I2C1_SCL/UART4_CTSN_M2	GRF_GPIO1D_IOMUX_SEL_L[15:12]=4'h3
uart4_rts_n	O	GPIO1_D2/I2C1_SDA/UART4_RTSN_M2	GRF_GPIO1D_IOMUX_SEL_L[11:8]=4'h3
UART5mux0 Interface			
uart5_sin	I	GPIO3_A7/CIF_D3_M0/RGMII_RXD2_M0/I2S0_SDI0_M1/UART5_RX_M0/CAN_TXD_M1/PWM11_IR_M0	GRF_GPIO3A_IOMUX_SEL_H[15:12]=4'h4
uart5_sout	O	GPIO3_A6/CIF_D2_M0/RGMII_COL_M0/I2S0_SDO0_M1/UART5_TX_M0/CAN_RXD_M1/PWM10_M0	GRF_GPIO3A_IOMUX_SEL_H[11:8]=4'h4
uart5_cts_n	I	GPIO3_B1/CIF_D5_M0/RGMII_TXD2_M0/I2S0_SCLK_RX_M1/UART5_CTSN_M0/I2C5_SDA_M1	GRF_GPIO3B_IOMUX_SEL_L[7:4]=4'h4
uart5_rts_n	O	GPIO3_B0/CIF_D4_M0/RGMII_RXD3_M0/I2S0_MCLK_M1/UART5_RTSN_M0/I2C5_SCL_M1	GRF_GPIO3B_IOMUX_SEL_L[3:0]=4'h4
UART5mux1 Interface			
uart5_sin	I	GPIO2_B1/LCDC_D5/I2S2_SCLK_M1/UART5_RX_M1/PWM2_M1/SPI0_MISO_M2	GRF_GPIO2B_IOMUX_SEL_L[7:4]=4'h4
uart5_sout	O	GPIO2_B0/LCDC_D4/I2S2_SDI_M1/UA	GRF_GPIO2B_IOMUX

		RT5_TX_M1/PWM3_IR_M1/SPI0_MOSI_M2	_SEL_L[3:0]=4'h4
uart5_cts_n	I	GPIO2_B3/LCDC_D7/I2S2_MCLK_M1/CIF_D3_M1/UART5_CTSN_M1/PWM0_M1/SPI0_CS1N_M2/I2C5_SDA_M0	GRF_GPIO2B_IOMUX_SEL_L[15:12]=4'h4
uart5_rts_n	O	GPIO2_B2/LCDC_D6/I2S2_LRCK_M1/UART5_RTSN_M1/PWM1_M1/SPI0_CLK_M2	GRF_GPIO2B_IOMUX_SEL_L[11:8]=4'h4
UART5mux2 Interface			
uart5_sin	I	GPIO2_A1/SPI0_CLK_M1/I2S1_SDO_M1/UART5_RX_M2	GRF_GPIO2A_IOMUX_SEL_L[7:4]=4'h3
uart5_sout	O	GPIO2_A0/SPI0_CS0n_M1/I2S1_SDI_M1/UART5_TX_M2	GRF_GPIO2A_IOMUX_SEL_L[3:0]=4'h3
uart5_cts_n	I	GPIO2_A3/MIPI_CSI_CLK0/UART5_CTSN_M2	GRF_GPIO2A_IOMUX_SEL_L[15:12]=4'h3
uart5_rts_n	O	GPIO2_A2/MIPI_CSI_CLK1/UART5_RTSN_M2	GRF_GPIO2A_IOMUX_SEL_L[11:8]=4'h3

24.6 Application Notes

24.6.1 None FIFO Mode Transfer Flow

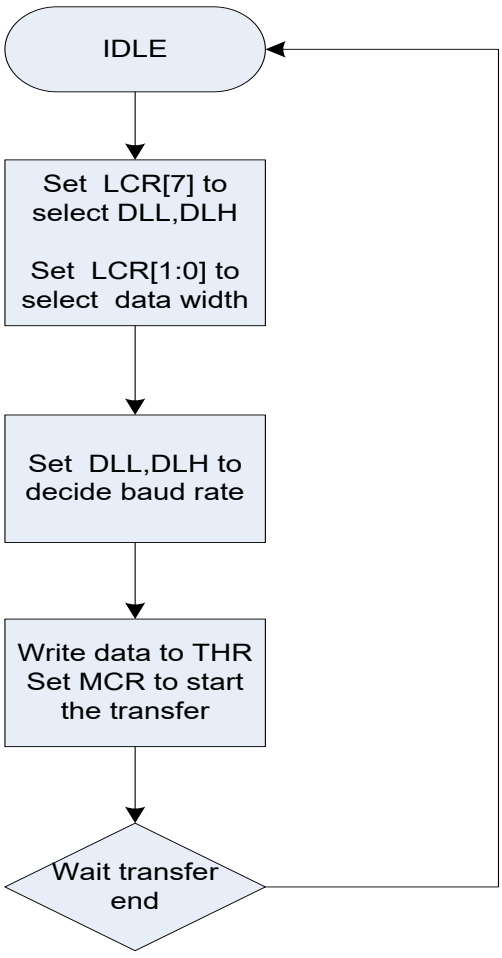


Fig. 23-16 UART none fifo mode

24.6.2 FIFO Mode Transfer Flow

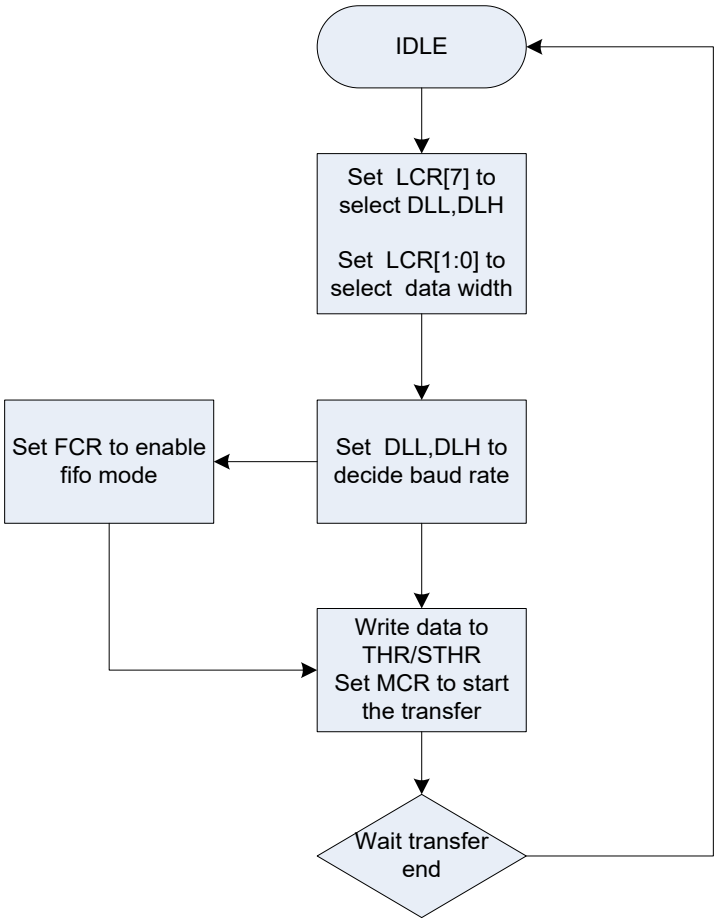


Fig. 23-17 UART fifo mode

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.

24.6.3 Baud Rate Calculation

UART clock generation

The following figures shows the UART clock generation.
UARTs source clocks can be selected from different PLL outputs. UART clocks can be generated by 1 to 64 division of its source clock, or can be fractionally divided again, or be provided by XIN24M.

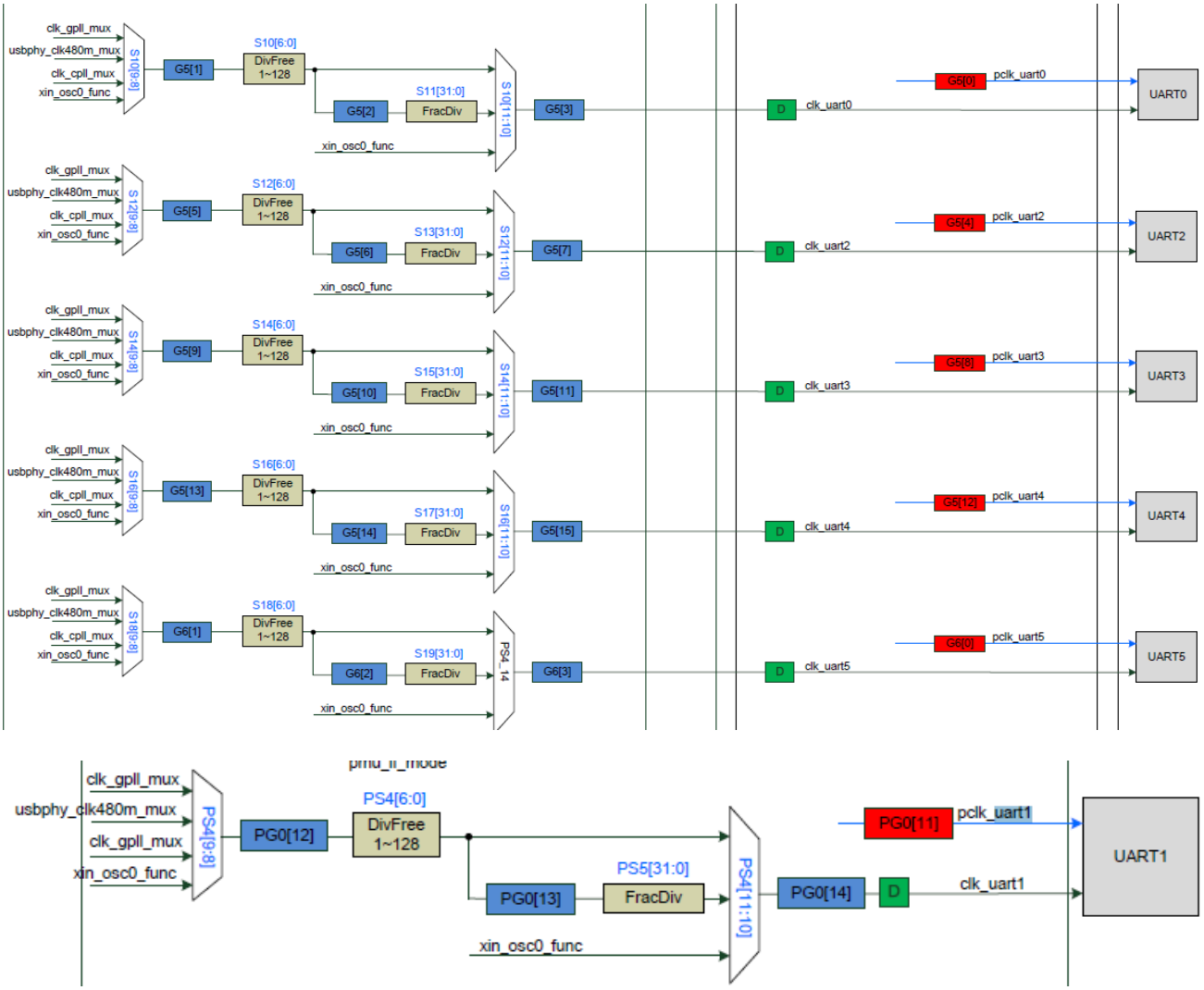


Fig. 23-18 UART clock generation

UART baud rate configuration

The following table provides some reference configuration for different UART baud rates.

Table 23-3 UART baud rate configuration

Baud Rate	Reference Configuration
115.2 Kbps	Configure PLL to get 648MHz clock output; Divide 648MHz clock by 1152/50625 to get 14.7456MHz clock;Configure UART_DLL to 8.
460.8 Kbps	Configure PLL to get 648MHz clock output; Divide 648MHz clock by 1152/50625 to get 14.7456MHz clock;Configure UART_DLL to 2.
921.6 Kbps	Configure PLL to get 648MHz clock output; Divide 648MHz clock by 1152/50625 to get 14.7456MHz clock;Configure UART_DLL to 1.
1.5 Mbps	Choose PLL to get 384MHz clock output;Divide 384MHz clock by 16 to get 24MHz clock; Configure UART_DLL to 1
3 Mbps	Choose PLL to get 384MHz clock output;Divide 384MHz clock by 8 to get 48MHz clock; Configure UART_DLL to 1
4 Mbps	Configure PLL to get 384MHz clock output;Divide 384MHz clock by 6 to get 64MHz clock; Configure UART_DLL to 1

Chapter 25 Serial Peripheral Interface (SPI)

25.1 Overview

The serial peripheral interface is an APB slave device. A four wire full duplex serial protocol from Motorola. There are four possible combinations for the serial clock phase and polarity. The clock phase (SCPH) determines whether the serial transfer begins with the falling edge of slave select signals or the first edge of the serial clock. The slave select line is held high when the SPI is idle or disabled. This SPI controller can work as either master or slave mode.

SPI Controller supports the following features:

- Support Motorola SPI,TI Synchronous Serial Protocol and National Semiconductor Micro wire interface
- Support 32-bit APB bus
- Support two internal 16-bit wide and 64-location deep FIFOs, one for transmitting and the other for receiving serial data
- Support two chip select signals in master mode
- Support 4,8,16 bit serial data transfer
- Support configurable interrupt polarity
- Support asynchronous APB bus and SPI clock
- Support master and slave mode
- Support DMA handshake interface and configurable DMA water level
- Support transmit FIFO empty, underflow, receive FIFO full, overflow, interrupt and all interrupts can be masked
- Support configurable water level of transmit FIFO empty and receive FIFO full interrupt
- Support 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 drive by SCLK_IN directly instead of spi_clk.

25.2 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

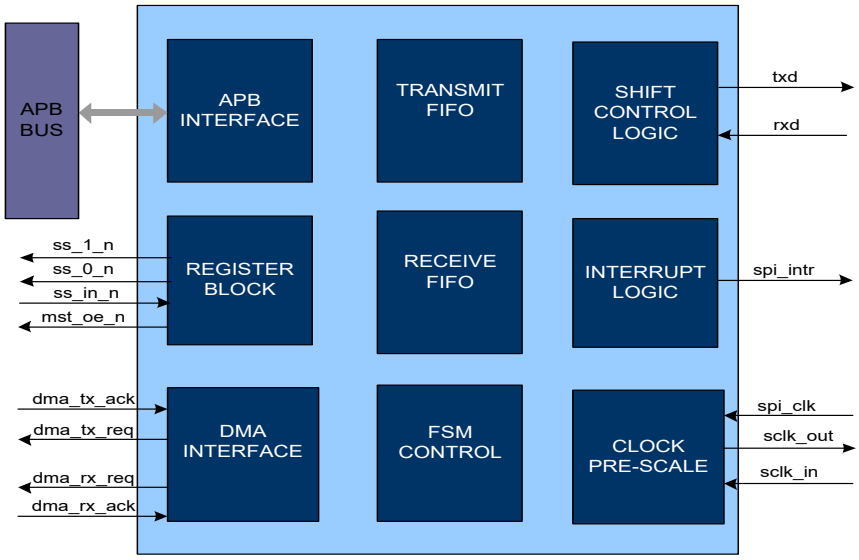


Fig. 25-1 SPI Controller Block Diagram

APB INTERFACE

The host processor accesses data, control, and status information on the SPI through the APB interface. The SPI supports APB data bus widths of 32 bits and 8 or 16 bits when reading or writing internal FIFO if data frame size(SPI_CTRL0[1:0]) is set to 8 bits.

DMA INTERFACE

This block has a handshaking interface to a DMA Controller to request and control transfers. The APB bus is used to perform the data transfer to or from the DMA Controller.

FIFO LOGIC

For transmit and receive transfers, data transmitted from the SPI to the external serial device is written into the transmit FIFO. Data received from the external serial device into the SPI is pushed into the receive FIFO. Both FIFOs are 64x16bits.

FSM CONTROL

Control the state's transformation of the design.

REGISTER BLOCK

All registers in the SPI are addressed at 32-bit boundaries to remain consistent with the APB bus. Where the physical size of any register is less than 32-bits wide, the upper unused bits of the 32-bit boundary are reserved. Writing to these bits has no effect; reading from these bits returns 0.

SHIFT CONTROL

Shift control logic shift the data from the transmit FIFO or to the receive FIFO. This logic automatically right-justifies receive data in the receive FIFO buffer.

INTERRUPT CONTROL

The SPI supports combined and individual interrupt requests, each of which can be masked. The combined interrupt request is the OR relationship between all other SPI interrupts after masking.

25.3 Function Description

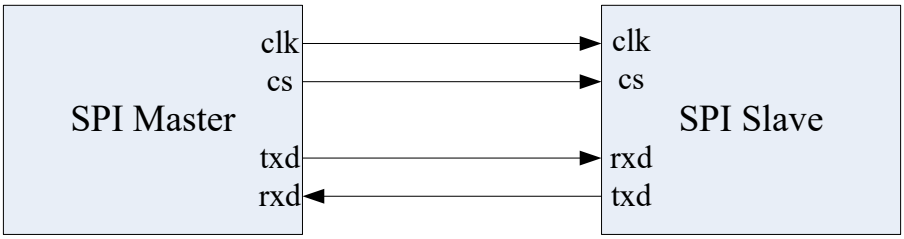


Fig. 25-2 SPI Master and Slave Interconnection

The SPI controller support dynamic switching between master and slave in a system. The diagram show how the SPI controller connects with other SPI devices.

Operation Modes

The SPI can be configured in the following two fundamental modes of operation: Master

Mode when SPI_CTRLR0 [20] is 1'b0, Slave Mode when SPI_CTRLR0 [20] is 1'b1.

Transfer Modes

The SPI operates in the following three modes when transferring data on the serial bus.

1). Transmit and Receive

When SPI_CTRLR0 [19:18] == 2'b00, both transmit and receive logic are valid.

2). Transmit Only

When SPI_CTRLR0 [19:18] == 2'b01, the receive data are invalid and should not be stored in the receive FIFO.

3). Receive Only

When SPI_CTRLR0 [19:18] == 2'b10, the transmit data are invalid.

Clock Ratios

A summary of the frequency ratio restrictions between the bit-rate clock (sclk_out/sclk_in) and the SPI peripheral clock (spi_clk) are described as,

When SPI Controller works as master, the F_{spi_clk}>= 2 × (maximum F_{sclk_out})

When SPI Controller works as slave, the F_{spi_clk}>= 6 × (maximum F_{sclk_in})

With the SPI, the clock polarity (SCPOL) configuration parameter determines whether the inactive state of the serial clock is high or low. To transmit data, both SPI peripherals must have identical serial clock phase (SCPH) and clock polarity (SCPOL) values. The data frame can be 4/8/16 bits in length.

When the configuration parameter SCPH = 0, data transmission begins on the falling edge of the slave select signal. The first data bit is captured by the master and slave peripherals on the first edge of the serial clock; therefore, valid data must be present on the txd and rxd lines prior to the first serial clock edge. The following two figures show a timing diagram for a single SPI data transfer with SCPH = 0. The serial clock is shown for configuration parameters SCPOL = 0 and SCPOL = 1.

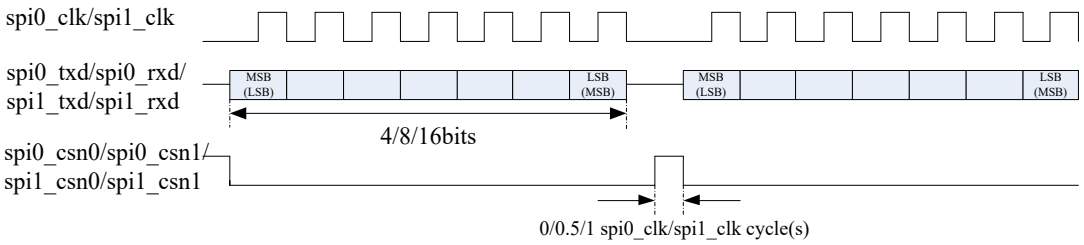


Fig. 25-3 SPI Format (SCPH=0 SCPOL=0)

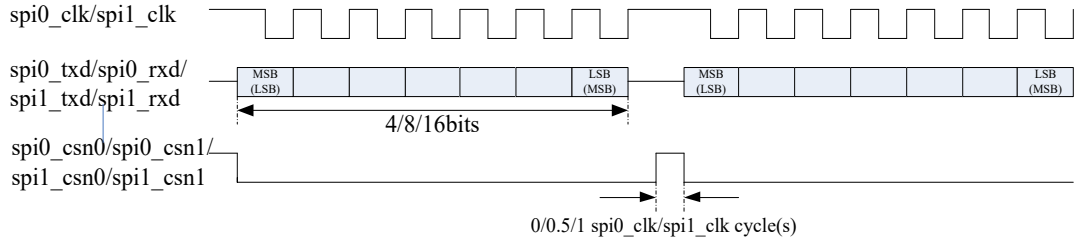


Fig. 25-4 SPI Format (SCPH=0 SCPOL=1)

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.

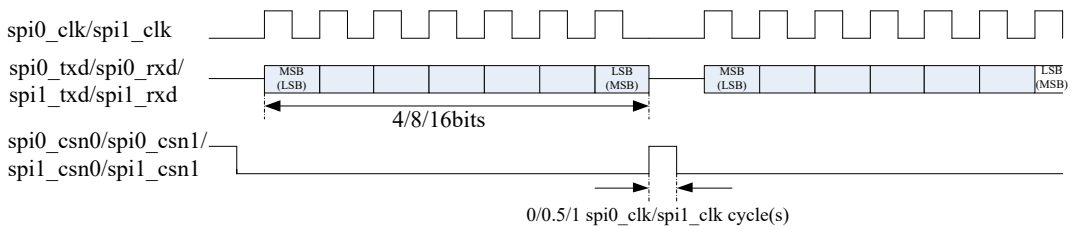


Fig. 25-5 SPI Format (SCPH=1 SCPOL=0)

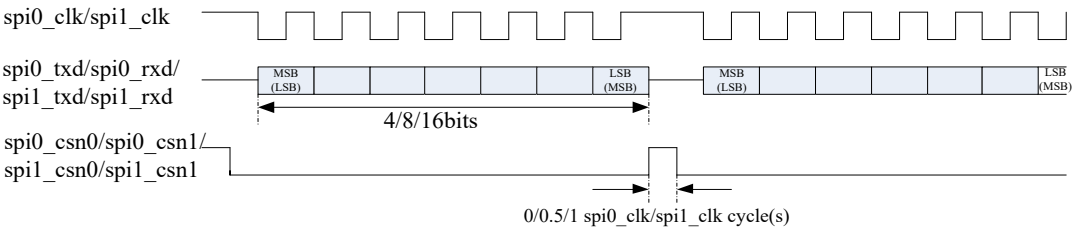


Fig. 25-6 SPI Format (SCPH=1 SCPOL=1)

25.4 Register Description

25.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
SPI_CTRLR0	0x0000	W	0x00000002	Control Register 0
SPI_CTRLR1	0x0004	W	0x00000000	Control Register 1
SPI_ENR	0x0008	W	0x00000000	SPI Enable Register
SPI_SER	0x000c	W	0x00000000	Slave Enable Register
SPI_BAUDR	0x0010	W	0x00000000	Baud Rate Select
SPI_TXFTLR	0x0014	W	0x00000000	Transmit FIFO Threshold Level
SPI_RXFTLR	0x0018	W	0x00000000	Receive FIFO Threshold Level
SPI_TXFLR	0x001c	W	0x00000000	Transmit FIFO Level
SPI_RXFLR	0x0020	W	0x00000000	Receive FIFO Level
SPI_SR	0x0024	W	0x0000004c	SPI Status
SPI_IPR	0x0028	W	0x00000000	Interrupt Polarity
SPI_IMR	0x002c	W	0x00000000	Interrupt Mask
SPI_ISR	0x0030	W	0x00000000	Interrupt Status
SPI_RISR	0x0034	W	0x00000001	Raw Interrupt Status
SPI_ICR	0x0038	W	0x00000000	Interrupt Clear
SPI_DMACR	0x003c	W	0x00000000	DMA Control
SPI_DMATDLR	0x0040	W	0x00000000	DMA Transmit Data Level
SPI_DMARDLR	0x0044	W	0x00000000	DMA Receive Data Level
SPI_TIMEOUT	0x004c	W	0x00000000	Timeout control register
SPI_BYPASS	0x0050	W	0x00000000	BYPASS control register
SPI_TXDR	0x0400	W	0x00000000	Transmit FIFO Data
SPI_RXDR	0x0800	W	0x00000000	Receive FIFO Data

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

25.4.2 Detail Register Description

SPI_CTRLR0

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:26	RO	0x00	reserved
25	RW	0x0	lbk Loop back mode select. 1'b0: Normal mode. 1'b1: Loop back mode,rxd is connected to txd.
24:23	RW	0x0	soi SS_N output inverted. 1'b0: Corresponding bit of ss_in is not inverted. 1'b1: Corresponding bit of ss_in is inverted.
22	RO	0x0	reserved
21	RW	0x0	mtm Valid when frame format is set to National Semiconductors Microwire. 1'b0: Non-sequential transfer 1'b1: Sequential transfer
20	RW	0x0	opm Master and slave mode select. 1'b0: Master Mode 1'b1: Slave Mode
19:18	RW	0x0	xfm Transmit and receive mode select. 2'b00 : Transmit & Receive 2'b01 : Transmit Only 2'b10 : Receive Only 2'b11 : Reserved
17:16	RW	0x0	frf 2'b00: Motorola SPI 2'b01: Texas Instruments SSP 2'b10: National Semiconductors Microwire 2'b11: Reserved
15:14	RW	0x0	rsd When SPI is configured as a master, if the rxd data cannot be sampled by the sclk_out edge at the right time, this register should be configured to define the number of the spi_clk cycles after the active sclk_out edge to sample rxd data later when SPI works at high frequency. 2'b00: Do not delay 2'b01: 1 cycle delay 2'b10: 2 cycles delay 2'b11: 3 cycles delay
13	RW	0x0	bht Valid when data frame size is 8bit. 1'b0: APB 16bit write/read, spi 8bit write/read. 1'b1: APB 8bit write/read, spi 8bit write/read.

Bit	Attr	Reset Value	Description
12	RW	0x0	fbm 1'b0: First bit is MSB. 1'b1: First bit is LSB.
11	RW	0x0	em Serial endian mode can be configured by this bit. APB endian mode is always little endian. 1'b0: Little endian 1'b1: Big endian
10	RW	0x0	ssd Valid when the frame format is set to Motorola SPI and SPI used as a master. 1'b0: The period between ss_n active and sclk_out active is half sclk_out cycles. 1'b1: The period between ss_n active and sclk_out active is one sclk_out cycle.
9:8	RW	0x0	csm Valid when the frame format is set to Motorola SPI and SPI used as a master. 2'b00: ss_n keep low after every frame data is transferred. 2'b01: ss_n be high for half sclk_out cycles after every frame data is transferred. 2'b10: ss_n be high for one sclk_out cycle after every frame data is transferred. 2'b11: Reserved
7	RW	0x0	scpol Valid when the frame format is set to Motorola SPI. 1'b0: Inactive state of serial clock is low. 1'b1: Inactive state of serial clock is high.
6	RW	0x0	scph Valid when the frame format is set to Motorola SPI. 1'b0: Serial clock toggles in middle of first data bit. 1'b1: Serial clock toggles at start of first data bit.

Bit	Attr	Reset Value	Description
5:2	RW	0x0	cfs Selects the length of the control word for the Microwire frame format. 4'b0000~0010: Reserved 4'b0011: 4-bit serial data transfer 4'b0100: 5-bit serial data transfer 4'b0101: 6-bit serial data transfer 4'b0110: 7-bit serial data transfer 4'b0111: 8-bit serial data transfer 4'b1000: 9-bit serial data transfer 4'b1001: 10-bit serial data transfer 4'b1010: 11-bit serial data transfer 4'b1011: 12-bit serial data transfer 4'b1100: 13-bit serial data transfer 4'b1101: 14-bit serial data transfer 4'b1110: 15-bit serial data transfer 4'b1111: 16-bit serial data transfer
1:0	RW	0x2	dfs Selects the data frame length. 2'b00: 4bit data 2'b01: 8bit data 2'b10: 16bit data 2'b11: Reserved

SPI_CTRLR1

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	ndm When Transfer Mode is receive only, this register field sets the number of data frames to be continuously received by the SPI. The SPI continues to receive serial data until the number of data frames received is equal to this register value plus 1, which enables you to receive up to 4GB of data in a continuous transfer.

SPI_ENR

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RW	0x0	enr Enables and disables all SPI operations. Transmit and receive FIFO buffers are cleared when the device is disabled.

SPI_SER

Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved
1:0	RW	0x0	ser Slave enable register.The register enable the individual slave select output lines, 2 slave-select output pins are available.This register is valid only when SPI is configured as a master device.

SPI_BAUDR

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15:0	RW	0x0000	baudr SPI Clock Divider. This register is valid only when the SPI is configured as a master device.The LSB for this field is always set to 0 and is unaffected by a write operation, which ensures an even value is held in this register. If the value is 0, the serial output clock (sclk_out) is disabled. The frequency of the sclk_out is derived from the following equation: Fsclk_out = Fspi_clk/ SCKDV Where SCKDV is any even value between 2 and 65534. For example: for Fspi_clk = 3.6864MHz and SCKDV =2 Fsclk_out = 3.6864/2= 1.8432MHz

SPI_TXFTLR

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:6	RO	0x0000000	reserved
5:0	RW	0x00	xftlr When the number of transmit FIFO entries is less than or equal to this value, the transmit FIFO empty interrupt is triggered.

SPI_RXFTLR

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:6	RO	0x0000000	reserved
5:0	RW	0x00	rxftlr When the number of receive FIFO entries is greater than or equal to this value + 1, the receive FIFO full interrupt is triggered.

SPI_TXFLR

Address: Operational Base + offset (0x001c)

Bit	Attr	Reset Value	Description
31:7	RO	0x0000000	reserved

Bit	Attr	Reset Value	Description
6:0	RO	0x00	txflr Contains the number of valid data entries in the transmit FIFO.

SPI_RXFLR

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:7	RO	0x0000000	reserved
6:0	RO	0x00	rxflr Contains the number of valid data entries in the receive FIFO.

SPI_SR

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:7	RO	0x0000000	reserved
6	RO	0x1	ssi 1'b0: ss_in_n is low. 1'b1: ss_in_n is high.
5	RO	0x0	stb 1'b0: Slave tx not busy. 1'b1: Slave tx busy.
4	RO	0x0	rff 1'b0: Receive FIFO is not full. 1'b1: Receive FIFO is full.
3	RO	0x1	rfe 1'b0: Receive FIFO is not empty. 1'b1: Receive FIFO is empty.
2	RO	0x1	tfe 1'b0: Transmit FIFO is not empty. 1'b1: Transmit FIFO is empty.
1	RO	0x0	tff 1'b0: Transmit FIFO is not full. 1'b1: Transmit FIFO is full.
0	RO	0x0	bsf When set, indicates that a serial transfer is in progress; when cleared, indicates that the SPI is idle or disabled. 1'b0: SPI is idle or disabled. 1'b1: SPI is actively transferring data.

SPI_IPR

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	ipr Interrupt Polarity Register. 1'b0: Active Interrupt Polarity Level is HIGH. 1'b1: Active Interrupt Polarity Level is LOW.

SPI_IMR

Address: Operational Base + offset (0x002c)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7	RW	0x0	txfim 1'b0: TX finish interrupt is masked. 1'b1: TX finish interrupt is not masked.
6	RW	0x0	sspim 1'b0: ss_in_n posedege interrupt is masked. 1'b1: ss_in_n posedege interrupt is not masked.
5	RW	0x0	toim 1'b0: spi timeout interrupt is masked. 1'b1: spi timeout interrupt is not masked.
4	RW	0x0	rffim 1'b0: spi_rxf_intr interrupt is masked. 1'b1: spi_rxf_intr interrupt is not masked.
3	RW	0x0	rfoim 1'b0: spi_rxo_intr interrupt is masked. 1'b1: spi_rxo_intr interrupt is not masked.
2	RW	0x0	rfuim 1'b0: spi_rxu_intr interrupt is masked. 1'b1: spi_rxu_intr interrupt is not masked.
1	RW	0x0	tfoim 1'b0: spi_txo_intr interrupt is masked. 1'b1: spi_txo_intr interrupt is not masked.
0	RW	0x0	tfeim 1'b0: spi_txe_intr interrupt is masked. 1'b1: spi_txe_intr interrupt is not masked.

SPI_ISR

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7	RW	0x0	txfis 1'b0: TX finish interrupt is not active after masking. 1'b1: TX finish interrupt is active after masking.
6	RW	0x0	sspis 1'b0: ss_in_n posedege interrupt is not active after masking. 1'b1: ss_in_n posedege interrupt is active after masking.

Bit	Attr	Reset Value	Description
5	RW	0x0	tois 1'b0: spi timeout interrupt is not active after masking. 1'b1: spi timeout interrupt is active after masking.
4	RO	0x0	rffis 1'b0: spi_rxf_intr interrupt is not active after masking. 1'b1: spi_rxf_intr interrupt is full after masking.
3	RO	0x0	rfois 1'b0: spi_rxo_intr interrupt is not active after masking. 1'b1: spi_rxo_intr interrupt is active after masking.
2	RO	0x0	rfuis 1'b0: spi_rxu_intr interrupt is not active after masking. 1'b1: spi_rxu_intr interrupt is active after masking.
1	RO	0x0	tfois 1'b0: spi_txo_intr interrupt is not active after masking. 1'b1: spi_txo_intr interrupt is active after masking.
0	RO	0x0	tfeis 1'b0: spi_txe_intr interrupt is not active after masking. 1'b1: spi_txe_intr interrupt is active after masking.

SPI_RISR

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7	RW	0x0	txfris 1'b0: TX finish interrupt is not active prior to masking. 1'b1: TX finish interrupt is active prior to masking.
6	RW	0x0	sspris 1'b0: ss_in_n posedege interrupt is not active prior to masking. 1'b1: ss_in_n posedege interrupt is active prior to masking.
5	RW	0x0	toris 1'b0: spi_timeout interrupt is not active prior to masking. 1'b1: spi_timeout interrupt is active prior to masking.
4	RO	0x0	rffris 1'b0: spi_rxf_intr interrupt is not active prior to masking. 1'b1: spi_rxf_intr interrupt is full prior to masking.
3	RO	0x0	rforis 1'b0: spi_rxo_intr interrupt is not active prior to masking. 1'b1: spi_rxo_intr interrupt is active prior to masking.
2	RO	0x0	rfuris 1'b0: spi_rxu_intr interrupt is not active prior to masking. 1'b1: spi_rxu_intr interrupt is active prior to masking.
1	RO	0x0	tforis 1'b0: spi_txo_intr interrupt is not active prior to masking. 1'b1: spi_txo_intr interrupt is active prior to masking.

Bit	Attr	Reset Value	Description
0	RO	0x1	tferis 1'b0: spi_txe_intr interrupt is not active prior to masking. 1'b1: spi_txe_intr interrupt is active prior to masking.

SPI_ICR

Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:7	RO	0x0000000	reserved
6	WO	0x0	ctxfi Write 1 to Clear tx finish Interrupt.
5	WO	0x0	csspi Write 1 to Clear ss_in_n posdege Interrupt.
4	WO	0x0	ctoi Write 1 to Clear Timeout Interrupt.
3	WO	0x0	ctfoi Write 1 to Clear Transmit FIFO Overflow Interrupt.
2	WO	0x0	crfoi Write 1 to Clear Receive FIFO Overflow Interrupt.
1	WO	0x0	crfui Write 1 to Clear Receive FIFO Underflow Interrupt.
0	WO	0x0	cci Write 1 to Clear Combined Interrupt.

SPI_DMACR

Address: Operational Base + offset (0x003c)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved
1	RW	0x0	tde 1'b0: Transmit DMA disabled. 1'b1: Transmit DMA enabled.
0	RW	0x0	rde 1'b0: Receive DMA disabled. 1'b1: Receive DMA enabled.

SPI_DMATDLR

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:6	RO	0x0000000	reserved
5:0	RW	0x00	tdl This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of valid data entries in the transmit FIFO is equal to or below this field value, and transmit DMA is enabled (DMACR[1] = 1).

SPI_DMARDLR

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:6	RO	0x0000000	reserved
5:0	RW	0x00	rdl This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or above this field value + 1, and receive DMA is enabled(DMACR[0]=1).

SPI_TIMEOUT

Address: Operational Base + offset (0x004c)

Bit	Attr	Reset Value	Description
31:17	RO	0x0000	reserved
16	RW	0x0	toe Timeout enable. 1'b0: Timeout counter is inactive. 1'b1: Timeout counter will be active after the first rising edge of sclk_in.
15:0	RW	0x0000	tov Timeout threshold value. If sclk_in keep inactive for a threshold time , timeout interrupt will be triggered .The timeout threshold time is TOV*pclk_perid*16.

SPI_BYPASS

Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:5	RO	0x0000000	reserved
4	RW	0x0	txcp TX clock polarity.This bit is only valid in bypass mode. 1'b0: TX logic use raw SCLK. 1'b1: TX logic use inverted SCLK.
3	RW	0x0	rxcp RX clock polarity.This bit is only valid in bypass mode. 1'b0: RX logic use raw SCLK. 1'b1: RX logic use inverted SCLK.
2	RW	0x0	end Endian mode.This bit is only valid in bypass mode. 1'b0: Work in littel endian mode. 1'b1: Work in big endian mode.
1	RW	0x0	fbm First bit mode.This bit is only valid in bypass mode. 1'b0: First bit is LSB. 1'b1: First bit is MSB.

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	0x0000	reserved
15:0	WO	0x0000	txdr When it is written to, data are moved into the transmit FIFO.

SPI_RXDR

Address: Operational Base + offset (0x0800)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15:0	RO	0x0000	rxdr When the register is read, data in the receive FIFO is accessed.

25.5 Interface Description

Table 25-1 SPI interface description

Module Pin	Direction	Pad Name	IOMUX Setting
SPI0mux0 Interface			
spi0_sclk	I/O	GPIO0_B0/SPI0_CLK_M0	PMUGRF_GPIO0B_IOMUX_SEL_L[3:0]=4'h1
spi0_rxd	I	GPIO0_A7/SPI0_MISO_M0	PMUGRF_GPIO0A_IOMUX_SEL_H[15:12]=4'h1
spi0_txd	O	GPIO0_A6/SPI0_MOSI_M0	PMUGRF_GPIO0A_IOMUX_SEL_H[11:8]=4'h1
spi0_csn0	I/O	GPIO0_A5/SPI0_CS0N_M0	PMUGRF_GPIO0A_IOMUX_SEL_H[7:4]=4'h1
spi0_csn1	O	GPIO0_A4/SPI0_CS1N_M0	PMUGRF_GPIO0A_IOMUX_SEL_H[3:0]=4'h1
SPI0mux1 Interface			
spi0_sclk	I/O	GPIO2_A1/SPI0_CLK_M1/I2S1_SDO_M1/UART5_RX_M2	GRF_GPIO2A_IOMUX_SEL_L[7:4]=4'h1
spi0_rxd	I	GPIO1_D7/SPI0_MISO_M1/I2S1_LRCK_M1/I2C3_SDA_M2	GRF_GPIO1D_IOMUX_SEL_H[15:12]=4'h1
spi0_txd	O	GPIO1_D6/SPI0_MOSI_M1/I2S1_SCLK_M1/I2C3_SCL_M2	GRF_GPIO1D_IOMUX_SEL_H[11:8]=4'h1
spi0_csn0	I/O	GPIO2_A0/SPI0_CS0n_M1/I2S1_SDI_M1/UART5_TX_M2	GRF_GPIO2A_IOMUX_SEL_L[3:0]=4'h1
spi0_csn1	O	GPIO1_D5/SPI0_CS1n_M1/I2S1_MCLK_M1/UART4_TX_M2	GRF_GPIO1D_IOMUX_SEL_H[7:4]=4'h1
SPI0mux2 Interface			
spi0_sclk	I/O	GPIO2_B2/LCDC_D6/I2S2_LRCK_M1/UART5_RTSN_M1/PWM1_M1/SPI0_CLK_M2	GRF_GPIO2B_IOMUX_SEL_L[11:8]=4'h6
spi0_rxd	I	GPIO2_B1/LCDC_D5/I2S2_SCLK_M1/UART5_RX_M1/PWM2_M1/SPI0_MISO_M2	GRF_GPIO2B_IOMUX_SEL_L[7:4]=4'h6
spi0_txd	O	GPIO2_B0/LCDC_D4/I2S2_SDI_M1/UART5_TX_M1/PWM3_IR_M1/SPI0_MOSI_M2	GRF_GPIO2B_IOMUX_SEL_L[3:0]=4'h6

Module Pin	Direction	Pad Name	IOMUX Setting
spi0_csn0	I/O	GPIO2_A7/LCDC_D3/I2S2_SDO_M1/UART4_RX_M1/PWM4_M1/SPI0_CS0N_M2	GRF_GPIO2A_IOMUX_SEL_H[15:12]=4'h6
spi0_csn1	O	GPIO2_B3/LCDC_D7/I2S2_MCLK_M1/CIF_D3_M1/UART5_CTSN_M1/PWM0_M1/SPI0_CS1N_M2/I2C5_SDA_M0	GRF_GPIO2B_IOMUX_SEL_L[15:12]=4'h6
SPI1mux0 Interface			
spi1_sclk	I/O	GPIO3_C0/CIF_D12_M0/RGMII_CLK_M0/PDM_CLK0_M1/SPI1_CLK_M0	GRF_GPIO3C_IOMUX_SEL_L[3:0]=4'h5
spi1_rxd	I	GPIO3_B7/CIF_D11_M0/RGMII_RXD1_M0/PDM_SDI3_M1/SPI1_MISO_M0	GRF_GPIO3B_IOMUX_SEL_H[15:12]=4'h5
spi1_txd	O	GPIO3_B6/CIF_D10_M0/RGMII_RXD0_M0/PDM_SDI2_M1/SPI1_MOSI_M0	GRF_GPIO3B_IOMUX_SEL_H[11:8]=4'h5
spi1_csn0	I/O	GPIO3_B5/CIF_D9_M0/RGMII_TXEN_M0/I2S0_SDO3_SDI1_M1/SPI1_CS0N_M0	GRF_GPIO3B_IOMUX_SEL_H[7:4]=4'h5
spi1_csn1	O	GPIO3_B4/CIF_D8_M0/RGMII_TXD1_M0/I2S0_SDO2_SDI2_M1/SPI1_CS1N_M0	GRF_GPIO3B_IOMUX_SEL_H[3:0]=4'h5
SPI1mux1 Interface			
spi1_sclk	I/O	GPIO1_C6/I2S2_SCLK_M0/SPI1_CLK_M1/PRELIGHT_TRIG_OUT/UART1_RTSN_M1	GRF_GPIO1C_IOMUX_SEL_H[11:8]=4'h3
spi1_rxd	I	GPIO1_C5/I2S2_SDI_M0/SPI1_MISO_M1/FLASH_TRIG_IN	GRF_GPIO1C_IOMUX_SEL_H[7:4]=4'h3
spi1_txd	O	GPIO1_C4/I2S2_SDO_M0/SPI1_MOSI_M1/FLASH_TRIG_OUT	GRF_GPIO1C_IOMUX_SEL_H[3:0]=4'h3
spi1_csn0	I/O	GPIO1_C7/I2S2_LRCK_M0/SPI1_CS0N_M1/UART1_CTSN_M1	GRF_GPIO1C_IOMUX_SEL_H[15:12]=4'h3
spi1_csn1	O	GPIO1_D0/I2S2_MCLK_M0/SDMMC1_DET/SPI1_CS1N_M1/I2C5_SCL_M2/UART1_TX_M1	GRF_GPIO1D_IOMUX_SEL_L[3:0]=4'h3
SPI1mux2 Interface			
spi1_sclk	I/O	GPIO2_D5/LCDC_HSYNC/PWM10_M1/SPI1_CLK_M2/I2C3_SDA_M1	GRF_GPIO2D_IOMUX_SEL_H[7:4]=4'h6
spi1_rxd	I	GPIO2_D7/LCDC_CLK/UART3_CTSN_M2/PWM8_M1/SPI1_MISO_M2	GRF_GPIO2D_IOMUX_SEL_H[15:12]=4'h6
spi1_txd	O	GPIO2_D6/LCDC_VSYNC/UART3_RTSN_M2/PWM9_M1/SPI1_MOSI_M2	GRF_GPIO2D_IOMUX_SEL_H[11:8]=4'h6
spi1_csn0	I/O	GPIO2_D4/LCDC_DEN/PWM6_M1/SPI1_CS0N_M2/I2C3_SCL_M1	GRF_GPIO2D_IOMUX_SEL_H[3:0]=4'h6
spi1_csn1	O	GPIO3_A0/CAN_RXD_M0/UART3_TX_M2/PWM7_IR_M1/SPI1_CS1N_M2/I2C4_SCL_M0	GRF_GPIO3A_IOMUX_SEL_L[3:0]=4'h6

Notes: I=input, O=output, I/O=input/output, bidirectional. spi_csn1 can only be used in master mode

25.6 Application Notes

Clock Ratios

A summary of the frequency ratio restrictions between the bit-rate clock (sclk_out/sclk_in) and the SPI peripheral clock (spi_clk) are described as,

When SPI Controller works as master, the Fspi_clk>= 2 × (maximum Fsclk_out)

When SPI Controller works as slave, the Fspi_clk>= 6 × (maximum Fsclk_in)

Master Transfer Flow

When configured as a serial-master device, the SPI initiates and controls all serial transfers. The serial bit-rate clock, generated and controlled by the SPI, is driven out on the sclk_out line. When the SPI is disabled (SPI_ENR = 0), no serial transfers can occur and sclk_out is held in "inactive" state, as defined by the serial protocol under which it operates.

Slave Transfer Flow

When the SPI is configured as a slave device, all serial transfers are initiated and controlled by the serial bus master.

When the SPI serial slave is selected during configuration, it enables its txd data onto the serial bus. All data transfers to and from the serial slave are regulated on the serial clock line (sclk_in), driven from the serial-master device. Data are propagated from the serial slave on one edge of the serial clock line and sampled on the opposite edge.

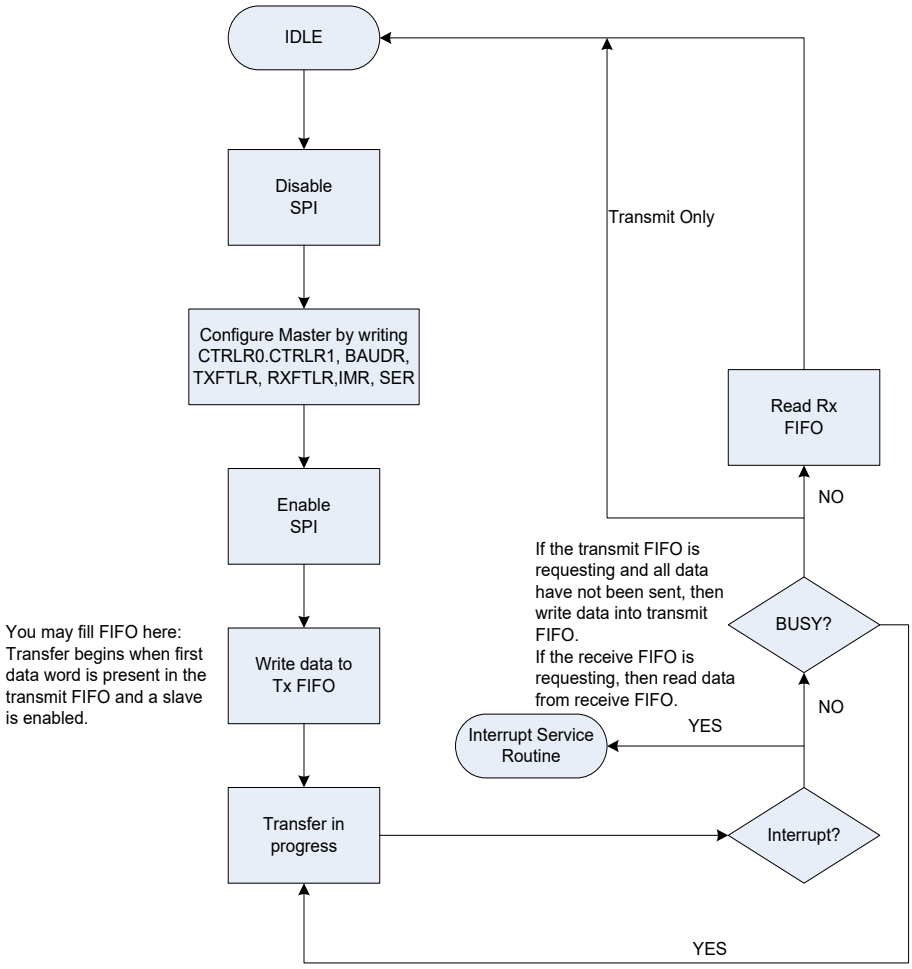


Fig. 25-7 SPI Master transfer flow diagram

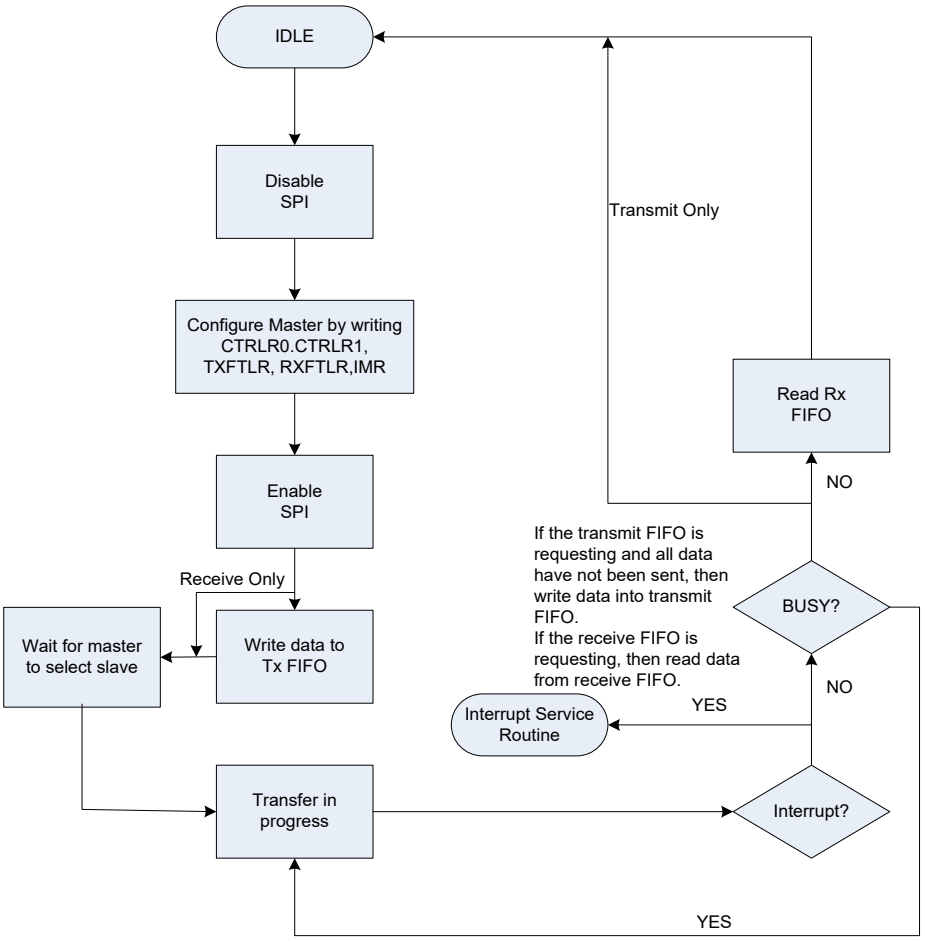


Fig. 25-8 SPI Slave transfer flow diagram

Chapter 26 Flexible Serial Peripheral Interface (FSPI)

26.1 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
- Support 2 CS# operation

26.2 Block Diagram

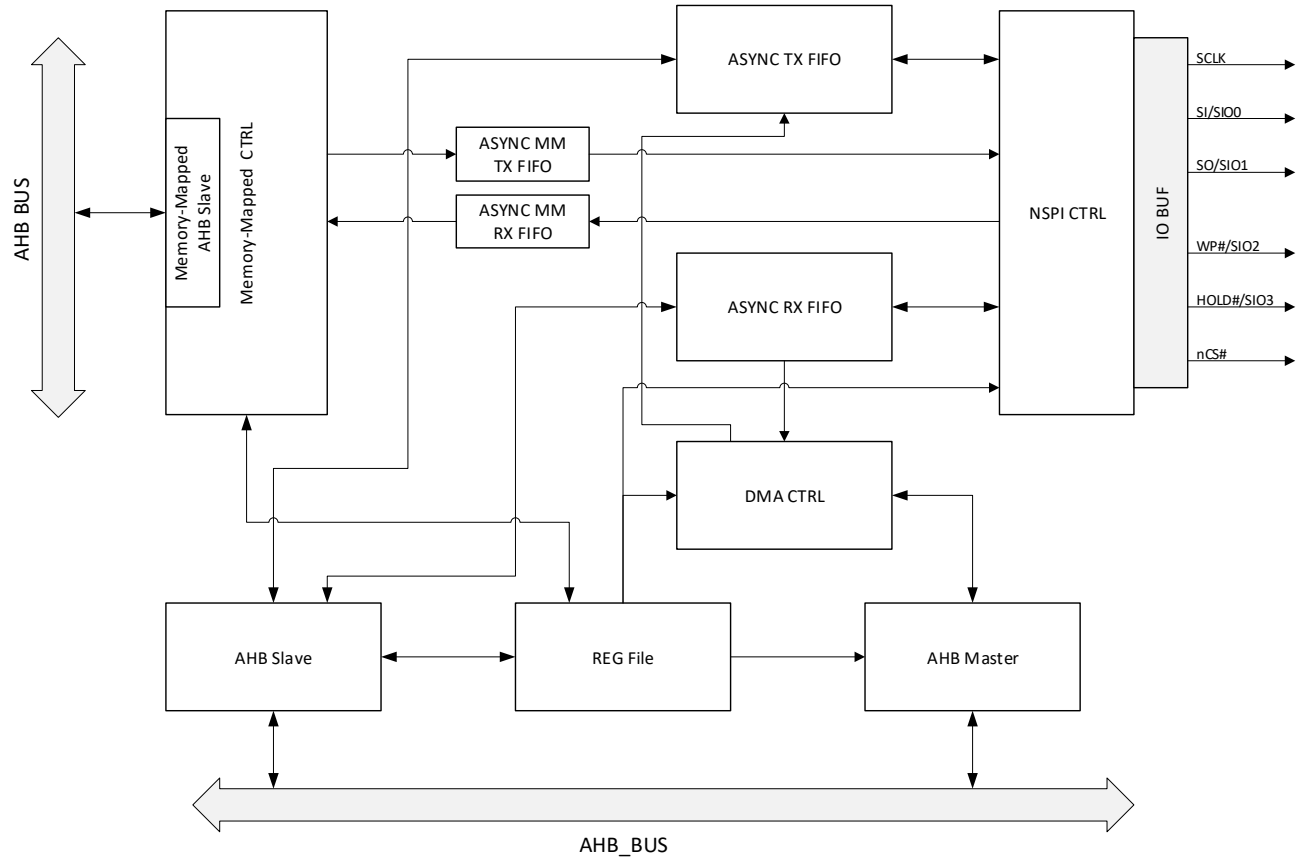


Fig. 26-1 FSPI Architecture

26.3 Function Description

26.3.1 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.

26.3.2 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.

26.3.3 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.

26.3.4 REG File

The REG File is configurable register bank to control the store the static configuration and dynamic status of controller.

26.3.5 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.

26.3.6 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.

26.3.7 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.

26.4 Register Description

26.4.1 Internal Address Mapping

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

Table 26-1 FSPI Address Mapping Table

Name	Address Base	Size
FSPI CFG	0xFFC90000	64KB
FSPI MEM0	0xFD000000	16MB
FSPI MEM1	0xFFCA0000	64KB

The address of FSPI MEM0 is accessible only when the INTCONNET_CON0[3] is set to “1”.

26.4.2 Registers Summary

Name	Offset	Size	Reset Value	Description
FSPI_CTRL0	0x0000	W	0x00000000	Control Register for CS0 Device
FSPI_IMR	0x0004	W	0x000001FF	Interrupt Mask Register
FSPI_ICLR	0x0008	W	0x00000000	Interrupt Clear Register
FSPI_FTLR	0x000C	W	0x00001010	FIFO Threshold Level Register
FSPI_RCVR	0x0010	W	0x00000000	FSPI Recover Register

Name	Offset	Size	Reset Value	Description
FSPI_AX0	0x0014	W	0x00000000	FSPI Auxiliary Data Value for CS0 Device
FSPI_ABIT0	0x0018	W	0x00000000	Extend Address Bits for CS0 Device
FSPI_ISR	0x001C	W	0x00000000	Interrupt Status
FSPI_FSR	0x0020	W	0x00000001	FIFO Status Register
FSPI_SR	0x0024	W	0x00000000	FSPI Status Register
FSPI_RISR	0x0028	W	0x00000000	Raw Interrupt Status Register
FSPI_VER	0x002C	W	0x00000004	Version Register
FSPI_QOP	0x0030	W	0x00000000	Quad Line Operation IO Level Pre-set Register
FSPI_EXT_CTRL	0x0034	W	0x00004023	Extend Control Register
FSPI_DLL_CTRL0	0x003C	W	0x00000001	Delay Line Control Register for CS0 Device
FSPI_EXT_AX	0x0044	W	0x0000F0FF	Extend Auxiliary Data Control Register
FSPI_SCLK_INATM_CNT	0x0048	W	0xFFFFFFFF	SCLK Inactive Timeout Counter
FSPI_XMMC_WCMD0	0x0050	W	0x00000000	Memory Mapped Control Write Command Register for CS0 Device
FSPI_XMMC_RCMD0	0x0054	W	0x00000000	Memory Mapped Control Read Command Register for CS0 Device
FSPI_XMMC_CTRL	0x0058	W	0x000072E0	Memory Mapped Control Register
FSPI_MODE	0x005C	W	0x00000000	Controller Working Mode Register
FSPI_DEVRGN	0x0060	W	0x00000017	Device Region Size Register
FSPI_DEVSIZE0	0x0064	W	0x00000012	Device Size Register for CS0 Device
FSPI_TME0	0x0068	W	0x00000000	Timeout Enable Control Register for CS0 Device
FSPI_XMMC_RX_WTMRK	0x0070	W	0x00000002	Memory Mapped Mode Receiver FIFO Water Mark Register
FSPI_DMATR	0x0080	W	0x00000000	DMA Trigger Register
FSPI_DMAADDR	0x0084	W	0x00000000	DMA Address Register
FSPI_LEN_CTRL	0x0088	W	0x00000000	Length Control Register
FSPI_LEN_EXT	0x008C	W	0x00000000	Length Extended Register
FSPI_XMMCSR	0x0094	W	0x00000000	Memory Mapped Status Register
FSPI_CMD	0x0100	W	0x00000000	Indirect Command Register
FSPI_ADDR	0x0104	W	0x00000000	Address Register
FSPI_DATA	0x0108	W	0x00000000	Data Register
FSPI_CTRL1	0x0200	W	0x00000000	Control Register for CS1 Device
FSPI_AX1	0x0214	W	0x00000000	FSPI Auxiliary Data Value for CS1 Device
FSPI_ABIT1	0x0218	W	0x00000000	Extend Address Bits for CS1 Device
FSPI_DLL_CTRL1	0x023C	W	0x00000001	Delay Line Control Register for CS1 Device
FSPI_XMMC_WCMD1	0x0250	W	0x00000000	Memory Mapped Control Write Command Register for CS1 Device
FSPI_XMMC_RCMD1	0x0254	W	0x00000000	Memory-Mapped Command Control Register for CS1 Device
FSPI_DEVSIZE1	0x0264	W	0x00000012	Device Size Register for CS1 Device
FSPI_TME1	0x0268	W	0x00000000	Timeout Enable Control Register for CS1 Device

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

26.4.3 Detail Registers Description

FSPI_CTRL0

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:14	RO	0x00000	reserved
13:12	RW	0x0	DATB Data Line Width 2'b00: 1 bit, x1 mode 2'b01: 2 bits, x2 mode 2'b10: 4 bits, x4 mode 2'b11: Reserved Set this DATB to match the CMD sequence before doing indirect access mode and memory mapped access mode.
11:10	RW	0x0	ADRB Address Line Width 2'b00: 1 bit, x1 mode 2'b01: 2 bits, x2 mode 2'b10: 4 bits, x4 mode 2'b11: Reserved Set this ADRB to match the CMD sequence before doing indirect access mode and memory mapped access mode.
9:8	RW	0x0	CMDB Command Line Width 2'b00: 1 bit, x1 mode 2'b01: 2 bits, x2 mode 2'b10: 4 bits, x4 mode 2'b11: Reserved Set this CMDB to match the CMD sequence before doing indirect access mode and memory mapped access mode.
7:4	RW	0x0	IDLE_CYCLE Idle Cycles When Switching IO from Output to Input 4'h0: Idle hold is disable 4'h1: Hold the SCLK in idle for 2 cycles when switch to shift in ... 4'hf: Hold the SCLK in idle for 16 cycles when switch to shift in To improve the transform IO timing, the application can set this register to hold the SCLK in low state or high state.
3:2	RO	0x0	reserved
1	RW	0x0	SHIFTPHASE Shift Phase of Data Input in Controller 1'b0: Shift input data at posedge SCLK 1'b1: Shift input data at negedge SCLK The application can select the input data captured by posedge of SCLK when "0" or negedge of SCLK when "1".
0	RW	0x0	SPIM Serial Peripheral Interface Mode 1'b0: Mode 0 1'b1: Mode 3 SPIM is used to control the serial mode (CPOL and CPHA). CPOL indicates clock polarity of Serial master, CPOL=1 for SCLK high while idle, CPOL=0 for SCLK low while not transmitting. CPHA indicates clock phase. The combination of CPOL bit and CPHA bit decides which Serial mode is supported.

FSPI_IMR

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7	RW	0x1	DMAM DMA Finish Interrupt Mask 1'b0: DMA finish interrupt is not masked 1'b1: DMA finish interrupt is masked Only valid in indirect access mode.
6	RW	0x1	NSPIM NSPI Interrupt Mask 1'b0: NSPI interrupt is not masked 1'b1: NSPI interrupt is masked Valid in indirect access mode and memory mapped mode.
5	RW	0x1	AHBM AMBA AHB Error Interrupt Mask 1'b0: AMBA AHB Error interrupt is not masked 1'b1: AMBA AHB Error interrupt is masked Only valid in indirect access mode.
4	RW	0x1	TRANSM Transfer Finish Interrupt Mask 1'b0: Transfer finish interrupt is not masked 1'b1: Transfer finish interrupt is masked Only valid in indirect access mode.
3	RW	0x1	TXEM Transmit FIFO Empty Interrupt Mask 1'b0: Transmit FIFO empty interrupt is not masked 1'b1: Transmit FIFO empty interrupt is masked Only valid in indirect access mode.
2	RW	0x1	TXOM Transmit FIFO Overflow Interrupt Mask 1'b0: Transmit FIFO overflow interrupt is not masked 1'b1: Transmit FIFO overflow interrupt is masked Only valid in indirect access mode.
1	RW	0x1	RXUM Receive FIFO Underflow Interrupt Mask 1'b0: Receive FIFO underflow interrupt is not masked 1'b1: Receive FIFO underflow interrupt is masked Only valid in indirect access mode.
0	RW	0x1	RXFM Receive FIFO Full Interrupt Mask 1'b0: Receive FIFO full interrupt is not masked 1'b1: Receive FIFO full interrupt is masked Only valid in indirect access mode.

FSPI_ICLR

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7	W1C	0x0	DMAC DMA Finish Interrupt Clear 1'b0: No action 1'b1: Clear interrupt Write "1" to clear the DMAS

Bit	Attr	Reset Value	Description
6	W1C	0x0	NSPIC NSPI Error Interrupt Clear 1'b0: No action 1'b1: Clear interrupt Write "1" to clear the NSPIS.
5	W1C	0x0	AHBC AMBA AHB Error Interrupt Clear 1'b0: No action 1'b1: Clear interrupt Write "1" to clear the AHBS.
4	W1C	0x0	TRANSC Transfer Finish Interrupt Clear 1'b0: No action 1'b1: Clear interrupt Write "1" to clear the TRANSS.
3	W1C	0x0	TXEC Transmit FIFO Empty Interrupt Clear 1'b0: No action 1'b1: Clear interrupt Write "1" to clear the TXES.
2	W1C	0x0	TXOC Transmit FIFO Overflow Interrupt Clear 1'b0: No action 1'b1: Clear interrupt Write "1" to clear the TXOS.
1	W1C	0x0	RXUC Receive FIFO Underflow Interrupt Clear 1'b0: No action 1'b1: Clear interrupt Write "1" to clear the RXUS.
0	W1C	0x0	RXFC Receive FIFO Full Interrupt Clear 1'b0: No action 1'b1: Clear interrupt Write "1" to clear the RXFS.

FSPI_FTLR

Address: Operational Base + offset (0x000C)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15:8	RW	0x10	RXFTLR Receive FIFO Threshold Level 8'h0: 0 entry level 8'h1: 1 entry level ... 8'h10: 16 entry level(default) ... When the number of receive FIFO entries is bigger than or equal to this value, the receive FIFO full interrupt is triggered.

Bit	Attr	Reset Value	Description
7:0	RW	0x10	TXFTLR Transmit FIFO Threshold Level 8'h0: 0 entry level 8'h1: 1 entry level ... 8'h10: 16 entry level(default) ... When the number of transmit FIFO entries is less than or equal to this value, the transmit FIFO empty interrupt is triggered.

FSPI_RCVR

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	R/WSC	0x0	RCVR FSPI Recover Write any values to trigger the recovery of SMC NSPI state machine, FIFO state and other logic state.

FSPI_AX0

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x00	AX Auxiliary Data The AX value when doing the continuous read (enhance mode or XIP mode). That is M7-M0 in "Continuous Read Mode".

FSPI_ABIT0

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:5	RO	0x0000000	reserved
4:0	RW	0x00	ABIT Address Bits Extend 5'h0: 1 bit 5'h1: 2 bits ... 5'h1f: 32 bits Only valid when ADDRB is set to 2'b11.

FSPI_ISR

Address: Operational Base + offset (0x001C)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7	RO	0x0	DMAS DMA Finish Interrupt Status 1'b0: No interrupt 1'b1: Active interrupt generated
6	RO	0x0	NSPIS NSPI Transaction Decode Error Interrupt Status 1'b0: No interrupt 1'b1: Active interrupt generated

Bit	Attr	Reset Value	Description
5	RO	0x0	AHBS AMBA AHB Error Interrupt Status 1'b0: No interrupt 1'b1: Active interrupt generated
4	RO	0x0	TRANSS Transfer Finish Interrupt Status 1'b0: No interrupt 1'b1: Active interrupt generated
3	RO	0x0	TXES Transmit FIFO Empty Interrupt Status 1'b0: No interrupt 1'b1: Active interrupt generated
2	RO	0x0	TXOS Transmit FIFO Overflow Interrupt Status 1'b0: No interrupt 1'b1: Active interrupt generated
1	RO	0x0	RXUS Receive FIFO Underflow Interrupt Status 1'b0: No interrupt 1'b1: Active interrupt generated
0	RW	0x0	RXFS Receive FIFO Full Interrupt Status 1'b0: No interrupt 1'b1: Active interrupt generated

FSPI_FSR

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:21	RO	0x000	reserved
20:16	RW	0x00	RXWLVL RX FIFO Water Level 5'h0: FIFO is empty 5'h1: 1 entry is taken ... 5'h10: 16 entry is taken, FIFO is full
15:13	RO	0x0	reserved
12:8	RO	0x00	TXWLVL TX FIFO Water Level 5'h0: FIFO is full 5'h1: 1 entry is left ... 5'h10: 16 entry is left, FIFO is empty
7:4	RO	0x0	reserved
3	RO	0x0	RXFS Receive FIFO Full Status 1'b0: RX FIFO is not full 1'b1: RX FIFO is full
2	RO	0x0	RXES Receive FIFO Empty Status 1'b0: RX FIFO is not empty 1'b1: RX FIFO is empty
1	RO	0x0	TXES Transmit FIFO Empty Status 1'b0: TX FIFO is not empty 1'b1: TX FIFO is empty

Bit	Attr	Reset Value	Description
0	RO	0x1	TXFS Transmit FIFO Full Status 1'b0: TX FIFO is not full 1'b1: TX FIFO is full

FSPI_SR

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RO	0x0	SR Status Register 1'b0: NSPI Controller is idle 1'b1: NSPI Controller is busy When controller is busy, don't change the setting of control register. When NSPI is idle, the software can initiate new transaction to external device.

FSPI_RISR

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7	RO	0x0	DMAS DMA Finish Interrupt Status 1'b0: No active raw interrupt 1'b1: Active raw interrupt is generated Cleared by writing corresponding ICLR bit to clear raw interrupt status.
6	RO	0x0	NSPIS NSPI Error Interrupt Status 1'b0: No active raw interrupt 1'b1: Active raw interrupt is generated Cleared by writing corresponding ICLR bit to clear raw interrupt status.
5	RO	0x0	AHBS AMBA AHB Error Interrupt Status 1'b0: No active raw interrupt 1'b1: Active raw interrupt is generated Cleared by writing corresponding ICLR bit to clear raw interrupt status.
4	RO	0x0	TRANSS Transfer Finish Interrupt Status 1'b0: No active raw interrupt 1'b1: Active raw interrupt is generated Cleared by writing corresponding ICLR bit to clear raw interrupt status.
3	RO	0x0	TXES Transmit FIFO Empty Interrupt Status 1'b0: No active raw interrupt 1'b1: Active raw interrupt is generated Cleared by writing corresponding ICLR bit to clear raw interrupt status.

Bit	Attr	Reset Value	Description
2	RO	0x0	TXOS Transmit FIFO Overflow Interrupt Status 1'b0: No active raw interrupt 1'b1: Active raw interrupt is generated Cleared by writing corresponding ICLR bit to clear raw interrupt status.
1	RO	0x0	RXUS Receive FIFO Underflow Interrupt Status 1'b0: No active raw interrupt 1'b1: Active raw interrupt is generated Cleared by writing corresponding ICLR bit to clear raw interrupt status.
0	RO	0x0	RXFS Receive FIFO Full Interrupt Status 1'b0: No active raw interrupt 1'b1: Active raw interrupt is generated Cleared by writing corresponding ICLR bit to clear raw interrupt status.

FSPI_VER

Address: Operational Base + offset (0x002C)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15:0	RO	0x0004	VER The Version ID of FSPI

FSPI_QOP

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved
1	RW	0x0	SO123BP SO123 Bypass Mode 1'b0: Disable bypass 1'b1: Enable bypass Default is enabled.
0	RW	0x0	SO123 D1/D2/D3 Data Value During Inactive When CS is Active 1'b0: Set to "0" 1'b1: Set to "1" The value of SO1, SO2 and SO3 during command and address bits input.

FSPI_EXT_CTRL

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:15	RO	0x00000	reserved
14	RW	0x1	SR_GEN_MODE Status Register Generation Mode 1'b0: Compatible mode with old controller 1'b1: Robust generation to indicates the status of controller If set to "1", the controller will only clear the SR bit after operation sequence done and CS is high.

Bit	Attr	Reset Value	Description
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.
3:0	RW	0x3	CS_DESEL_CTRL CS Inactive Control 4'h0: 1 cycle 4'h1: 2 cycles 4'h2: 3 cycles 4'h3: 4 cycles ... 4'hf: 16 cycles The target cycles to hold CS inactive after de-assert the CS. Default value are 4 SCLK cycles that is enough for normal device.

FSPI_DLL_CTRL0

Address: Operational Base + offset (0x003C)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15	RW	0x0	SCLK_SMP_SEL SCLK Sampling Selection 1'b0: Bypass DLL 1'b1: From DLL The sampling SCLK source selection.
14:9	RO	0x00	reserved

Bit	Attr	Reset Value	Description
8:0	RW	0x001	SMP_DLL_CFG Sample Delay Line Configuration 9'h0: 1 DLL element cell 9'h1: 1 DLL element cell 9'h2: 2 DLL element cells ... 9'h1ff: 511 DLL element cells This register to control the sampling delay line cell used. The maximum DLL element cells is decided by process.

FSPI_EXT_AX

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15:8	RW	0xf0	AX_SETUP_PAT Auxiliary Setup Data Pattern The AX data pattern that setup the continuous/enhance/XIP read mode
7:0	RW	0xff	AX_CANCEL_PAT Auxiliary Cancel Data Pattern The AX data pattern that cancel the continuous/enhance/XIP read mode.

FSPI_SCLK_INATM_CNT

Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31:0	RW	0xffffffff	SCLK_INATM_CNT SCLK Inactive Timeout Counter When CS is active and SCLK is hold in high or low due to TX FIFO is empty or RX FIFO is full, if SCLK_INATM_EN is enabled, and timeout occurs, the controller will go back to idle and RX FIFO is flushed.

FSPI_XMMC_WCMD0

Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15:14	WO	0x0	ADDRB Address Bits 2'b00: 0 bit 2'b01: 24 bits 2'b10: 32 bits 2'b11: From the ABIT register Address bits number select in memory mapped mode, if there is not address command to send, set to zero.
13	WO	0x0	CONT Continuous 1'b0: Disable continuous mode 1'b1: Enable continuous mode AX mode or Continuous mode or XIP mode for device which begins with address.
12	RO	0x0	reserved

Bit	Attr	Reset Value	Description
11:8	WO	0x0	DUMM Dummy Cycles 4'h0: No dummy cycle ... 4'h8: 8 cycles ... Dummy bit cycles in memory mapped mode.
7:0	WO	0x00	CMD Command Command data in memory mapped access mode.

FSPI_XMMC_RCMD0

Address: Operational Base + offset (0x0054)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15:14	WO	0x0	ADDRB Address Bits 2'b00: 0 bit 2'b01: 24 bits 2'b10: 32 bits 2'b11: From the ABIT register Address bits number select in memory mapped mode, if there is not address command to send, set to zero.
13	WO	0x0	CONT Continuous 1'b0: Disable continuous mode 1'b1: Enable continuous mode AX mode or Continuous mode or XIP mode for device which begins with address.
12	RO	0x0	reserved
11:8	WO	0x0	DUMM Dummy Cycles 4'h0: No dummy cycle ... 4'h8: 8 cycles ... Dummy bit cycles in memory mapped mode.
7:0	WO	0x00	CMD Command Command data in memory mapped access mode.

FSPI_XMMC_CTRL

Address: Operational Base + offset (0x0058)

Bit	Attr	Reset Value	Description
31:14	RO	0x00000	reserved
13	RO	0x0	reserved
12	RO	0x0	reserved
11:8	RO	0x0	reserved
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.

Bit	Attr	Reset Value	Description
5	RW	0x1	DEV_HWEN Device AMBA AHB HWRITE Enable 1'b0: Disable 1'b1: Enable
4:0	RO	0x00	reserved

FSPI_MODE

Address: Operational Base + offset (0x005C)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RW	0x0	XMMC_MODE_EN Memory Mapped Mode Enable 1'b0: Disable, indirect access mode 1'b1: Enable, Memory-Mapped mode Before switching from indirect access mode to Memory-Mapped mode, the application should make sure the controller is in idle state and no pending transaction. If switch from Memory-Mapped to indirect access mode, software should initiate a dummy read by CPU before that.

FSPI_DEVRGN

Address: Operational Base + offset (0x0060)

Bit	Attr	Reset Value	Description
31:10	RO	0x000000	reserved
9:8	RW	0x0	DEC_CTRL Decode Control 2'b00: 1 CS# 2'b01: 2 CS# 2'b10: 4 CS# 2'b11: Reserved Only valid in XMMC mode.
7:5	RO	0x0	reserved
4:0	RW	0x17	RSIZE Region Size 5'd0: 1 byte 5'd1: 2 bytes 5'd2: 4 bytes ..... 5'd10: 1K bytes ..... 5'd20: 1M bytes ..... 5'd31: 4G bytes In Memory-Mapped mode, the CS is controlled by AHB address bus, region size is used to generate CS.

FSPI_DEVSIZE0

Address: Operational Base + offset (0x0064)

Bit	Attr	Reset Value	Description
31:5	RO	0x0000000	reserved

Bit	Attr	Reset Value	Description
4:0	RW	0x12	DSIZE Device Size 5'd0: 1 byte 5'd1: 2 bytes 5'd2: 4 bytes ..... 5'd10: 1K bytes ..... 5'd20: 1M bytes ..... 5'd31: 4G byte Device size is used to generate slop over status.

FSPI_TME0

Address: Operational Base + offset (0x0068)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved
1	RW	0x0	SCLK_INATM_EN SCLK Inactive Timeout Enable 1'b0: Disable 1'b1: Enable
0	RO	0x0	reserved

FSPI_XMMC_RX_WTMRK

Address: Operational Base + offset (0x0070)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x02	RX_FULL_WTMRK Memory Mapped Mode Receiver FIFO Water Mark. Default is enough.

FSPI_DMATR

Address: Operational Base + offset (0x0080)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	W1C	0x0	DMATR DMA Trigger Write "1" to start the DMA transfer.

FSPI_DMAADDR

Address: Operational Base + offset (0x0084)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMAADDR DMA Address The destination or source data address in current system.

FSPI_LEN_CTRL

Address: Operational Base + offset (0x0088)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RW	0x0	TRB_SEL Total Transfer Bytes Selection 1'b0: TRB controlled by CMD[TRB] 1'b1: TRB controlled by LEN_EXT

FSPI_LEN_EXT

Address: Operational Base + offset (0x008C)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	TRB_EXT Total Transfer Bytes Extended 32'd0: No data 32'd1: 1 Byte 32'd2: 2 Bytes ... Total data bytes number that will write to /read from the device.

FSPI_XMMCSR

Address: Operational Base + offset (0x0094)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved
1	R/WSC	0x0	SLOPOVER1 Slop Over Register for CS1 1'b0: Normal state 1'b1: Address slop over When the access address in memory map mode is bigger than DEVSIZE, this bit will be set. Write "1" to clear this bit.
0	R/WSC	0x0	SLOPOVER0 Slop Over Register for CS0 1'b0: Normal state 1'b1: Address slop over When the access address in memory map mode is bigger than DEVSIZE, this bit will be set. Write "1" to clear this bit.

FSPI_CMD

Address: Operational Base + offset (0x0100)

Bit	Attr	Reset Value	Description
31:30	WO	0x0	CS Device Chip Select. 2'b00: Chip select 0 2'b01: Chip select 1 2'b10: Reserved 2'b11: Reserved
29:16	WO	0x0000	TRB Total Transfer Bytes 14'd0: No data 14'd1: 1 Byte 14'd2: 2 Bytes ... Total data bytes number that will write to or read from the device.
15:14	WO	0x0	ADDRB Address Bits 2'b00: 0 bit 2'b01: 24 bits 2'b10: 32 bits 2'b11: From the ABIT register Address bits number select in indirect access mode. If there is not address command to send, set to zero.

Bit	Attr	Reset Value	Description
13	WO	0x0	CONT Continuous 1'b0: Disable continuous mode 1'b1: Enable continuous mode AX mode or Continuous mode or XIP mode for device which begins with address.
12	WO	0x0	WR Write or Read 1'b0: Read 1'b1: Write
11:8	WO	0x0	DUMM Dummy Cycles 4'h0: No dummy cycle ... 4'h8: 8 cycles ... Dummy bit cycles in indirect access mode.
7:0	WO	0x00	CMD Command Command data in indirect access mode.

FSPI_ADDR

Address: Operational Base + offset (0x0104)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	ADDR Address Register Indirect access start address data for current command sequence.

FSPI_DATA

Address: Operational Base + offset (0x0108)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DATA Data Register Device data read or write from/to device.

FSPI_CTRL1

Address: Operational Base + offset (0x0200)

Bit	Attr	Reset Value	Description
31:14	RO	0x00000	reserved
13:12	RW	0x0	DATB Data Line Width 2'b00: 1 bit, x1 mode 2'b01: 2 bits, x2 mode 2'b10: 4 bits, x4 mode 2'b11: Reserved Set this DATB to match the CMD sequence before doing indirect access mode and memory mapped access mode.

Bit	Attr	Reset Value	Description
11:10	RW	0x0	ADRB Address Line Width 2'b00: 1 bit, x1 mode 2'b01: 2 bits, x2 mode 2'b10: 4 bits, x4 mode 2'b11: Reserved Set this ADRB to match the CMD sequence before doing indirect access mode and memory mapped access mode.
9:8	RW	0x0	CMDB Command Line Width 2'b00: 1 bit, x1 mode 2'b01: 2 bits, x2 mode 2'b10: 4 bits, x4 mode 2'b11: Reserved Set this CMDB to match the CMD sequence before doing indirect access mode and memory mapped access mode.
7:4	RW	0x0	IDLE_CYCLE Idle Cycles When Switching IO from Output to Input 4'h0: Idle hold is disable 4'h1: Hold the SCLK in idle for 2 cycles when switch to shift in ... 4'hf: Hold the SCLK in idle for 16 cycles when switch to shift in To improve the transform IO timing, the application can set this register to hold the SCLK in low state or high state.
3:2	RO	0x0	reserved
1	RW	0x0	SHIFTPHASE Shift Phase of Data Input in Controller 1'b0: Shift input data at posedge SCLK 1'b1: Shift input data at negedge SCLK The application can select the input data captured by posedge of SCLK when "0" or negedge of SCLK when "1".
0	RW	0x0	SPIM Serial Peripheral Interface Mode 1'b0: Mode 0 1'b1: Mode 3 SPIM is used to control the serial mode (CPOL and CPHA). CPOL indicates clock polarity of Serial master, CPOL=1 for SCLK high while idle, CPOL=0 for SCLK low while not transmitting. CPHA indicates clock phase. The combination of CPOL bit and CPHA bit decides which Serial mode is supported.

FSPI_AX1

Address: Operational Base + offset (0x0214)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x00	AX Auxiliary Data The AX value when doing the continuous read (enhance mode or XIP mode). That is M7-M0 in "Continuous Read Mode".

FSPI_ABIT1

Address: Operational Base + offset (0x0218)

Bit	Attr	Reset Value	Description
31:5	RO	0x0000000	reserved

Bit	Attr	Reset Value	Description
4:0	RW	0x00	ABIT Address Bits Extend 5'h0: 1 bit 5'h1: 2 bits ... 5'h1f: 32 bits Only valid when ADDRB is set to 2'b11.

FSPI_DLL_CTRL1

Address: Operational Base + offset (0x023C)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15	RW	0x0	SCLK_SMP_SEL SCLK sampling selection 1'b0: Bypass DLL 1'b1: From DLL The sampling SCLK source selection.
14:9	RO	0x00	reserved
8:0	RW	0x001	SMP_DLL_CFG Sample Delay Line Configuration 9'h0: 1 DLL element cell 9'h1: 1 DLL element cell 9'h2: 2 DLL element cells ... 9'h1ff: 511 DLL element cells This register to control the sampling delay line cell used. The maximum DLL element cells is decided by process.

FSPI_XMMC_WCMD1

Address: Operational Base + offset (0x0250)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15:14	WO	0x0	ADDRB Address Bits 2'b00: 0 bit 2'b01: 24 bits 2'b10: 32 bits 2'b11: From the ABIT register Address bits number select in memory mapped mode, if there is not address command to send, set to zero.
13	WO	0x0	CONT Continuous 1'b0: Disable continuous mode 1'b1: Enable continuous mode AX mode or Continuous mode or XIP mode for device which begins with address.
12	RO	0x0	reserved
11:8	WO	0x0	DUMM Dummy Cycles 4'h0: No dummy cycle ... 4'h8: 8 cycles ... Dummy bit cycles in memory mapped mode.

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	0x0000	reserved
15:14	WO	0x0	ADDRB Address Bits 2'b00: 0 bit 2'b01: 24 bits 2'b10: 32 bits 2'b11: From the ABIT register Address bits number select in memory mapped mode, if there is not address command to send, set to zero.
13	WO	0x0	CONT Continuous 1'b0: Disable continuous mode 1'b1: Enable continuous mode AX mode or Continuous mode or XIP mode for device which begins with address.
12	RO	0x0	reserved
11:8	WO	0x0	DUMM Dummy Cycles 4'h0: No dummy cycle ... 4'h8: 8 cycles ... Dummy bit cycles in memory mapped mode.
7:0	WO	0x00	CMD Command Command data in memory mapped access mode.

FSPI_DEVSIZE1

Address: Operational Base + offset (0x0264)

Bit	Attr	Reset Value	Description
31:5	RO	0x0000000	reserved
4:0	RW	0x12	DSIZE Device Size 5'd0: 1 byte 5'd1: 2 bytes 5'd2: 4 bytes ..... 5'd10: 1K bytes ..... 5'd20: 1M bytes ..... 5'd31: 4G bytes Device size is used to generate slop over status.

FSPI_TME1

Address: Operational Base + offset (0x0268)

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

Bit	Attr	Reset Value	Description
1	RW	0x0	SCLK_INATM_EN SCLK Inactive Timeout Enable 1'b0: Disable 1'b1: Enable
0	RO	0x0	reserved

26.5 Interface Description

Table 26-2 FSPI(SFC) interface description

Module Pin	Direction	Pin Name	IOMUX Setting
sfc_clk	O	FLASH_WPn/EMMC_RSTn/FSPI_CLK/GPIO1_A3_d	GRF_GPIO1A_IOMUX_L[15:12]=3'b011
sfc_csn0	O	FLASH_CS0n/FSPI_CS0n /I2S1_MCLK_M0/GPIO0_D4_u	GRF_GPIO0B_IOMUX_H[2:0]=3'b011
sfc_csn1	O	FLASH_D5/EMMC_D5/FSPI_CS1n/GPIO0_D1_u	GRF_GPIO0D_IOMUX_L[6:4]=3'b011
sfc_sio0	I/O	FLASH_ALE/FSPI_D0/I2S1_LRCK_M0/GPIO1_A0_d	GRF_GPIO0D_IOMUX_H[2:0]=3'b011
sfc_sio1	I/O	FLASH_RDYn/FSPI_D1/I2S1_SCLK_M0/GPIO1_A1_u	GRF_GPIO1A_IOMUX_L[6:4]=3'b011
sfc_sio2	I/O	FSPI_D2/I2S1_SDO_M0/GPIO0_D6_d	GRF_GPIO0D_IOMUX_H[10:8]=3'b011
sfc_sio3	I/O	FLASH_RDn/FSPI_D3/I2S1_SDI_M0/GPIO1_A2_u	GRF_GPIO1A_IOMUX_L[10:8]=3'b011

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

26.6 Application Notes

26.6.1 Typical Program Flow Without DMA

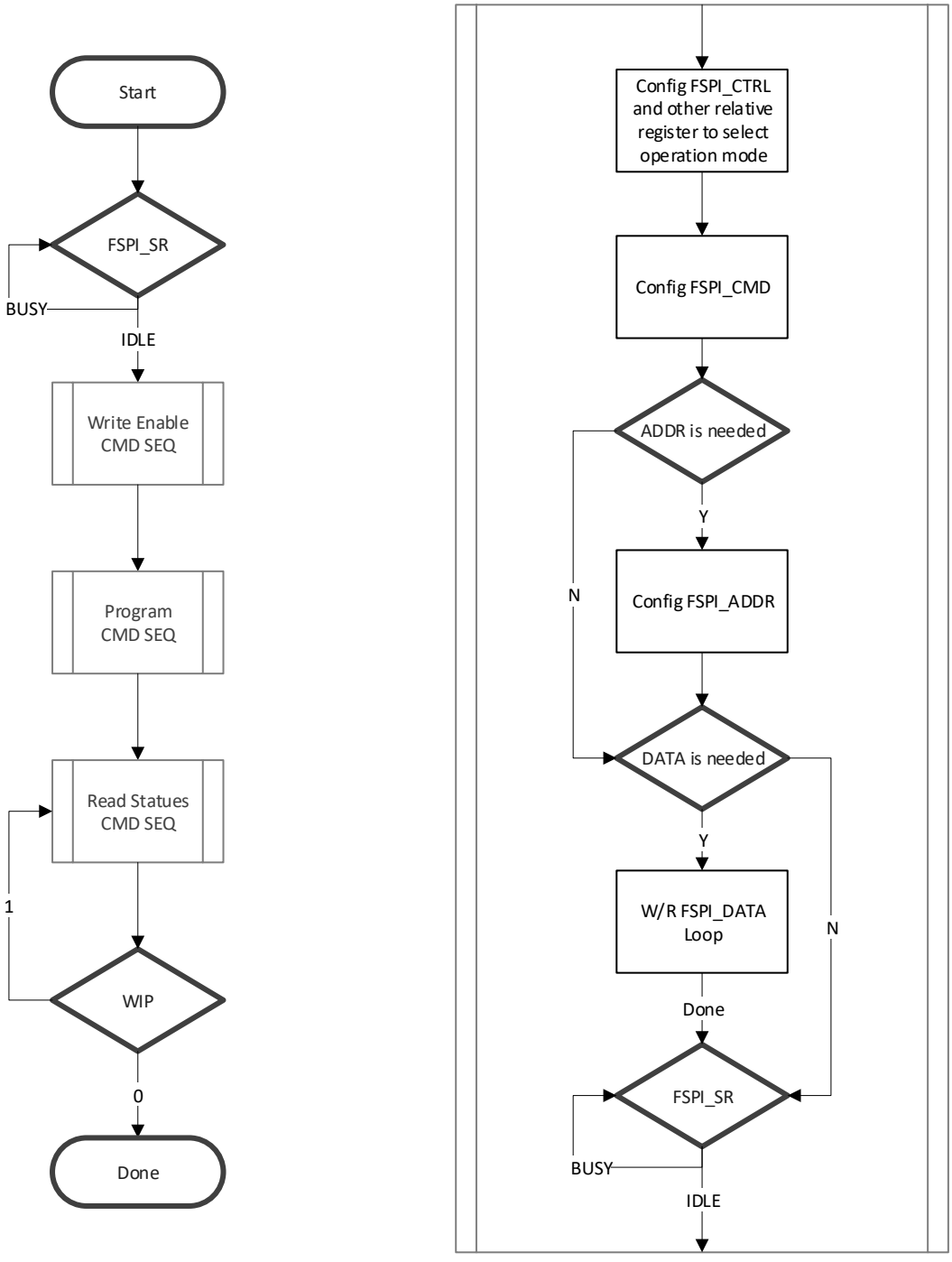


Fig. 26-2 Program Flow

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.

26.6.2 Typical READ Flow Without DMA

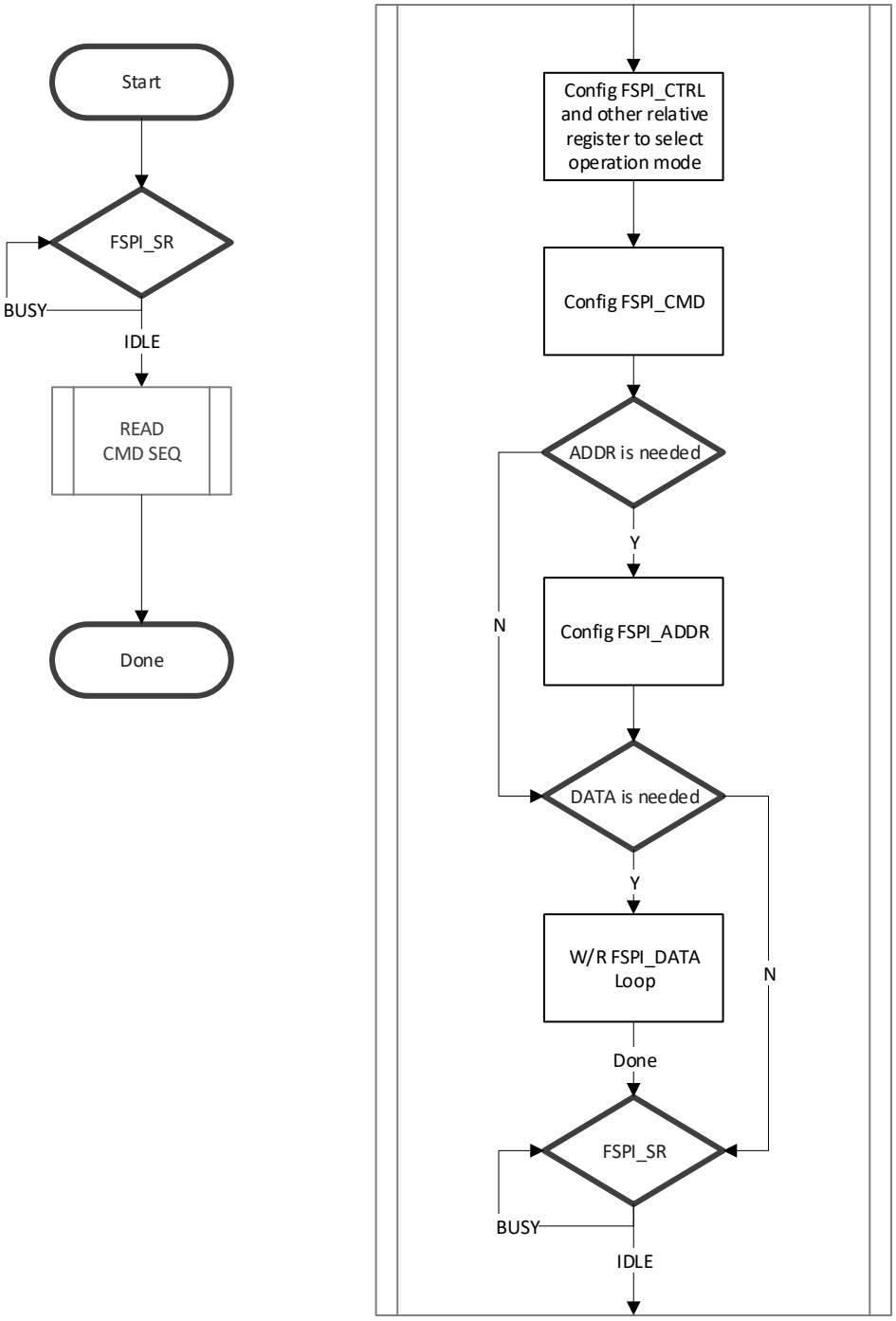


Fig. 26-3 Read Flow

26.6.3 Command Flow with DMA

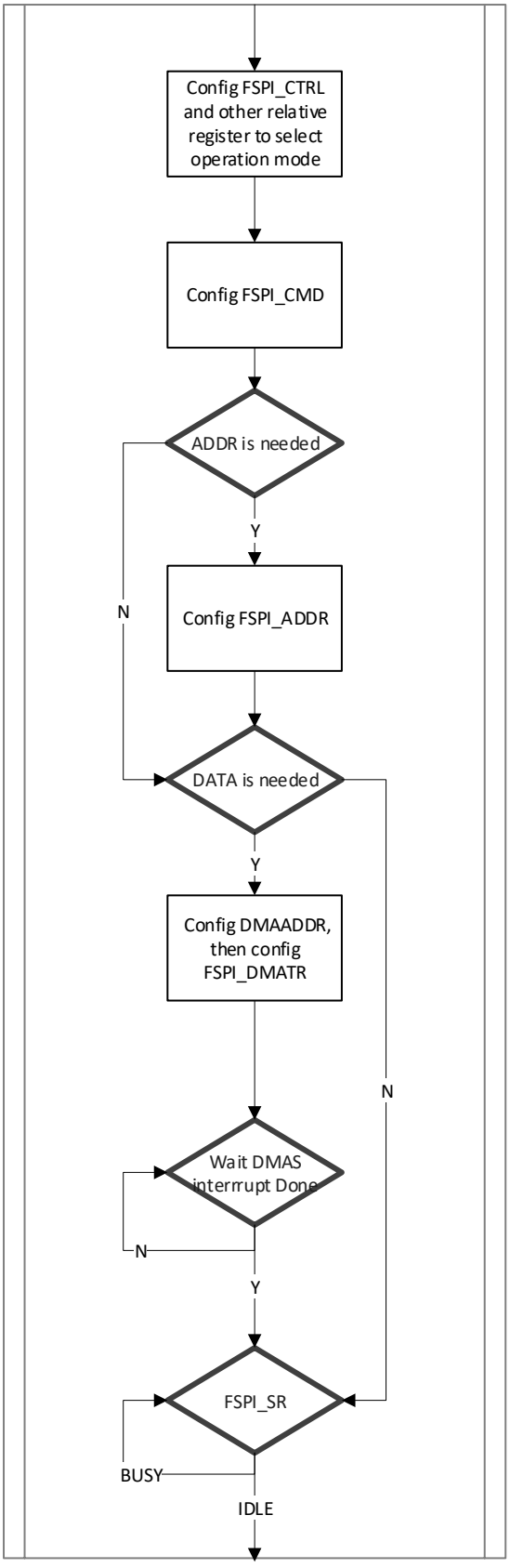


Fig. 26-4 Command with DMA Flow

The total transfer bytes is decided by TRB register in FSPI_CMD and must be aligned to 2 bytes.

26.6.4 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.

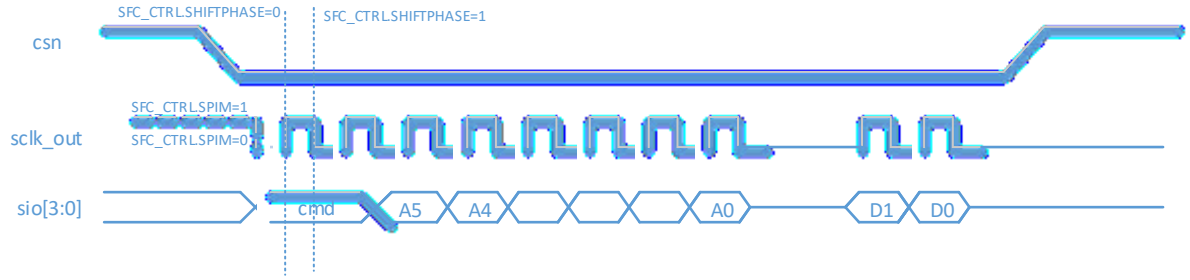


Fig. 26-5 SPI mode

26.6.5 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.

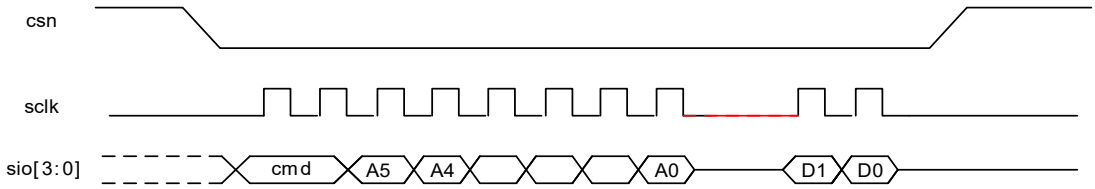


Fig. 26-6 Idle cycles

26.6.6 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 27 Mobile Storage Host Controller

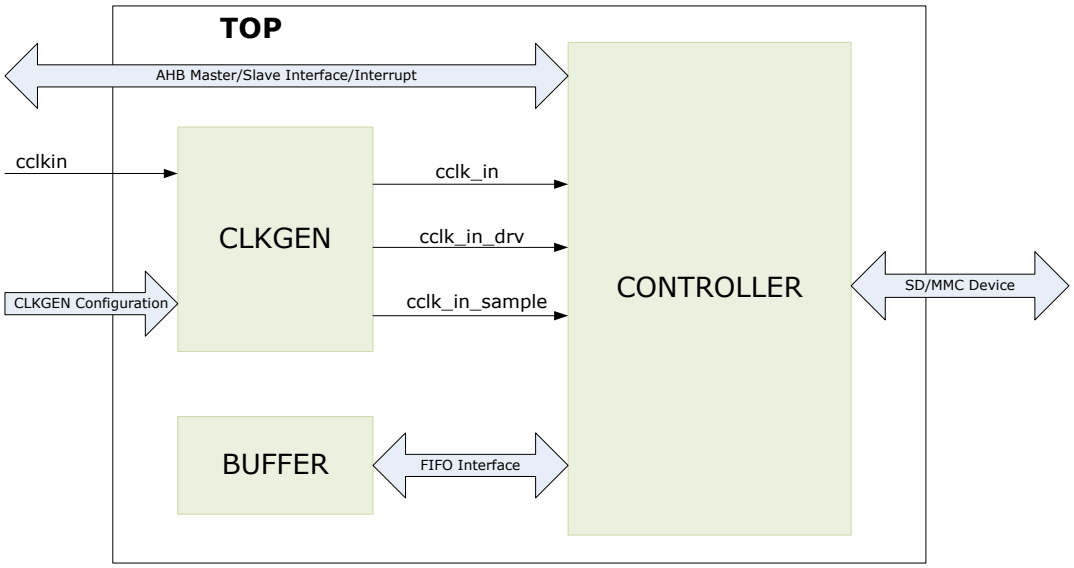
27.1 Overview

The Mobile Storage Host Controller is designed to support Secure Digital memory (SD-max version 3.01) with 1 bits or 4 bits data width, Multimedia Card(MMC-max version 4.51) with 1 bits or 4 bits or 8 bits data width.
The Host Controller is instantiated for SDMMC, SDIO and EMMC. The interface difference between these instances is shown in "Interface Description".
The Host Controller supports following features:

- Bus Interface Features:
  - Support AMBA AHB interface for master and slave
  - Supports internal DMA interface(IDMAC)
    - Supports 16/32-bit data transfers
    - Single engine used for Transmit and Receive, which are mutually exclusive
    - Dual-buffer and chained descriptor linked list
    - Each descriptor can transfer up to 4KB of data in chained mode and 8KB of data in dual-buffer mode
    - Programmable burst size for optimal host bus utilization
  - Support combined single FIFO for both transmit and receive operations
  - Support FIFO size of 256x32
  - Support FIFO over-run and under-run prevention by stopping card clock
- Card Interface Features:
  - Support Secure Digital memory protocol commands
  - Support Secure Digital I/O protocol commands
  - Support Multimedia Card protocol commands
  - Support Command Completion Signal and interrupts to host
  - Support CRC generation and error detection
  - Support programmable baud rate
  - Support power management and power switch
  - Support card detection
  - Support write protection
  - Support hardware reset
  - Support SDIO interrupts in 1-bit and 4-bit modes
  - Support 4-bit mode in SDIO3.0
  - Support SDIO suspend and resume operation
  - Support SDIO read wait
  - Support block size of 1 to 65,535 bytes
  - Support 1-bit, 4-bit and 8-bit SDR modes
  - Supports 4-bit and 8-bit DDR
  - Support boot in 1-bit, 4-bit and 8-bit SDR modes
- Clock Interface Features:
  - Support 0/90/180/270-degree phase shift operation for sample clock (cclk_in_sample) and drive clock(cclk_in_drv) relative to function clock(cclk_in) respectively
  - Support phase tuning using delay line for sample clock(cclk_in_sample) and drive clock(cclk_in_drv) relative to function clock (cclk_in) respectively. The max number of delay element number is 256

27.2 Block Diagram

The Host Controller consists of the following main functional blocks.



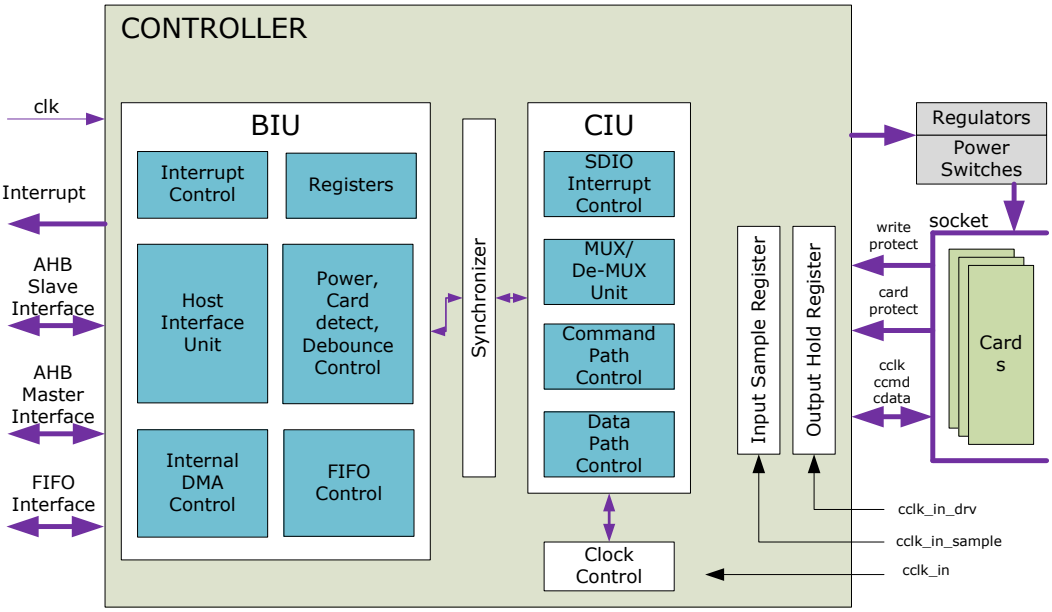


Fig. 27-1 Mobile Storage Host Control Block Diagram

- Clock Generate Unit (CLKGEN): Generates card interface clock cclk_in/ cclk_sample/cclk_drv based on cclkin and configuration information.
  - cclkin: original clock
  - cclk_in: functional clock
  - cclk_sample: sample clock
  - cclk_drv: driver clock
- Asynchronous dual-port memory (BUFFER): Use a two-clock synchronous read and synchronous write dual-port RAM. One of the ports is connected to the host clock, and the other port is connected to the card clock.
- Bus Interface Unit (BIU): Provides AMBA AHB interfaces for register and data read/write.
- Card Interface Unit (CIU): Takes care of the SD/MMC protocols and provides clock management.

27.3 Function Description

27.3.1 Bus Interface Unit

The Bus Interface Unit provides the following functions:

- Host interface
- Interrupt control
- Register access

- External FIFO access
- Power control and card detection

27.3.1.1 Host Interface Unit

The Host Interface Unit is an AHB slave interface, which provides the interface between the SD/MMC card and the host bus.

27.3.1.2 Register Unit

The register unit is part of the bus interface unit; it provides read and write access to the registers.

All registers reside in the Bus Interface Unit clock domain. When a command is sent to a card by setting the start_bit, which is bit[31] of the SDMMC_CMD register, all relevant registers needed for the CIU operation are transferred to the CIU block. During this time, the registers that are transferred from the BIU to the CIU should not be written. The software should wait for the hardware to clear the start bit before writing to these registers again. The register unit has a hardware locking feature to prevent illegal writes to registers. The lock is necessary in order to avoid metastability violations, both because the host and card clock domains are different and to prevent illegal software operations.

Once a command start is issued by setting the start_bit of the SDMMC_CMD register, the following registers cannot be reprogrammed until the command is accepted by the card interface unit:

- SDMMC_CMD – Command
- SDMMC_CMDARG – Command Argument
- SDMMC_BYTCNT – Byte Count
- SDMMC_BLKSIZ – Block Size
- SDMMC_CLKDIV – Clock Divider
- SDMMC_CLKENA – Clock Enable
- SDMMC_CLKSRC – Clock Source
- SDMMC_TMOUT – Timeout
- SDMMC_CTYPE – Card Type

The hardware resets the start_bit once the CIU accepts the command. If a host write to any of these registers is attempted during this locked time, then the write is ignored and the hardware lock error bit is set in the raw interrupt status register. Additionally, if the interrupt is enabled and not masked for a hardware lock error, then an interrupt is sent to the host.

When the Card Interface Unit is in an idle state, it typically takes the following number of clocks for the command handshake, where clk is the BIU clock and cclk_in is the CIU clock:

3 (clk) + 3 (cclk_in)

Once a command is accepted, you can send another command to the CIU-which has a one-deep command queue-under the following conditions:

- If the previous command was not a data transfer command, the new command is sent to the SD/MMC card once the previous command completes.
- If the previous command is a data transfer command and if wait_prvdata_complete (bit[13]) of the SDMMC_CMD register is set for the new command, the new command is sent to the SD/MMC card only when the data transfer completes.
- If the wait_prvdata_complete is 0, then the new command is sent to the SD/MMC card as soon as the previous command is sent. Typically, you should use this only to stop or abort a previous data transfer or query the card status in the middle of a data transfer.

27.3.1.3 Interrupt Controller Unit

The interrupt controller unit generates an interrupt that depends on the controller raw interrupt status, the interrupt-mask register, and the global interrupt-enable register bit. Once an interrupt condition is detected, it sets the corresponding interrupt bit in the raw interrupt status register SDMMC_RINTSTS. The raw interrupt status bit stays on until the software clears the bit by writing a 1 to the interrupt bit; a 0 leaves the bit untouched.

The interrupt port is an active-high, level-sensitive interrupt. The interrupt port is active only when any bit in the raw interrupt status register is active, the corresponding interrupt mask bit is 1, and the global interrupt enable bit is 1. The interrupt port is registered in order to avoid any combinational glitches.

The int_enable is reset to 0 on power-on, and the interrupt mask bits are set to 32'h0, which masks all the interrupts.

Notes:
Before enabling the interrupt, it is always recommended that you write 32'hffff_ffff to the raw interrupt status register in order to clear any pending unserviced interrupts. When clearing interrupts during normal operation, ensure that you clear only the interrupt bits that you serviced.
The SDIO Interrupts, Receive FIFO Data Request (RXDR), and Transmit FIFO Data Request (TXDR) are set by level-sensitive interrupt sources. Therefore, the interrupt source should be first cleared before you can clear the interrupt bit of the Raw Interrupt register. For example, on seeing the Receive FIFO Data Request (RXDR) interrupt, the FIFO should be emptied so that the "FIFO count greater than the RX-Watermark" condition, which triggers the interrupt, becomes inactive. The rest of the interrupts are triggered by a single clock-pulse-width source.

Table 27-1 Bits in Interrupt Status Register

Bits	Interrupt	Description
24	sdio_interrupt	Interrupt from SDIO card. In MMC-Ver3.3-only mode, these bits are always 0.
16	Card no-busy	If card exit busy status, the interrupt happened.
15	End Bit Error (read) / Write no CRC (EBE)	Error in end-bit during read operation, or no data CRC received during write operation. For MMC CMD19, there may be no CRC status returned by the card. Hence, EBE is set for CMD19. The application should not treat this as an error.
14	Auto Command Done (ACD)	Stop/abort commands automatically sent by card unit and not initiated by host; similar to Command Done (CD) interrupt. Recommendation: Software typically need not enable this for non CE-ATA accesses; Data Transfer Over (DTO) interrupt that comes after this interrupt determines whether data transfer has correctly competed.
13	Start Bit Error (SBE)	Error in data start bit when data is read from a card. In 4-bit mode, if DAT[0] line indicates start bit-that is, 0-and any of the other data bits do not have start bit, then this error is set. Busy Complete Interrupt when data is written to the card. This interrupt is generated after completion of busy driven by the card after the last data block is written into the card.
12	Hardware Locked write Error (HLE)	During hardware-lock period, write attempted to one of locked registers. When software sets the start_cmd bit in the SDMMC_CMD register, the Host Controller tries to load the command. If the command buffer is already filled with a command, this error is raised. The software then has to reload the command.
11	FIFO Underrun/ Overrun Error (FRUN)	Host tried to push data when FIFO was full, or host tried to read data when FIFO was empty. Typically this should not happen, except due to error in software. Card unit never pushes data into FIFO when FIFO is full, and pop data when FIFO is empty. If IDMAC is enabled, FIFO under-run/over-run can occur due to a programming error on MSIZE and watermark values in SDMMC_FIFOTH register.
10	Data Starvation by Host Timeout (HTO)	To avoid data loss, card clock out is stopped if FIFO is empty when writing to card, or FIFO is full when reading from card. Whenever card clock is stopped to avoid data loss, data-starvation timeout counter is started with data-timeout value. This interrupt is set if host does not fill data into FIFO during write to card, or does not read from FIFO during read from card

Bits	Interrupt	Description
		before timeout period. Even after timeout, card clock stays in stopped state, with CIU state machines waiting. It is responsibility of host to push or pop data into FIFO upon interrupt, which automatically restarts cclk_out and card state machines. Even if host wants to send stop/abort command, it still needs to ensure it has to push or pop FIFO so that clock starts in order for stop/abort command to send on command signal along with data that is sent or received on data line.
9	Data Read Timeout (DRTO)	In Normal functioning mode: Data read timeout (DRTO) Data timeout occurred. Data Transfer Over (DTO) also set if data timeout occurs. In Boot Mode: Boot Data Start (BDS) When set, indicates that Host Controller has started to receive boot data from the card. A write to this register with a value of 1 clears this interrupt.
8	Response Timeout (RTO)	In normal functioning mode: Response timeout (RTO) Response timeout occurred. Command Done (CD) also set if response timeout occurs. If command involves data transfer and when response times out, no data transfer is attempted by Host Controller. In Boot Mode: Boot Ack Received (BAR) When expect_boot_ack is set, on reception of a boot acknowledge pattern—0-1-0—this interrupt is asserted. A write to this register with a value of 1 clears this interrupt.
7	Data CRC Error (DCRC)	Received Data CRC does not match with locally-generated CRC in CIU. Can also occur if the Write CRC status is incorrectly sampled by the Host.
6	Response CRC Error (RCRC)	Response CRC does not match with locally-generated CRC in CIU.
5	Receive FIFO Data Request (RXDR)	Interrupt set during read operation from card when FIFO level is greater than Receive-Threshold level. Recommendation: In DMA modes, this interrupt should not be enabled. In non-DMA mode: pop RX_WMark + 1 data from FIFO.

4	Transmit FIFO Data Request (TXDR)	Interrupt set during write operation to card when FIFO level reaches less than or equal to Transmit-Threshold level.
		Recommendation:
		In DMA modes, this interrupt should not be enabled.
		In non-DMA mode:
		if (pending_bytes > (FIFO_DEPTH - TX_WMark))
		push (FIFO_DEPTH - TX_WMark) data into FIFO
		else
		push pending_bytes data into FIFO

3	Data Transfer Over (DTO)	Indicates Data transfer completed. Though on detection of errors-Start Bit Error, Data CRC error, and so on, DTO may or may not be set; the application must issue CMD12, which ensures that DTO is set.

Bits	Interrupt	Description
		Recommendation: In non-DMA mode, when data is read from card, on seeing interrupt, host should read any pending data from FIFO. In DMA mode, DMA controllers guarantee FIFO is flushed before interrupt. DTO bit is set at the end of the last data block, even if the device asserts MMC busy after the last data block.
2	Command Done(CD)	Command sent to card and got response from card, even if Response Error or CRC error occurs.
1	Response Error (RE)	Error in received response set if one of following occurs: ● Transmission bit != 0 ● Command index mismatch ● End-bit != 1
0	Card-Detect (CDT)	When one or more cards inserted or removed, this interrupt occurs. Software should read card-detect register to determine current card status. Recommendation: After power-on and before enabling interrupts, software should read card detect register and store it in memory. When interrupt occurs, it should read card detect register and compare it with value stored in memory to determine which card(s) were removed/inserted. Before exiting ISR, software should update memory with new card-detect value.

27.3.1.4 FIFO Controller Unit

The FIFO controller interfaces the external FIFO to the host interface and the card controller unit. When FIFO overrun and under-run conditions occur, the card clock stops in order to avoid data loss.

The FIFO uses a two-clock synchronous read and synchronous write dual-port RAM. One of the ports is connected to the host clock(clk), and the second port is connected to the card clock(cclk_in).

Notes: The FIFO controller does not support simultaneous read/write access from the same port. For debugging purposes, the software may try to write into the FIFO and read back the data; results are indeterminate, since the design does not support read/write access from the same port.

27.3.1.5 Card Detection Unit

The register unit has registers that control the power. Power to each card can be selectively turned on or off.

The card detection unit looks for any changes in the card-detect signals for card insertion or card removal. It filters out the debounce associated with mechanical insertion or removal, and generates one interrupt to the host. You can program the debounce filter value.

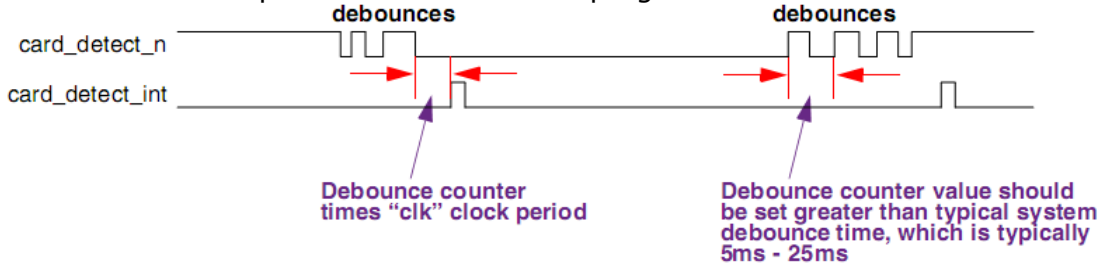


Fig. 27-2 SD/MMC Card-Detect Signal

27.3.2 Card Interface Unit

The Card Interface Unit (CIU) interfaces with the Bus Interface Units (BIU) and the external devices. The host writes command parameters to the BIU control registers, and these parameters are then passed to the CIU. Depending on control register values, the CIU generates SD/MMC command and data traffic on a selected card bus according to SD/MMC protocol. The Host Controller accordingly controls the command and data path.

The following software restrictions should be met for proper CIU operation:
- Only one data transfer command can be issued at a time.
- During an open-ended card write operation, if the card clock is stopped because the FIFO is empty, the software must first fill the data into the FIFO and start the card clock. It can then issue only a stop/abort command to the card.
- When issuing card reset commands (CMD0, CMD15 or CMD52_reset) while a card data transfer is in progress, the software must set the stop_abort_cmd bit in the SDMMC_CMD register so that the Host Controller can stop the data transfer after issuing the card reset command.
- When the data end bit error is set in the SDMMC_RINTSTS register, the Host Controller does not guarantee SDIO interrupts. The software should ignore the SDIO interrupts and issue the stop/abort command to the card, so that the card stops sending the read data.
- If the card clock is stopped because the FIFO is full during a card read, the software should read at least two FIFO locations to start the card clock.

The CIU block consists of the following primary functional blocks:
- Command path
- Data path
- SDIO interrupt control
- Clock control
- Error detection

27.3.2.1 Command Path

The command path performs the following functions:
- Loads clock parameters
- Loads card command parameters
- Sends commands to card bus (ccmd_out line)
- Receives responses from card bus (ccmd_in line)
- Sends responses to BIU
- Drives the P-bit on command line

A new command is issued to the Host Controller by programming the BIU registers and setting the start_cmd bit in the SDMMC_CMD register. The BIU asserts start_cmd, which indicates that a new command is issued to the SD/MMC device. The command path loads this new command (command, command argument, timeout) and sends acknowledge to the BIU by asserting cmd_taken.

Once the new command is loaded, the command path state machine sends a command to the device bus-including the internally generated CRC7-and receives a response, if any. The state machine then sends the received response and signals to the BIU that the command is done, and then waits for eight clocks before loading a new command.

Load Command Parameters

One of the following commands or responses is loaded in the command path:
- New command from BIU – When start_cmd is asserted, then the start_cmd bit is set in the SDMMC_CMD register.
- Internally generated auto-stop command – When the data path ends, the stop command request is loaded.
- IRQ response with RCA 0x000 – When the command path is waiting for an IRQ response from the MMC card and a "send irq response" request is signaled by the BIU, then the send_irq_response bit is set in the SDMMC_CTRL register.

Loading a new command from the BIU in the command path depends on the following SDMMC_CMD register bit settings:
- update_clock_registers_only – If this bit is set in the SDMMC_CMD register, the command path updates only the clock enable, clock divider, and clock source registers. If this bit is not set, the command path loads the command, command argument, and timeout registers; it then starts processing the new command.
- wait_prvdata_complete – If this bit is set, the command path loads the new command under one of the following conditions:
  - Immediately, if the data path is free (that is, there is no data transfer in progress), or if an open-ended data transfer is in progress (byte_count = 0).
  - After completion of the current data transfer, if a predefined data transfer is in

progress.

Send Command and Receive Response

Once a new command is loaded in the command path, update_clock_registers_only bit is unset – the command path state machine sends out a command on the device bus; the command path state machine is illustrated in following figure.

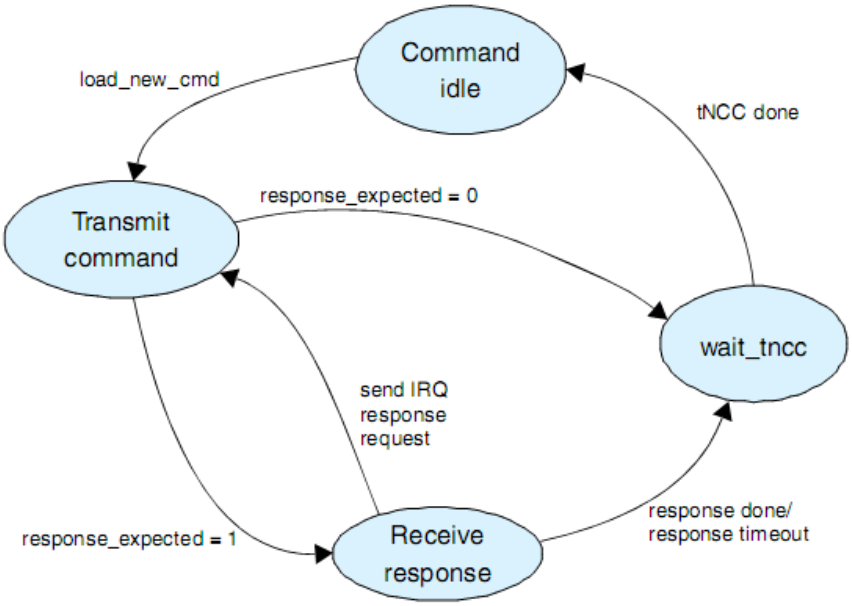


Fig. 27-3 Host Controller Command Path State Machine

The command path state machine performs the following functions, according to SDMMC_CMD register bit values:

- send_initialization – Initialization sequence of 80 clocks is sent before sending the command.
- response_expected – Response is expected for the command. After the command is sent out, the command path state machine receives a 48-bit or 136-bit response and sends it to the BIU. If the start bit of the card response is not received within the number of clocks programmed in the timeout register, then the response timeout and command done bit is set in the Raw Interrupt Status register as a signal to the BIU. If the response-expected bit is not set, the command path sends out a command and signals a response done to the BIU; that is, the command done bit is set in the Raw Interrupt Status register.
- response_length – If this bit is set, a 136-bit response is received; if it is not set, a 48-bit response is received.
- check_response_crc – If this bit is set, the command path compares CRC7 received in the response with the internally-generated CRC7. If the two do not match, the response CRC error is signaled to the BIU; that is, the response CRC error bit is set in the Raw Interrupt Status register SDMMC_RINTSTS.

Send Response to BIU

If the response_expected bit is set in the SDMMC_CMD register, the received response is sent to the BIU. The Response0 register is updated for a short response, and the Response3, Response2, Response1, and Response0 registers are updated on a long response, after which the Command Done bit is set. If the response is for an auto_stop command sent by the CIU, the response is saved in the Response1 register, after which the Auto Command Done bit is set.

Additionally, the command path checks for the following:

- Transmission bit = 0
- Command index matches command index of the sent command
- End bit = 1 in received card response

The command index is not checked for a 136-bit response or if the check_response_crc bit is unset. For a 136-bit response and reserved CRC 48-bit responses, the command index is reserved-that is, 111111.

Polling Command Completion Signal

The device generates the Command Completion Signal in order to notify the host controller of the normal command completion or command termination.

Command Completion Signal Detection and Interrupt to Host Processor

If the ccs_expected bit is set in the SDMMC_CMD register, the Command Completion Signal (CCS) from the device is indicated by setting the Data Transfer Over (DTO) bit in the SDMMC_RINTSTS register. The Host Controller generates a Data Transfer Over (DTO) interrupt if this interrupt is not masked.

Command Completion Signal Timeout

If the command expects a CCS from the device—if the ccs_expected bit is set in the SDMMC_CMD register—the command state machine waits for the CCS and remains in a wait_CCSS state. If the device fails to send out the CCS, the host software should implement a timeout mechanism to free the command and data path. The host controller does not implement a hardware timer; it is the responsibility of the host software to maintain a software timer.

In the event of a CCS timeout, the host should issue a CCSD by setting the send_ccsd bit in the CTRL register. The host controller command state machine sends the CCSD to the device and exits to an idle state. After sending the CCSD, the host should also send a CMD12 to the device in order to abort the outstanding command.

Send Command Completion Signal Disable

If the send_ccsd bit is set in the SDMMC_CTRL register, the host sends a Command Completion Signal Disable (CCSD) pattern on the CMD line. The host can send the CCSD while waiting for the CCS or after a CCS timeout happens.

After sending the CCSD pattern, the host sets the Command Done (CD) bit in SDMMC_RINTSTS and also generates an interrupt to the host if the Command Done interrupt is not masked.

27.3.2.2 Data Path

The data path block pops the data FIFO and transmits data on cdata_out during a write data transfer, or it receives data on cdata_in and pushes it into the FIFO during a read data transfer. The data path loads new data parameters—that is, data expected, read/write data transfer, stream/block transfer, block size, byte count, card type, timeout registers—whenever a data transfer command is not in progress.

If the data_expected bit is set in the SDMMC_CMD register, the new command is a data transfer command and the data path starts one of the following:

- Transmit data if the read/write bit = 1
- Data receive if read/write bit = 0

Data Transmit

The data transmit state machine, illustrated in following figure, starts data transmission two clocks after a response for the data write command is received; this occurs even if the command path detects a response error or response CRC error. If a response is not received from the card because of a response timeout, data is not transmitted. Depending upon the value of the transfer_mode bit in the SDMMC_CMD register, the data transmit state machine puts data on the card data bus in a stream or in block(s).

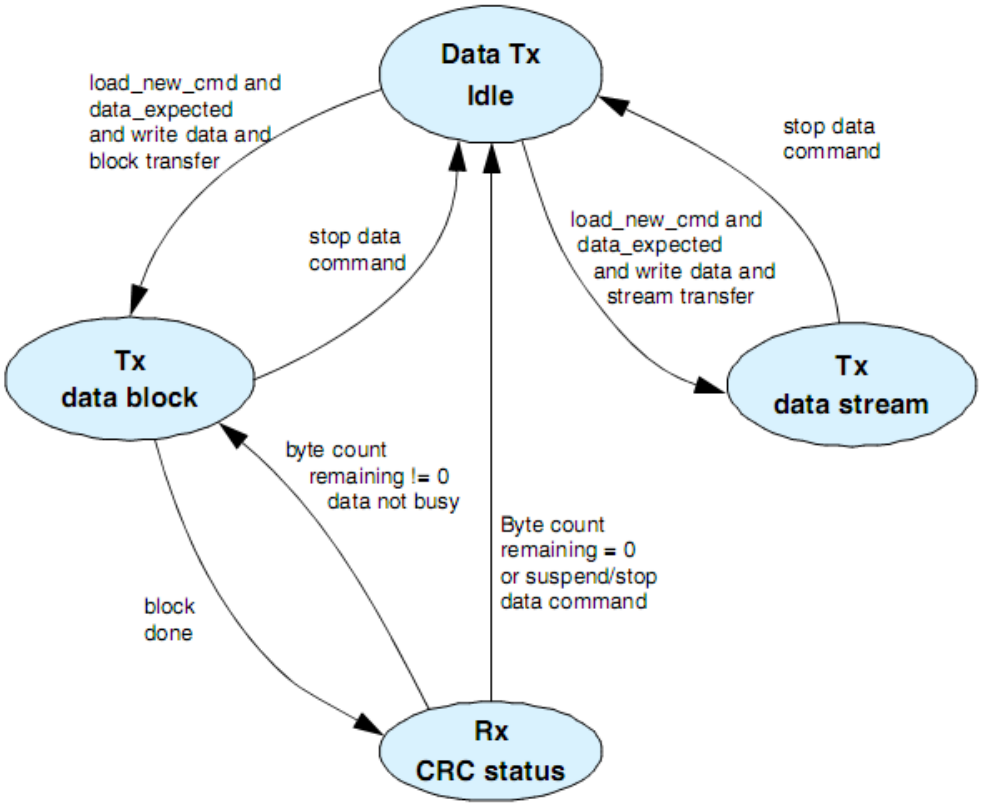


Fig. 27-4 Host Controller Data Transmit State Machine

Stream Data Transmit

If the transfer_mode bit in the SDMMC_CMD register is set to 1, it is a stream-write data transfer. The data path pops the FIFO from the BIU and transmits in a stream to the card data bus. If the FIFO becomes empty, the card clock is stopped and restarted once data is available in the FIFO.

If the byte_count register is programmed to 0, it is an open-ended stream-write data transfer. During this data transfer, the data path continuously transmits data in a stream until the host software issues a stop command. A stream data transfer is terminated when the end bit of the stop command and end bit of the data match over two clocks.

If the byte_count register is programmed with a non-zero value and the send_auto_stop bit is set in the SDMMC_CMD register, the stop command is internally generated and loaded in the command path when the end bit of the stop command occurs after the last byte of the stream write transfer matches.

This data transfer can also terminate if the host issues a stop command before all the data bytes are transferred to the card bus.

Single Block Data

If the transfer_mode bit in the SDMMC_CMD register is set to 0 and the byte_count register value is equal to the value of the block_size register, a single-block write-data transfer occurs. The data transmit state machine sends data in a single block, where the number of bytes equals the block size, including the internally-generated CRC16.

If the SDMMC_CTYPE register bit for the selected card – indicated by the card_num value in the SDMMC_CMD register – is set for a 1-bit, 4-bit, or 8-bit data transfer, the data is transmitted on 1, 4, or 8 data lines, respectively, and CRC16 is separately generated and transmitted for 1, 4, or 8 data lines, respectively.

After a single data block is transmitted, the data transmit state machine receives the CRC status from the card and signals a data transfer to the BIU; this happens when the data-transfer-over bit is set in the SDMMC_RINTSTS register.

If a negative CRC status is received from the card, the data path signals a data CRC error to the BIU by setting the data CRC error bit in the SDMMC_RINTSTS register.

Additionally, if the start bit of the CRC status is not received by two clocks after the end of the data block, a CRC status start bit error is signaled to the BIU by setting the write-no-CRC bit in the SDMMC_RINTSTS register.

Multiple Block Data

A multiple-block write-data transfer occurs if the transfer_mode bit in the SDMMC_CMD register is set to 0 and the value in the byte_count register is not equal to the value of the block_size register. The data transmit state machine sends data in blocks, where the number of bytes in a block equals the block size, including the internally-generated CRC16.

If the SDMMC_CTYPE register bit for the selected card – indicated by the card_num value in the SDMMC_CMD register – is set to 1-bit, 4-bit, or 8-bit data transfer, the data is transmitted on 1, 4, or 8 data lines, respectively, and CRC16 is separately generated and transmitted on 1, 4, or 8 data lines, respectively.

After one data block is transmitted, the data transmit state machine receives the CRC status from the card. If the remaining byte_count becomes 0, the data path signals to the BIU that the data transfer is done; this happens when the data-transfer-over bit is set in the SDMMC_RINTSTS register.

If the remaining data bytes are greater than 0, the data path state machine starts to transmit another data block.

If a negative CRC status is received from the card, the data path signals a data CRC error to the BIU by setting the data CRC error bit in the SDMMC_RINTSTS register, and continues further data transmission until all the bytes are transmitted.

Additionally, if the CRC status start bit is not received by two clocks after the end of a data block, a CRC status start bit error is signaled to the BIU by setting the write-no-CRC bit in the SDMMC_RINTSTS register; further data transfer is terminated.

If the send_auto_stop bit is set in the SDMMC_CMD register, the stop command is internally generated during the transfer of the last data block, where no extra bytes are transferred to the card. The end bit of the stop command may not exactly match the end bit of the CRC status in the last data block.

If the block size is less than 4, 16, or 32 for card data widths of 1 bit, 4 bits, or 8 bits, respectively, the data transmit state machine terminates the data transfer when all the data is transferred, at which time the internally generated stop command is loaded in the command path.

If the byte_count is 0 – the block size must be greater than 0 – it is an open-ended block transfer. The data transmit state machine for this type of data transfer continues the block-write data transfer until the host software issues a stop or abort command.

Data Receive

The data-receive state machine, illustrated in following figure, receives data two clock cycles after the end bit of a data read command, even if the command path detects a response error or response CRC error. If a response is not received from the card because a response timeout occurs, the BIU does not receive a signal that the data transfer is complete; this happens if the command sent by the Host Controller is an illegal operation for the card, which keeps the card from starting a read data transfer.

If data is not received before the data timeout, the data path signals a data timeout to the BIU and an end to the data transfer done. Based on the value of the transfer_mode bit in the SDMMC_CMD register, the data-receive state machine gets data from the card data bus in a stream or block(s).

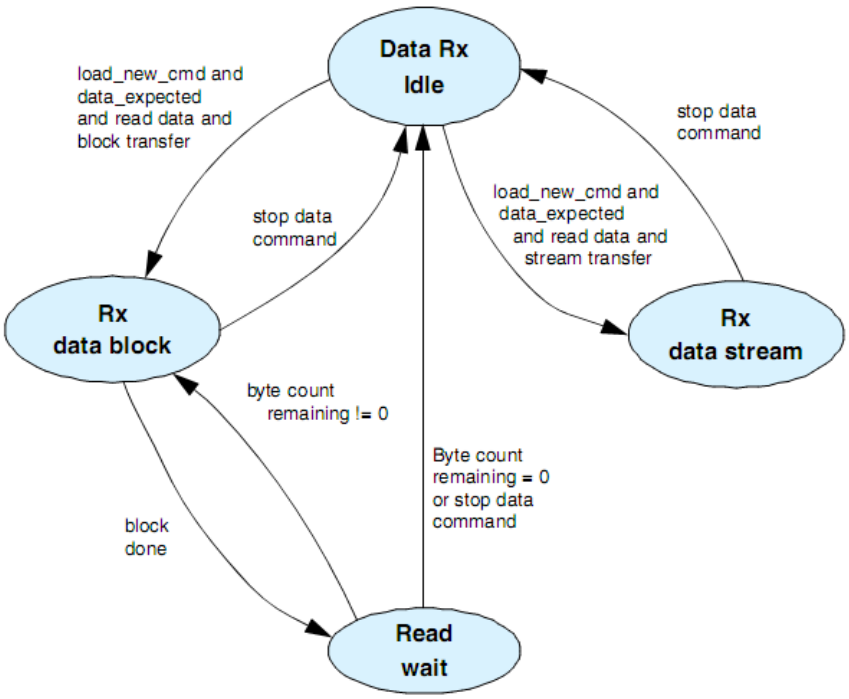


Fig. 27-5 Host Controller Data Receive State Machine

Stream Data Read

A stream-read data transfer occurs if the transfer_mode bit in the SDMMC_CMD register equals 1, at which time the data path receives data from the card and pushes it to the FIFO. If the FIFO becomes full, the card clock stops and restarts once the FIFO is no longer full. An open-ended stream-read data transfer occurs if the byte_count register equals 0. During this type of data transfer, the data path continuously receives data in a stream until the host software issues a stop command. A stream data transfer terminates two clock cycles after the end bit of the stop command.

If the byte_count register contains a non-zero value and the send_auto_stop bit is set in the SDMMC_CMD register, a stop command is internally generated and loaded into the command path, where the end bit of the stop command occurs after the last byte of the stream data transfer is received. This data transfer can terminate if the host issues a stop or abort command before all the data bytes are received from the card.

Single-Block Data Read

A single-block read-data transfer occurs if the transfer_mode bit in the SDMMC_CMD register is set to 0 and the value of the byte_count register is equal to the value of the block_size register. When a start bit is received before the data times out, data bytes equal to the block size and CRC16 are received and checked with the internally-generated CRC16. If the SDMMC_CTYPE register bit for the selected card – indicated by the card_num value in the SDMMC_CMD register – is set to a 1-bit, 4-bit, or 8-bit data transfer, data is received from 1, 4, or 8 data lines, respectively, and CRC16 is separately generated and checked for 1, 4, or 8 data lines, respectively. If there is a CRC16 mismatch, the data path signals a data CRC error to the BIU. If the received end bit is not 1, the BIU receives an end-bit error.

Multiple-Block Data Read

If the transfer_mode bit in the SDMMC_CMD register is set to 0 and the value of the byte_count register is not equal to the value of the block_size register, it is a multiple-block read-data transfer. The data-receive state machine receives data in blocks, where the number of bytes in a block is equal to the block size, including the internally-generated CRC16.

If the SDMMC_CTYPE register bit for the selected card – indicated by the card_num value in the SDMMC_CMD register – is set to a 1-bit, 4-bit, or 8-bit data transfer, data is received from 1, 4, or 8 data lines, respectively, and CRC16 is separately generated and checked for 1, 4, or 8 data lines, respectively.

After a data block is received, if the remaining byte_count becomes 0, the data path signals a data transfer to the BIU.

If the remaining data bytes are greater than 0, the data path state machine causes another data block to be received. If CRC16 of a received data block does not match the internally-generated CRC16, a data CRC error to the BIU and data reception continue further data transmission until all bytes are transmitted.
Additionally, if the end of a received data block is not 1, data on the data path signals terminate the bit error to the CIU and the data-receive state machine terminates data reception, waits for data timeout, and signals to the BIU that the data transfer is complete.
If the send_auto_stop bit is set in the SDMMC_CMD register, the stop command is internally generated when the last data block is transferred, where no extra bytes are transferred from the card; the end bit of the stop command may not exactly match the end bit of the last data block.
If the requested block size for data transfers to cards is less than 4, 16, or 32 bytes for 1-bit, 4-bit, or 8-bit data transfer modes, respectively, the data-transmit state machine terminates the data transfer when all data is transferred, at which point the internally-generated stop command is loaded in the command path. Data received from the card after that are then ignored by the data path.
If the byte_count is 0—the block size must be greater than 0—it is an open-ended block transfer. For this type of data transfer, the data-receive state machine continues the block-read data transfer until the host software issues a stop or abort command.

Auto-Stop

The Host Controller internally generates a stop command and is loaded in the command path when the send_auto_stop bit is set in the SDMMC_CMD register.
The software should set the send_auto_stop bit according to details listed in following table.

Table 27-2 Auto-Stop Generation

Card type	Transfer type	Byte Count	send_auto_stop bit set	Comments
MMC	Stream read	0	No	Open-ended stream
MMC	Stream read	>0	Yes	Auto-stop after all bytes transfer
MMC	Stream write	0	No	Open-ended stream
MMC	Stream write	>0	Yes	Auto-stop after all bytes transfer
MMC	Single-block read	>0	No	Byte count =0 is illegal
MMC	Single-block write	>0	No	Byte count =0 is illegal
MMC	Multiple-block read	0	No	Open-ended multiple block
MMC	Multiple-block read	>0	Yes①	Pre-defined multiple block
MMC	Multiple-block write	0	No	Open-ended multiple block
MMC	Multiple-block write	>0	Yes①	Pre-defined multiple block
SDMEM	Single-block read	>0	No	Byte count =0 is illegal
SDMEM	Single-block write	>0	No	Byte count =0 illegal
SDMEM	Multiple-block read	0	No	Open-ended multiple block
SDMEM	Multiple-block read	>0	Yes	Auto-stop after all bytes transfer
SDMEM	Multiple-block write	0	No	Open-ended multiple block
SDMEM	Multiple-block write	>0	Yes	Auto-stop after all bytes transfer
SDIO	Single-block read	>0	No	Byte count =0 is illegal
SDIO	Single-block write	>0	No	Byte count =0 illegal
SDIO	Multiple-block read	0	No	Open-ended multiple block
SDIO	Multiple-block read	>0	No	Pre-defined multiple block
SDIO	Multiple-block write	0	No	Open-ended multiple block
SDIO	Multiple-block	>0	No	Pre-defined multiple block

Card type	Transfer type	Byte Count	send_auto_stop bit set	Comments
	write			

*①:The condition under which the transfer mode is set to block transfer and byte_count is equal to block size is treated as a single-block data transfer command for both MMC and SD cards. If byte_count = n*block_size (n = 2, 3, …), the condition is treated as a predefined multiple-block data transfer command. In the case of an MMC card, the host software can perform a predefined data transfer in two ways: 1) Issue the CMD23 command before issuing CMD18/CMD25 commands to the card – in this case, issue MD18/CMD25 commands without setting the send_auto_stop bit. 2) Issue CMD18/CMD25 commands without issuing CMD23 command to the card, with the send_auto_stop bit set. In this case, the multiple-block data transfer is terminated by an internally-generated auto-stop command after the programmed byte count.*

The following list conditions for the auto-stop command.

- Stream read for MMC card with byte count greater than 0 – The Host Controller generates an internal stop command and loads it into the command path so that the end bit of the stop command is sent out when the last byte of data is read from the card and no extra data byte is received. If the byte count is less than 6 (48 bits), a few extra data bytes are received from the card before the end bit of the stop command is sent.
- Stream write for MMC card with byte count greater than 0 - The Host Controller generates an internal stop command and loads it into the command path so that the end bit of the stop command is sent when the last byte of data is transmitted on the card bus and no extra data byte is transmitted. If the byte count is less than 6 (48 bits), the data path transmits the data last in order to meet the above condition.
- Multiple-block read memory for SD card with byte count greater than 0 – If the block size is less than 4 (single-bit data bus), 16 (4-bit data bus), or 32 (8-bit data bus), the auto-stop command is loaded in the command path after all the bytes are read. Otherwise, the top command is loaded in the command path so that the end bit of the stop command is sent after the last data block is received.
- Multiple-block write memory for SD card with byte count greater than 0 – If the block size is less than 3 (single-bit data bus), 12 (4-bit data bus), or 24 (8-bit data bus), the auto-stop command is loaded in the command path after all data blocks are transmitted. Otherwise, the stop command is loaded in the command path so that the end bit of the stop command is sent after the end bit of the CRC status is received.
- Precaution for host software during auto-stop – Whenever an auto-stop command is issued, the host software should not issue a new command to the SD/MMC device until the auto-stop is sent by the Host Controller and the data transfer is complete. If the host issues a new command during a data transfer with the auto-stop in progress, an auto-stop command may be sent after the new command is sent and its response is received; this can delay sending the stop command, which transfers extra data bytes. For a stream write, extra data bytes are erroneous data that can corrupt the card data. If the host wants to terminate the data transfer before the data transfer is complete, it can issue a stop or abort command, in which case the Host Controller does not generate an auto-stop command.

27.3.2.3 Non-Data Transfer Commands that Use Data Path

Some non-data transfer commands (non-read/write commands) also use the data path. Following table lists the commands and register programming requirements for them.

Table 27-3 Non-data Transfer Commands and Requirements

Base Address [12:8]	CMD 27	CMD 30	CMD 42	ACMD 13	ACMD 22	ACMD 51
SDMMC_CMD register programming						
cmd_index	6'h1B	6'h1E	6'h2A	6'h0D	6'h16	6'h33
response_expect	1	1	1	1	1	1
rResponse_length	0	0	0	0	0	0
check_response_crc	1	1	1	1	1	1
data_expected	1	1	1	1	1	1
read/write	1	0	1	0	0	0
transfer_mode	0	0	0	0	0	0
send_auto_stop	0	0	0	0	0	0

Base Address [12:8]	CMD 27	CMD 30	CMD 42	ACMD 13	ACMD 22	ACMD 51
wait_prevdata_complete	0	0	0	0	0	0
stop_abort_cmd	0	0	0	0	0	0
Command Argument register programming						
	stuff bits	32-bit write protect data address	stuff bits	stuff bits	stuff bits	stuff bits
Block Size register programming						
	16	4	Num_bytes①	64	4	8
Byte Count register programming						
	16	4	Num_bytes①	64	4	8

①: Num_bytes = No. of bytes specified as per the lock card data structure (Refer to the SD specification and the MMC specification)

27.3.2.4 SDIO Interrupt Control

Interrupts for SD cards are reported to the BIU by asserting an interrupt signal for two clock cycles. SDIO cards signal an interrupt by asserting cdata_in low during the interrupt period; an interrupt period for the selected card is determined by the interrupt control state machine. An interrupt period is always valid for non-active or non-selected cards, and 1-bit data mode for the selected card. An interrupt period for a wide-bus active or selected card is valid for the following conditions:

- Card is idle
- Non-data transfer command in progress
- Third clock after end bit of data block between two data blocks
- From two clocks after end bit of last data until end bit of next data transfer command

Bear in mind that, in the following situations, the controller does not sample the SDIO interrupt of the selected card when the card data width is 4 bits. Since the SDIO interrupt is level-triggered, it is sampled in a further interrupt period and the host does not lose any SDIO interrupt from the card.

- Read/Write Resume – The CIU treats the resume command as a normal data transfer command. SDIO interrupts during the resume command are handled similarly to other data commands. According to the SDIO specification, for the normal data command the interrupt period ends after the command end bit of the data command; for the resume command, it ends after the response end bit. In the case of the resume command, the Controller stops the interrupt sampling period after the resume command end bit, instead of stopping after the response end bit of the resume command.
- Suspend during read transfer – If the read data transfer is suspended by the host, the host sets the abort_read_data bit in the controller to reset the data state machine. In the CIU, the SDIO interrupts are handled such that the interrupt sampling starts after the abort_read_data bit is set by the host. In this case the controller does not sample SDIO interrupts between the period from response of the suspend command to setting the abort_read_data bit, and starts sampling after setting the abort_read_data bit.

27.3.2.5 Clock Control

The clock control block provides different clock frequencies required for SD/MMC cards. The cclk_in signal is the source clock (cclk_in >= card max operating frequency) for clock divider of the clock control block. This source clock (cclk_in) is used to generate different card clock frequencies (cclk_out). The card clock can have different clock frequencies, since the card can be a low-speed card or a full-speed card. The Host Controller provides one clock signal (cclk_out).

The clock frequency of a card depends on the following clock control registers:

- Clock Divider register – Internal clock dividers are used to generate different clock frequencies required for card. The division factor for each clock divider can be programmed by writing to the Clock Divider register. A value of 0 represents a clock-divider bypass, a value of 1 represents a divide by 2.
- Clock Control register – cclk_out can be enabled or disabled for each card under the

following conditions:

- clk_enable – cclk_out for a card is enabled if the clk_enable bit for a card in the Clock Control register is programmed (set to 1) or disabled (set to 0).
- Low-power mode – Low-power mode of a card can be enabled by setting the low-power mode bit of the Clock Control register to 1. If low-power mode is enabled to save card power, the cclk_out is disabled when the card is idle for at least 8 card clock cycles. It is enabled when a new command is loaded and the command path goes to a non-idle state.

Additionally, cclk_out is disabled when an internal FIFO is full – card read (no more data can be received from card) – or when the FIFO is empty – card write (no data is available for transmission). This helps to avoid FIFO overrun and underrun conditions. It is used by the command and data path to qualify cclk_in for driving outputs and sampling inputs at the programmed clock frequency for the selected card, according to the Clock Divider and Clock Source register values.

Under the following conditions, the card clock is stopped or disabled, along with the active clk_en, for the selected card:

- Clock can be disabled by writing to Clock Enable register (clk_en bit = 1).
- If low-power mode is selected and card is idle, or not selected for 8 clocks.
- FIFO is full and data path cannot accept more data from the card and data transfer is incomplete –to avoid FIFO overrun.
- FIFO is empty and data path cannot transmit more data to the card and data transfer is incomplete – to avoid FIFO underrun.

27.3.2.6 Error Detection

- Response
  - Response timeout – Response expected with response start bit is not received within programmed number of clocks in timeout register.
  - Response CRC error – Response is expected and check response CRC requested; response CRC7 does not match with the internally-generated CRC7.
  - Response error – Response transmission bit is not 0, command index does not match with the command index of the send command, or response end bit is not 1.
- Data transmit
  - No CRC status – During a write data transfer, if the CRC status start bit is not received two clocks after the end bit of the data block is sent out, the data path does the following:
    - Signals no CRC status error to the BIU
    - Terminates further data transfer
    - Signals data transfer done to the BIU
  - Negative CRC – If the CRC status received after the write data block is negative (that is, not 010), a data CRC error is signaled to the BIU and further data transfer is continued.
  - Data starvation due to empty FIFO – If the FIFO becomes empty during a write data transmission, or if the card clock is stopped and the FIFO remains empty for data timeout clocks, then a data-starvation error is signaled to the BIU and the data path continues to wait for data in the FIFO.
- Data receive
  - Data timeout – During a read-data transfer, if the data start bit is not received before the number of clocks that were programmed in the timeout register, the data path does the following:
    - Signals data-timeout error to the BIU
    - Terminates further data transfer
    - Signals data transfer done to BIU
  - Data start bit error – During a 4-bit or 8-bit read-data transfer, if the all-bit data line does not have a start bit, the data path signals a data start bit error to the BIU and waits for a data timeout, after which it signals that the data transfer is done.
  - Data CRC error – During a read-data-block transfer, if the CRC16 received does not match with the internally generated CRC16, the data path signals a data CRC error to the BIU and continues further data transfer.

- Data end-bit error – During a read-data transfer, if the end bit of the received data is not 1, the data path signals an end-bit error to the BIU, terminates further data transfer, and signals to the BIU that the data transfer is done.
- Data starvation due to FIFO full – During a read data transmission and when the FIFO becomes full, the card clock is stopped. If the FIFO remains full for data timeout clocks, a data starvation error is signaled to the BIU (Data Starvation by Host Timeout bit is set in SDMMC_RINTSTS register) and the data path continues to wait for the FIFO to start to empty.

27.3.3 Internal Direct Memory Access Controller (IDMAC)

The Internal Direct Memory Access Controller (IDMAC) has a Control and Status Register (CSR) and a single Transmit/Receive engine, which transfers data from host memory to the device port and vice versa. The controller utilizes a descriptor to efficiently move data from source to destination with minimal Host CPU intervention. You can program the controller to interrupt the Host CPU in situations such as data Transmit and Receive transfer completion from the card, as well as other normal or error conditions.

The IDMAC and the Host driver communicate through a single data structure. CSR addresses 0x80 to 0x98 are reserved for host programming.

The IDMAC transfers the data received from the card to the Data Buffer in the Host memory, and it transfers Transmit data from the Data Buffer in the Host memory to the FIFO. Descriptors that reside in the Host memory act as pointers to these buffers.

A data buffer resides in physical memory space of the Host and consists of complete data or partial data. Buffers contain only data, while buffer status is maintained in the descriptor. Data chaining refers to data that spans multiple data buffers. However, a single descriptor cannot span multiple data.

A single descriptor is used for both reception and transmission. The base address of the list is written into Descriptor List Base Address Register (SDMMC_DBADDR @0x88). A descriptor list is forward linked. The Last Descriptor can point back to the first entry in order to create a ring structure. The descriptor list resides in the physical memory address space of the Host. Each descriptor can point to a maximum of two data buffers.

27.3.3.1 IDMAC CSR Access

When an IDMAC is introduced, an additional CSR space resides in the IDMAC that controls the IDMAC functionality. The host accesses the new CSR space in addition to the existing control register set in the BIU. The IDMAC CSR primarily contains descriptor information. For a write operation to the CSR, the respective CSR logic of the IDMAC and BIU decodes the address before accepting. For a read operation from the CSR, the appropriate CSR read path is enabled.

You can enable or disable the IDMAC operation by programming bit[25] in the SDMMC_CTRL register of the BIU. This allows the data transfer by accessing the slave interface on the AMBA bus if the IDMAC is present but disabled. When IDMAC is enabled, the FIFO cannot be accessed through the slave interface.

27.3.3.2 Descriptors

- Descriptor structures

The IDMAC uses these types of descriptor structures:

- Dual-Buffer Structure – The distance between two descriptors is determined by the Skip Length value programmed in the Descriptor Skip Length (DSL) field of the Bus Mode Register (SDMMC_BMOD @0x80).

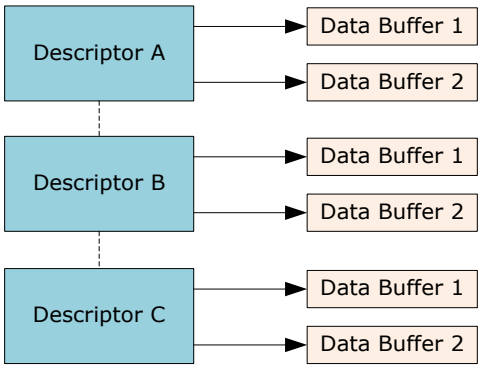


Fig. 27-6 Dual-Buffer Descriptor Structure

- Chain Structure – Each descriptor points to a unique buffer and the next descriptor.

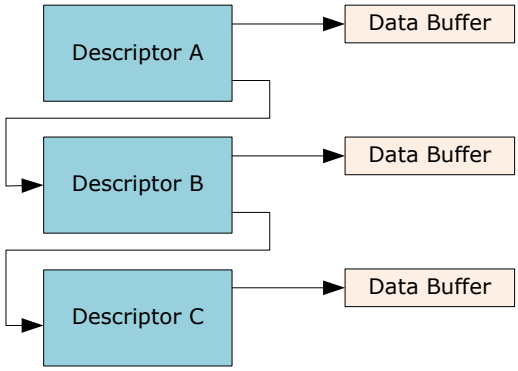


Fig. 27-7 Chain Descriptor Structure

- Descriptor formats

Following figure illustrates the internal formats of a descriptor. The descriptor addresses must be aligned to the bus width used for 32-bit AHB data buses. Each descriptor contains 16 bytes of control and status information. DES0 is a notation used to denote the [31:0] bits, DES1 to denote [63:32] bits, DES2 to denote [95:64] bits, DES3 to denote [127:96] bits.

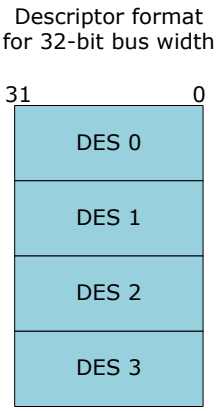


Fig. 27-8 Descriptor Formats for 32-bit AHB Address Bus Width

- The DES0 element in the IDMAC contains control and status information.

Table 27-4 Bits in IDMAC DES0 Element

Bit	Name	Description
31	OWN	When set, this bit indicates that the descriptor is owned by the IDMAC. When this bit is reset, it indicates that the descriptor is owned by the Host. The IDMAC clears this bit when it completes the data transfer.
30	Card Error Summary (CES)	These error bits indicate the status of the transaction to or from the card. These bits are also present in SDMMC_RINTSTS Indicates the logical OR of the following bits:

Bit	Name	Description
		● EBE: End Bit Error ● RTO: Response Time out ● RCRC: Response CRC ● SBE: Start Bit Error ● DRTO: Data Read Timeout ● DCRC: Data CRC for Receive ● RE: Response Error
29:6	Reserved	-
5	End of Ring (ER)	When set, this bit indicates that the descriptor list reached its final descriptor. The IDMAC returns to the base address of the list, creating a Descriptor Ring. This is meaningful for only a dual-buffer descriptor structure.
4	Second Address Chained (CH)	When set, this bit indicates that the second address in the descriptor is the Next Descriptor address rather than the second buffer address. When this bit is set, BS2 (DES1[25:13]) should be all zeros.
3	First Descriptor (FS)	When set, this bit indicates that this descriptor contains the first buffer of the data. If the size of the first buffer is 0, next Descriptor contains the beginning of the data.
2	Last Descriptor (LD)	This bit is associated with the last block of a DMA transfer. When set, the bit indicates that the buffers pointed to by this descriptor are the last buffers of the data. After this descriptor is completed, the remaining byte count is 0. In other words, after the descriptor with the LD bit set is completed, the remaining byte count should be 0.
1	Disable Interrupt on Completion (DIC)	When set, this bit will prevent the setting of the TI/RI bit of the IDMAC Status Register (IDSTS) for the data that ends in the buffer pointed to by this descriptor.
0	Reserved	-

■ The DES1 element contains the buffer size.

Table 27-5 Bits in IDMAC DES1 Element

Bit	Name	Description
31:26	Reserved	-
25:13	Buffer 2 Size (BS2)	These bits indicate the second data buffer byte size. The buffer size must be a multiple of 2, 4, or 8, depending upon the bus widths—16, 32, and 64 respectively. In the case where the buffer size is not a multiple of 2, 4, or 8, the resulting behavior is undefined. If this field is 0, the DMA ignores this buffer and proceeds to the next buffer in case of a dual-buffer structure. This field is not valid for chain structure; that is, if DES0[4] is set.
12:0	Buffer 1 Size (BS1)	Indicates the data buffer byte size, which must be a multiple of 2, 4, or 8 bytes, depending upon the bus widths—16, 32, and 64, respectively. In the case where the buffer size is not a multiple of 2, 4, or 8, the resulting behavior is undefined. This field should not be zero. Note: If there is only one buffer to be programmed, you need to use only the Buffer 1, and not Buffer 2.

■ The DES2 element contains the address pointer to the data buffer.

Table 27-6 Bits in IDMAC DES2 Element

Bit	Name	Description
31:26	Reserved	
25:13	Buffer 2 Size (BS2)	These bits indicate the second data buffer byte size. The buffer size must be a multiple of 2, 4, or 8, depending upon the bus

Bit	Name	Description
		widths—16, 32, and 64 respectively. In the case where the buffer size is not a multiple of 2, 4, or 8, the resulting behavior is undefined. If this field is 0, the DMA ignores this buffer and proceeds to the next buffer in case of a dual-buffer structure. This field is not valid for chain structure; that is, if DES0[4] is set.
12:0	Buffer 1 Size (BS1)	Indicates the data buffer byte size, which must be a multiple of 2, 4, or 8 bytes, depending upon the bus widths—16, 32, and 64, respectively. In the case where the buffer size is not a multiple of 2, 4, or 8, the resulting behavior is undefined. This field should not be zero. Note: If there is only one buffer to be programmed, you need to use only the Buffer 1, and not Buffer 2.

- The DES3 element contains the address pointer to the next descriptor if the present descriptor is not the last descriptor in a chained descriptor structure or the second buffer address for a dual-buffer structure.

Table 27-7 Bits in IDMAC DES3 Element

Bit	Name	Description
31:0	Buffer Address Pointer 2/ Next Descriptor Address (BAP2)	These bits indicate the physical address of the second buffer when the dual-buffer structure is used. If the Second Address Chained (DES0[4]) bit is set, then this address contains the pointer to the physical memory where the Next Descriptor is present. If this is not the last descriptor, then the Next Descriptor address pointer must be bus-width aligned.

27.3.3.3 Initialization

IDMAC initialization occurs as follows:

1) Write to IDMAC Bus Mode Register—SDMMC_BMOD to set Host bus access parameters.
2) Write to IDMAC Interrupt Enable Register—SDMMC_IDINTEN to mask unnecessary interrupt causes.
3) The software driver creates either the Transmit or the Receive descriptor list. Then it writes to IDMAC Descriptor List Base Address Register (SDMMC_DBADDR), providing the IDMAC with the starting address of the list.
4) The IDMAC engine attempts to acquire descriptors from the descriptor lists.

- Host Bus Burst Access

The IDMAC attempts to execute fixed-length burst transfers on the AHB Master interface if configured using the FB bit of the IDMAC Bus Mode register. The maximum burst length is indicated and limited by the PBL field. The descriptors are always accessed in the maximum possible burst-size for the 16-bytes to be read— 16*8/bus-width.

The IDMAC initiates a data transfer only when sufficient space to accommodate the configured burst is available in the FIFO or the number of bytes to the end of data, when less than the configured burst-length.

The IDMAC indicates the start address and the number of transfers required to the AHB Master Interface. When the AHB Interface is configured for fixed-length bursts, then it transfers data using the best combination of INCR4/8/16 and SINGLE transactions. Otherwise, in no fixed-length bursts, it transfers data using INCR (undefined length) and SINGLE transactions.

- Host Data Buffer Alignment

The Transmit and Receive data buffers in host memory must be aligned, depending on the data width.

- Buffer Size Calculations

The driver knows the amount of data to transmit or receive. For transmitting to the card, the IDMAC transfers the exact number of bytes to the FIFO, indicated by the buffer size field of DES1.

If a descriptor is not marked as last-LS bit of DES0-then the corresponding buffer(s) of the descriptor are full, and the amount of valid data in a buffer is accurately indicated by its

buffer size field. If a descriptor is marked as last, then the buffer cannot be full, as indicated by the buffer size in DES1. The driver is aware of the number of locations that are valid in this case.

- Transmission

IDMAC transmission occurs as follows:

1) The Host sets up the elements (DES0-DES3) for transmission and sets the OWN bit (DES0[31]). The Host also prepares the data buffer.
2) The Host programs the write data command in the SDMMC_CMD register in BIU.
3) The Host will also program the required transmit threshold level (TX_WMark field in SDMMC_FIFOTH register).
4) The IDMAC determines that a write data transfer needs to be done as a consequence of step 2.
5) The IDMAC engine fetches the descriptor and checks the OWN bit. If the OWN bit is not set, it means that the host owns the descriptor. In this case the IDMAC enters suspend state and asserts the Descriptor Unable interrupt in the SDMMC_IDSTS register. In such a case, the host needs to release the IDMAC by writing any value to the poll demand register.
6) It will then wait for Command Done (CD) bit and no errors from BIU which indicates that a transfer can be done.
7) The IDMAC engine will now wait for a DMA interface request from BIU. This request will be generated based on the programmed transmit threshold value. For the last bytes of data which can't be accessed using a burst, SINGLE transfers are performed on AHB Master Interface.
8) The IDMAC fetches the Transmit data from the data buffer in the Host memory and transfers to the FIFO for transmission to card.
9) When data spans across multiple descriptors, the IDMAC will fetch the next descriptor and continue with its operation with the next descriptor. The Last Descriptor bit in the descriptor indicates whether the data spans multiple descriptors or not.
10) When data transmission is complete, status information is updated in SDMMC_IDSTS register by setting Transmit Interrupt, if enabled. Also, the OWN bit is cleared by the IDMAC by performing a write transaction to DES0.

- Reception

IDMAC reception occurs as follows:

1) The Host sets up the element (DES0-DES3) for reception, sets the OWN (DES0[31]).
2) The Host programs the read data command in the SDMMC_CMD register in BIU.
3) The Host will program the required receive threshold level (RX_WMark field in FIFOTH register).
4) The IDMAC determines that a read data transfer needs to be done as a consequence of step 2.
5) The IDMAC engine fetches the descriptor and checks the OWN bit. If the OWN bit is not set, it means that the host owns the descriptor. In this case the DMA enters suspend state and asserts the Descriptor Unable interrupt in the SDMMC_IDSTS register. In such a case, the host needs to release the IDMAC by writing any value to the poll demand register.
6) It will then wait for Command Done (CD) bit and no errors from BIU which indicates that a transfer can be done.
7) The IDMAC engine will now wait for a DMA interface request from BIU. This request will be generated based on the programmed receive threshold value. For the last bytes of data which can't be accessed using a burst, SINGLE transfers are performed on AHB.
8) The IDMAC fetches the data from the FIFO and transfer to Host memory.
9) When data spans across multiple descriptors, the IDMAC will fetch the next descriptor and continue with its operation with the next descriptor. The Last Descriptor bit in the descriptor indicates whether the data spans multiple descriptors or not.
10) When data reception is complete, status information is updated in SDMMC_IDSTS register by setting Receive Interrupt, if enabled. Also, the OWN bit is cleared by the IDMAC by performing a write transaction to DES0.

- Interrupts

Interrupts can be generated as a result of various events. SDMMC_IDSTS register contains all the bits that might cause an interrupt. SDMMC_IDINTEN register contains an Enable bit for each of the events that can cause an interrupt.
There are two groups of summary interrupts-Normal and Abnormal-as outlined in SDMMC_IDSTS register. Interrupts are cleared by writing a 1 to the corresponding bit position. When all the enabled interrupts within a group are cleared, the corresponding summary bit is cleared. When both the summary bits are cleared, the interrupt signal dmac_intr_o is de-asserted.
Interrupts are not queued and if the interrupt event occurs before the driver has responded to it, no additional interrupts are generated. For example, Receive Interrupt—SDMMC_IDSTS[1] indicates that one or more data was transferred to the Host buffer.
An interrupt is generated only once for simultaneous, multiple events. The driver must scan SDMMC_IDSTS register for the interrupt cause.

27.3.4 Variable Delay/Clock Generation Unit

Variable delay mechanism for the cclk_in_drv is useful in order to meet a range of hold-time requirements across modes. Variable delay mechanism for the cclk_in_sample is mandatory and is required to achieve the correct sampling point for data.
The Clock Generation Unit (CLKGEN) includes Phase Shift Unit and Delay Line Unit.
The Phase Shift Unit can shift cclk_in_sample/cclk_in_drv by 0/90/180/270-degree relative to cclk_in. The Delay Line Unit can shift cclk_in_sample/cclk_in_drv step by step in the unit of delay element. The delay value range is 25ps~56ps for every delay element; the max delay element number is 256.
The architecture is as follows.

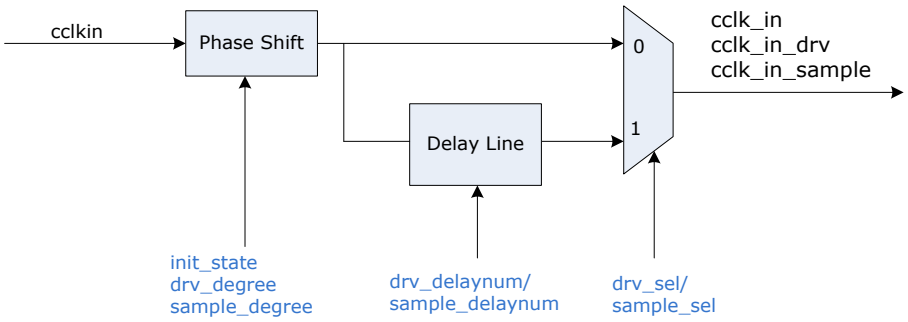


Fig. 27-9 Clock Generation Unit

27.4 Register Description

27.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
SDMMC_CTRL	0x0000	W	0x00000000	Control register
SDMMC_PWREN	0x0004	W	0x00000000	Power enable register
SDMMC_CLKDIV	0x0008	W	0x00000000	Clock divider register
SDMMC_CLKSRC	0x000c	W	0x00000000	SD clock source register
SDMMC_CLKENA	0x0010	W	0x00000000	Clock enable register
SDMMC_TMOUT	0x0014	W	0xFFFFFF40	Timeout register
SDMMC_CTYPE	0x0018	W	0x00000000	Card type register
SDMMC_BLKSIZ	0x001c	W	0x00000200	Block size register
SDMMC_BYTCNT	0x0020	W	0x00000200	Byte count register
SDMMC_INTMASK	0x0024	W	0x00000000	Interrupt mask register
SDMMC_CMDARG	0x0028	W	0x00000000	Command argument register
SDMMC_CMD	0x002c	W	0x20000000	Command register
SDMMC_RESP0	0x0030	W	0x00000000	Response register 0
SDMMC_RESP1	0x0034	W	0x00000000	Response register 1
SDMMC_RESP2	0x0038	W	0x00000000	Response register 2
SDMMC_RESP3	0x003c	W	0x00000000	Response register 3
SDMMC_MINTSTS	0x0040	W	0x00000000	Masked interrupt status register

Name	Offset	Size	Reset Value	Description
SDMMC_RINTSTS	0x0044	W	0x00000000	Raw interrupt status register
SDMMC_STATUS	0x0048	W	0x00000106	Status register
SDMMC_FIFOTH	0x004c	W	0x00FF0000	FIFO threshold register
SDMMC_CDETECT	0x0050	W	0x00000001	Card detect register
SDMMC_WRTPRT	0x0054	W	0x00000000	Write protect register
SDMMC_TCBCNT	0x005c	W	0x00000000	Transferred card byte count register
SDMMC_TBBCNT	0x0060	W	0x00000000	Transferred host to FIFO byte count register
SDMMC_DEBNCE	0x0064	W	0x00FFFFFF	Debounce count register
SDMMC_USRID	0x0068	W	0x00000000	User ID register
SDMMC_VERID	0x006c	W	0x5342270A	Version ID register
SDMMC_HCON	0x0070	W	0x04C434C1	Hardware configuration register
SDMMC_UHSREG	0x0074	W	0x00000000	UHS-1 control register
SDMMC_RSTN	0x0078	W	0x00000001	Hardware reset register
SDMMC_BMOD	0x0080	W	0x00000000	Bus mode register
SDMMC_PLDMND	0x0084	W	0x00000000	Poll demand register
SDMMC_DBADDR	0x0088	W	0x00000000	Descriptor list base address register
SDMMC_IDSTS	0x008c	W	0x00000000	Internal DMAC status register
SDMMC_IDINTEN	0x0090	W	0x00000000	Internal DMAC interrupt enable register
SDMMC_DSCADDR	0x0094	W	0x00000000	Current host descriptor address register
SDMMC_BUFADDR	0x0098	W	0x00000000	Current buffer descriptor address register
SDMMC_CARDTHRCTL	0x0100	W	0x00000000	Card threshold control register
SDMMC_BACKEND_POWER	0x0104	W	0x00000000	Back-end power register
SDMMC_EMMCDDR_REG	0x010c	W	0x00000000	eMMC4.5 DDR start bit detection control register
SDMMC_RDYINT_GEN	0x0120	W	0x00FF0000	Card ready interrupt generation control register
SDMMC_FIFO_BASE	0x0200	W	0x00000000	FIFO base address register

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

27.4.2 Detail Register Description

SDMMC_CTRL

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:26	RO	0x00	reserved
25	RW	0x0	use_internal_dmac Present only for the Internal DMAC configuration; else, it is reserved. 1'b0: The host performs data transfers through the slave interface 1'b1: Internal DMAC used for data transfer
24:12	RO	0x0000	reserved

Bit	Attr	Reset Value	Description
11	RW	0x0	ceata_device_interrupt_status 1'b0: Interrupts not enabled in CE-ATA device 1'b1: Interrupts are enabled in CE-ATA device Software should appropriately write to this bit after power-on reset or any other reset to CE-ATA device. After reset, usually CE-ATA device interrupt is disabled. If the host enables CE-ATA device interrupt, then software should set this bit.
10	RW	0x0	send_auto_stop_ccsd 1'b0: Clear bit if Host Controller does not reset the bit 1'b1: Send internally generated STOP after sending CCSD to CE-ATA device NOTE: Always set send_auto_stop_ccsd and send_ccsd bits together send_auto_stop_ccsd should not be set independent of send_ccsd. When set, the Host Controller automatically sends internally-generated STOP command (CMD12) to CE-ATA device. After sending internally-generated STOP command, Auto Command Done (ACD) bit in SDMMC_RINTSTS is set and generates interrupt to host if Auto Command Done interrupt is not masked. After sending the CCSD, the Host Controller automatically clears send_auto_stop_ccsd bit.
9	RW	0x0	send_ccsd 1'b0: Clear bit if Host Controller does not reset the bit 1'b1: Send Command Completion Signal Disable (CCSD) to CE-ATA device When set, the Host Controller sends CCSD to CE-ATA device. Software sets this bit only if current command is expecting CCS (that is, RW_BLK) and interrupts are enabled in CE-ATA device. Once the CCSD pattern is sent to device, the Host Controller automatically clears send_ccsd bit. It also sets Command Done (CD) bit in SDMMC_RINTSTS register and generates interrupt to host if Command Done interrupt is not masked. NOTE: Once send_ccsd bit is set, it takes two card clock cycles to drive the CCSD on the CMD line. Due to this, during the boundary conditions it may happen that CCSD is sent to the CE-ATA device, even if the device signalled CCS.
8	RW	0x0	abort_read_data 1'b0: No change 1'b1: After suspend command is issued during read-transfer, software polls card to find when suspend happened. Once suspend occurs, software sets bit to reset data state-machine, which is waiting for next block of data. Bit automatically clears once data state machine resets to idle. Used in SDIO card suspend sequence.
7	RW	0x0	send_irq_response 1'b0: No change 1'b1: Send auto IRQ response Bit automatically clears once response is sent. To wait for MMC card interrupts, software issues CMD40, and the Host Controller waits for interrupt response from MMC card. In meantime, if software wants the Controller to exit waiting for interrupt state, it can set this bit, at which time the Host Controller command state-machine sends CMD40 response on bus and returns to idle state.

Bit	Attr	Reset Value	Description
6	RW	0x0	read_wait 1'b0: Clear read wait 1'b1: Assert read wait For sending read-wait to SDIO cards.
5	RW	0x0	dma_enable 1'b0: Disable DMA transfer mode 1'b1: Enable DMA transfer mode Even when DMA mode is enabled, host can still push/pop data into or from FIFO; this should not happen during the normal operation. If there is simultaneous FIFO access from host/DMA, the data coherency is lost. Also, there is no arbitration inside the controller to prioritize simultaneous host/DMA access.
4	RW	0x0	int_enable Global interrupt enable/disable bit. 1'b0: Disable interrupts 1'b1: Enable interrupts The int port is 1 only when this bit is 1 and one or more unmasked interrupts are set.
3	RO	0x0	reserved
2	WO	0x0	dma_reset 1'b0: No change 1'b1: Reset internal DMA interface control logic To reset DMA interface, firmware should set bit to 1. This bit is auto-cleared after two AHB clocks.
1	WO	0x0	fifo_reset 1'b0: No change 1'b1: Reset to data FIFO to reset FIFO pointers To reset FIFO, firmware should set bit to 1. This bit is auto-cleared after completion of reset operation.
0	WO	0x0	controller_reset 1'b0: No change 1'b1: Reset Host Controller To reset Host Controller, firmware should set bit to 1. This bit is auto-cleared after two AHB and two cclk_in clock cycles. This resets: a. BIU/CIU interface b. CIU and state machines c. abort_read_data, send_irq_response, and read_wait bits of SDMMC_CTRL register d. start_cmd bit of SDMMC_CMD register Does not affect any registers or DMA interface, or FIFO or controller interrupts.

SDMMC_PWREN

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RW	0x0	power_enable Power on/off switch for the card. Once power is turned on, firmware should wait for regulator/switch ramp-up time before trying to initialize card. 1'b0: Power off 1'b1: Power on Bit values output to card_power_en port.

SDMMC_CLKDIV

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x00	clk_divider0 Clock divider-0 value. Clock division is 2*n. For example, value of 0 means divide by 2*0 = 0 (no division, bypass), value of 1 means divide by 2*1 = 2, and so on. The recommended value is 0 or 1.

SDMMC_CLKSRC

Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved
1:0	RW	0x0	clk_source Clock divider source. 2'b00: Clock divider 0 The cclk_out is always from clock divider 0, and this register is not implemented.

SDMMC_CLKENA

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:17	RO	0x0000	reserved
16	RW	0x0	cclk_low_power Low-power control for SD card clock and MMC card clock supported. 1'b0: Non-low-power mode 1'b1: Low-power mode Stop clock when card in IDLE (should be normally set to only MMC and SD memory cards; for SDIO cards, if interrupts must be detected, clock should not be stopped).
15:1	RO	0x0000	reserved
0	RW	0x0	cclk_enable Clock-enable control for SD card clock and MMC card clock supported. 1'b0: Clock disabled 1'b1: Clock enabled

SDMMC_TMOUT

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:8	RW	0xffffff	data_timeout Value for card data read timeout; same value also used for data starvation by host timeout. Value is in number of card output clock. Note: The software timer should be used if the timeout value is in the order of 100 ms. In this case, read data timeout interrupt needs to be disabled.
7:0	RW	0x40	response_timeout Response timeout value. Value is in number of card output clock cclk_out.

SDMMC_CTYPE

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:17	RO	0x0000	reserved
16	RW	0x0	card_width_8 Indicates if card is 8-bit. 1'b0: Non 8-bit mode 1'b1: 8-bit mode
15:1	RO	0x0000	reserved
0	RW	0x0	card_width Indicates if card is 1-bit or 4-bit. 1'b0: 1-bit mode 1'b1: 4-bit mode

SDMMC_BLKSIZ

Address: Operational Base + offset (0x001c)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15:0	RW	0x0200	block_size Block size

SDMMC_BYTCNT

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000200	byte_count Number of bytes to be transferred; should be integer multiple of block size for block transfers. For undefined number of byte transfers, byte count should be set to 0. When byte count is set to 0, it is responsibility of host to explicitly send stop/abort command to terminate data transfer.

SDMMC_INTMASK

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:25	RO	0x00	reserved
24	RW	0x0	sdio_int_mask 1'b0: SDIO interrupt not masked 1'b1: SDIO interrupt masked
23:17	RO	0x00	reserved
16	RW	0x0	data_nobusy_int_mask 1'b0: Data no busy interrupt not masked 1'b1: Data no busy interrupt masked

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	int_mask Bits used to mask unwanted interrupts. Value of 0 masks interrupt; value of 1 enables interrupt. bit 15: End-bit error (read)/Write no CRC (EBE) bit 14: Auto command done (ACD) bit 13: Start-bit error (SBE) bit 12: Hardware locked write error (HLE) bit 11: FIFO underrun/overrun error (FRUN) bit 10: Data starvation-by-host timeout (HTO) /Volt_switch_int bit 9: Data read timeout (DRTO) bit 8: Response timeout (RTO) bit 7: Data CRC error (DCRC) bit 6: Response CRC error (RCRC) bit 5: Receive FIFO data request (RXDR) bit 4: Transmit FIFO data request (TXDR) bit 3: Data transfer over (DTO) bit 2: Command done (CD) bit 1: Response error (RE) bit 0: Card detect (CD)

SDMMC_CMDARG

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	cmd_arg Value indicates command argument to be passed to card.

SDMMC_CMD

Address: Operational Base + offset (0x002c)

Bit	Attr	Reset Value	Description
31	RW	0x0	start_cmd Start command. Once command is taken by CIU, bit is cleared. When bit is set, host should not attempt to write to any command registers. If write is attempted, hardware lock error is set in raw interrupt register. Once command is sent and response is received from SD/MMC cards, Command Done bit is set in raw interrupt register.
30	RO	0x0	reserved
29	RW	0x1	use_hold_reg Use hold register. 1'b0: CMD and DATA sent to card bypassing hold register 1'b1: CMD and DATA sent to card through the hold register
28	RW	0x0	volt_switch Voltage switch bit. 1'b0: No voltage switching 1'b1: Voltage switching enabled; must be set for CMD11 only.
27	RW	0x0	boot_mode Boot mode selection. 1'b0: Mandatory boot operation 1'b1: Alternate boot operation
26	RW	0x0	disable_boot Disable boot. When software sets this bit along with start_cmd, CIU terminates the boot operation. Do not set disable_boot and enable_boot together.

Bit	Attr	Reset Value	Description
25	RW	0x0	expect_boot_ack Expect boot acknowledge. When software sets this bit along with enable_boot, CIU expects a boot acknowledge start pattern of 0-1-0 from the selected card.
24	RW	0x0	enable_boot Enable boot. This bit should be set only for mandatory boot mode. When Software sets this bit along with start_cmd, CIU starts the boot sequence for the corresponding card by asserting the CMD line low. Do not set disable_boot and enable_boot together.
23	RW	0x0	ccs_expected 1'b0: Interrupts are not enabled in CE-ATA device or command does not expect CCS from device 1'b1: Interrupts are enabled in CE-ATA device and RW_BLK command expects command completion signal from CE-ATA device If the command expects command completion signal (CCS) from the CE-ATA device, the software should set this control bit. The Host Controller sets data transfer over (DTO) bit in SDMMC_RINTSTS register and generates interrupt to host if data transfer over interrupt is not masked.
22	RW	0x0	read_ceata_device 1'b0: Host is not performing read access towards CE-ATA device 1'b1: Host is performing read access towards CE-ATA device Software should set this bit to indicate that CE-ATA device is being accessed for read transfer. This bit is used to disable read data timeout indication while performing CE-ATA read transfers. Maximum value of I/O transmission delay can be no less than 10 seconds. The Host Controller should not indicate read data timeout while waiting for data from CE-ATA device.
21	RW	0x0	update_clock_regs_only 1'b0: Normal command sequence 1'b1: Do not send commands, just update clock register value into card clock domain. Following register values transferred into card clock domain: SDMMC_CLKDIV, SDMMC_CLRSRC, SDMMC_CLKENA. Changes card clocks (change frequency, truncate off or on, and set low-frequency mode); provided in order to change clock frequency or stop clock without having to send command to cards. During normal command sequence, when update_clock_regs_only = 0, following control registers are transferred from BIU to CIU: SDMMC_CMD, SDMMC_CMDARG, SDMMC_TMOUT, SDMMC_CTYPE, SDMMC_BLKSIZ, SDMMC_BYTCNT. CIU uses new register values for new command sequence to card. When bit is set, there are no Command Done interrupts because no command is sent to SD_MMC_CEATA cards.
20:16	RO	0x00	reserved

Bit	Attr	Reset Value	Description
15	RW	0x0	send_initialization 1'b0: Do not send initialization sequence (80 clocks of 1) before sending this command 1'b1: Send initialization sequence before sending this command After power on, 80 clocks must be sent to card for initialization before sending any commands to card. Bit should be set while sending first command to card so that controller will initialize clocks before sending command to card. This bit should not be set for either of the boot modes (alternate or mandatory).
14	RW	0x0	stop_abort_cmd 1'b0: Neither stop nor abort command to stop current data transfer in progress. If abort is sent to function-number currently selected or not in data-transfer mode, then bit should be set to 0. 1'b1: Stop or abort command intended to stop current data transfer in progress. When open-ended or predefined data transfer is in progress, and host issues stop or abort command to stop data transfer, bit should be set so that command/data state-machines of CIU can return correctly to idle state. This is also applicable for Boot mode transfers. To Abort boot mode, this bit should be set along with SDMMC_CMD[26]=disable_boot.
13	RW	0x0	wait_prvdata_complete 1'b0: Send command at once, even if previous data transfer has not completed 1'b1: Wait for previous data transfer completion before sending command The wait_prvdata_complete=0 option typically used to query status of card during data transfer or to stop current data transfer; card_number should be same as in previous command.
12	RW	0x0	send_auto_stop 1'b0: No stop command sent at end of data transfer 1'b1: Send stop command at end of data transfer When set, the Host Controller sends stop command to card at end of data transfer. a. When send_auto_stop bit should be set, since some data transfers do not need explicit stop commands b. Open-ended transfers that software should explicitly send to stop command Additionally, when "resume" is sent to resume-suspended memory access of SD-Combo card, bit should be set correctly if suspended data transfer needs send_auto_stop. Don't care if no data expected from card.
11	RW	0x0	transfer_mode 1'b0: Block data transfer command 1'b1: Stream data transfer command Don't care if no data expected.
10	RW	0x0	wr 1'b0: Read from card 1'b1: Write to card Don't care if no data expected from card.
9	RW	0x0	data_expected 1'b0: No data transfer expected (read/write) 1'b1: Data transfer expected (read/write)

Bit	Attr	Reset Value	Description
8	RW	0x0	check_response_crc 1'b0: Do not check response CRC 1'b1: Check response CRC Some of command responses do not return valid CRC bits. Software should disable CRC checks for those commands in order to disable CRC checking by controller.
7	RW	0x0	response_length 1'b0: Short response expected from card 1'b1: Long response expected from card
6	RW	0x0	response_expect 1'b0: No response expected from card 1'b1: Response expected from card
5:0	RW	0x00	cmd_index Command index

SDMMC_RESP0

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	response0 Bit[31:0] of response

SDMMC_RESP1

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	response1 Register represents bit[63:32] of long response. When CIU sends auto-stop command, then response is saved in register. Response for previous command sent by host is still preserved in response 0 register. Additional auto-stop issued only for data transfer commands, and response type is always "short" for them.

SDMMC_RESP2

Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	response2 Bit[95:64] of long response.

SDMMC_RESP3

Address: Operational Base + offset (0x003c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	response3 Bit[127:96] of long response.

SDMMC_MINTSTS

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:25	RO	0x00	reserved
24	RO	0x0	sdio_interrupt SDIO interrupt status when sdio_int_mask is set.
23:17	RO	0x00	reserved
16	RW	0x0	data_nobusy_int_status Data no busy interrupt status when data_nobusy_int_mask is set

Bit	Attr	Reset Value	Description
15:0	RO	0x0000	int_status Interrupt enabled only if corresponding bit in interrupt mask register is set. bit 15: End-bit error (read)/Write no CRC (EBE) bit 14: Auto command done (ACD) bit 13: Start-bit error (SBE) bit 12: Hardware locked write error (HLE) bit 11: FIFO underrun/overrun error (FRUN) bit 10: Data starvation-by-host timeout (HTO) /Volt_switch_int bit 9: Data read timeout (DRTO) bit 8: Response timeout (RTO) bit 7: Data CRC error (DCRC) bit 6: Response CRC error (RCRC) bit 5: Receive FIFO data request (RXDR) bit 4: Transmit FIFO data request (TXDR) bit 3: Data transfer over (DTO) bit 2: Command done (CD) bit 1: Response error (RE) bit 0: Card detect (CD)

SDMMC_RINTSTS

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:25	RO	0x00	reserved
24	WO	0x0	sdio_interrupt Raw SDIO interrupt status. Write value of 1 clears this bit, and value of 0 leaves bit intact.
23:17	RO	0x00	reserved
16	WO	0x0	data_nobusy_int_status Raw data no busy interrupt status. Write value of 1 clears this bit, and value of 0 leaves bit intact.
15:0	WO	0x0000	int_status Raw interrupt status. Writes to bits clear status bit. Write value of 1 clears status bit, and value of 0 leaves bit intact. bit 15: End-bit error (read)/Write no CRC (EBE) bit 14: Auto command done (ACD) bit 13: Start-bit error (SBE) bit 12: Hardware locked write error (HLE) bit 11: FIFO underrun/overrun error (FRUN) bit 10: Data starvation-by-host timeout (HTO) /Volt_switch_int bit 9: Data read timeout (DRTO) bit 8: Response timeout (RTO) bit 7: Data CRC error (DCRC) bit 6: Response CRC error (RCRC) bit 5: Receive FIFO data request (RXDR) bit 4: Transmit FIFO data request (TXDR) bit 3: Data transfer over (DTO) bit 2: Command done (CD) bit 1: Response error (RE) bit 0: Card detect (CD)

SDMMC_STATUS

Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31	RO	0x0	dma_req DMA request signal state
30	RO	0x0	dma_ack DMA acknowledge signal state
29:17	RO	0x0000	fifo_count Number of filled locations in FIFO
16:11	RO	0x00	response_index Index of previous response, including any auto-stop sent by core.
10	RO	0x0	data_state_mc_busy Data transmit or receive state-machine is busy.
9	RO	0x0	data_busy Inverted version of raw selected card_data[0]. 1'b0: Card data not busy 1'b1: Card data busy
8	RO	0x1	data_3_status Raw selected card_data[3]; checks whether card is present. 1'b0: Card not present 1'b1: Card present
7:4	RO	0x0	command_fsm_states Command FSM states: 4'h0: Idle 4'h1: Send init sequence 4'h2: Tx cmd start bit 4'h3: Tx cmd tx bit 4'h4: Tx cmd index + arg 4'h5: Tx cmd crc7 4'h6: Tx cmd end bit 4'h7: Tx resp start bit 4'h8: Rx resp IRQ response 4'h9: Rx resp tx bit 4'ha: Rx resp cmd idx 4'hb: Rx resp data 4'hc: Rx resp crc7 4'hd: Rx resp end bit 4'he: Cmd path wait NCC 4'hf: Wait; CMD-to-response turnaround The command FSM state is represented using 19 bits. The SDMMC_STATUS register[7:4] has 4 bits to represent the command FSM states. Using these 4 bits, only 16 states can be represented. Thus three states cannot be represented in the SDMMC_STATUS[7:4] register. The three states that are not represented in the SDMMC_STATUS register[7:4] are: Bit 16: Wait for CCS Bit 17: Send CCSD Bit 18: Boot Mode Due to this, while command FSM is in "Wait for CCS state" or "Send CCSD" or "Boot Mode", the SDMMC_STATUS register indicates status as 0 for the bit field [7:4].
3	RO	0x0	fifo_full FIFO is full status
2	RO	0x1	fifo_empty FIFO is empty status

Bit	Attr	Reset Value	Description
1	RO	0x1	fifo_tx_watermark FIFO reached Transmit watermark level; not qualified with data transfer.
0	RO	0x0	fifo_rx_watermark FIFO reached Receive watermark level; not qualified with data transfer.

SDMMC_FIFOTH

Address: Operational Base + offset (0x004c)

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30:28	RW	0x0	dma_mutiple_transaction_size Burst size of multiple transaction; should be programmed same as DMA controller multiple-transaction-size SRC/DEST_MSIZE. 3'b000: 1 transfers 3'b001: 4 transfers 3'b010: 8 transfers 3'b011: 16 transfers 3'b100: 32 transfers 3'b101: 64 transfers 3'b110: 128 transfers 3'b111: 256 transfers The unit for transfer is 32bits.
27:16	RW	0x0ff	rx_wmark FIFO threshold watermark level when receiving data to card. When FIFO data count reaches greater than this number, DMA/FIFO request is raised. During end of packet, request is generated regardless of threshold programming in order to complete any remaining data. In non-DMA mode, when receiver FIFO threshold (RXDR) interrupt is enabled, then interrupt is generated instead of DMA request. During end of packet, interrupt is not generated if threshold programming is larger than any remaining data. It is responsibility of host to read remaining bytes on seeing Data Transfer Done interrupt. In DMA mode, at end of packet, even if remaining bytes are less than threshold, DMA request does single transfers to flush out any remaining bytes before Data Transfer Done interrupt is set. 12 bits-1 bit less than FIFO-count of status register, which is 13 bits. Limitation: rx_wmark <= FIFO_DEPTH-2 Recommended: (FIFO_DEPTH/2) - 1; (means greater than (FIFO_DEPTH/2) - 1) NOTE: In DMA mode during CCS time-out, the DMA does not generate the request at the end of packet, even if remaining bytes are less than threshold. In this case, there will be some data left in the FIFO. It is the responsibility of the application to reset the FIFO after the CCS timeout.
15:12	RO	0x0	reserved

Bit	Attr	Reset Value	Description
11:0	RW	0x000	tx_wmark FIFO threshold watermark level when transmitting data to card. When FIFO data count is less than or equal to this number, DMA/FIFO request is raised. If Interrupt is enabled, then interrupt occurs. During end of packet, request or interrupt is generated, regardless of threshold programming. In non-DMA mode, when transmit FIFO threshold (TXDR) interrupt is enabled, then interrupt is generated instead of DMA request. During end of packet, on last interrupt, host is responsible for filling FIFO with only required remaining bytes (not before FIFO is full or after CIU completes data transfers, because FIFO may not be empty). In DMA mode, at end of packet, if last transfer is less than burst size, DMA controller does single cycles until required bytes are transferred. 12 bits -1 bit less than FIFO-count of status register, which is 13 bits. Limitation: tx_wmark >= 1; Recommended: FIFO_DEPTH/2; (means less than or equal to FIFO_DEPTH/2)

SDMMC_CDETECT

Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RO	0x1	card_detect_n Value on card_detect_n input port. 1'b0: Represents presence of card 1'b1: Represents absence of card

SDMMC_WRTPRT

Address: Operational Base + offset (0x0054)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RW	0x0	write_protect Value on card_write_prt input port. 1 represents write protection.

SDMMC_TCBCNT

Address: Operational Base + offset (0x005c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	trans_card_byte_count Number of bytes transferred by CIU unit to card.

SDMMC_TBBCNT

Address: Operational Base + offset (0x0060)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	trans_fifo_byte_count Number of bytes transferred between host/DMA memory and BIU FIFO.

SDMMC_DEBNCE

Address: Operational Base + offset (0x0064)

Bit	Attr	Reset Value	Description
31:24	RO	0x00	reserved

Bit	Attr	Reset Value	Description
23:0	RW	0xffffff	debounce_count Number of host clock used by debounce filter logic; typical debounce time is 5-25 ms.

SDMMC_USRID

Address: Operational Base + offset (0x0068)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	usrid User identification register

SDMMC_VERID

Address: Operational Base + offset (0x006c)

Bit	Attr	Reset Value	Description
31:0	RO	0x5342270a	verid Version identification register

SDMMC_HCON

Address: Operational Base + offset (0x0070)

Bit	Attr	Reset Value	Description
31:27	RO	0x00	reserved
26	RO	0x1	area_optimized 1'b0: No area optimization 1'b1: Area optimization
25:24	RO	0x0	num_clk_div divider number-1.
23	RO	0x1	set_clk_false_path 1'b0: No false path 1'b1: False path set
22	RO	0x1	impl_hold_reg 1'b0: No hold register 1'b1: Hold register
21	RO	0x0	fifo_ram_inside 1'b0: Outside 1'b1: Inside
20:18	RO	0x1	ge_dma_data_width 3'b000: 16 bits 3'b001: 32 bits 3'b010: 64 bits others: Reserved
17:16	RO	0x0	dma_interface 2'b00: None 2'b01: DW_DMA 2'b10: GENERIC_DMA 2'b11: NON-DW-DMA
15:10	RO	0x1f	h_addr_width 6'h8: 9 bits 6'h9: 10 bits ..... 6'h1f: 32 bits others: Reserved
9:7	RO	0x1	h_data_width 3'b000: 16 bits 3'b001: 32 bits 3'b010: 64 bits others: Reserved

Bit	Attr	Reset Value	Description
6	RO	0x1	h_bus_type 1'b0: APB 1'b1: AHB
5:1	RO	0x00	card_num Card number -1.
0	RO	0x1	card_type Card type. 1'b0: MMC_ONLY 1'b1: SD_MMC

SDMMC_UHSREG

Address: Operational Base + offset (0x0074)

Bit	Attr	Reset Value	Description
31:17	RO	0x0000	reserved
16	RW	0x0	ddr_reg DDR mode. These bits indicate DDR mode of operation to the core for the data transfer. 1'b0: Non-DDR mode 1'b1: DDR mode
15:0	RO	0x0000	reserved

SDMMC_RSTN

Address: Operational Base + offset (0x0078)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RW	0x1	card_reset Hardware reset. 1'b0: Active mode 1'b1: Reset These bits cause the cards to enter pre-idle state, which requires them to be re-initialized.

SDMMC_BMOD

Address: Operational Base + offset (0x0080)

Bit	Attr	Reset Value	Description
31:11	RO	0x000000	reserved

Bit	Attr	Reset Value	Description
10:8	RO	0x0	pbl Programmable burst length. These bits indicate the maximum number of beats to be performed in one IDMAC transaction. The IDMAC will always attempt to burst as specified in PBL each time it starts a Burst transfer on the host bus. The permissible values are 1, 4, 8, 16, 32, 64, 128 and 256. This value is the mirror of MSIZE of SDMMC_FIFOTH register. In order to change this value, write the required value to SDMMC_FIFOTH register. This is an encode value as follows. 3'b000: 1 transfers 3'b001: 4 transfers 3'b010: 8 transfers 3'b011: 16 transfers 3'b100: 32 transfers 3'b101: 64 transfers 3'b110: 128 transfers 3'b111: 256 transfers Transfer unit is 32 bits. PBL is a read-only value and is applicable only for data access; it does not apply to descriptor accesses.
7	RW	0x0	de IDMAC enable. When set, the IDMAC is enabled.
6:2	RW	0x00	dsl Descriptor skip length. Specifies the number of word to skip between two unchained descriptors. This is applicable only for dual buffer structure.
1	RW	0x0	fb Fixed burst. Controls whether the AHB Master interface performs fixed burst transfers or not. When set, the AHB will use only SINGLE, INCR4, INCR8 or INCR16 during start of normal burst transfers. When reset, the AHB will use SINGLE and INCR burst transfer operations.
0	RW	0x0	swr Software reset. When set, the DMA Controller resets all its internal registers. It is automatically cleared after 1 clock cycle.

SDMMC_PLDMND

Address: Operational Base + offset (0x0084)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	pd Poll demand. If the OWN bit of a descriptor is not set, the FSM goes to the suspend state. The host needs to write any value into this register for the IDMAC FSM to resume normal descriptor fetch operation.

SDMMC_DBADDR

Address: Operational Base + offset (0x0088)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sbl Start of descriptor list. Contains the base address of the first descriptor. The LSB bits[1:0] are ignored and taken as all-zero by the IDMAC internally. Hence these LSB bits are read-only.

SDMMC_IDSTS

Address: Operational Base + offset (0x008c)

Bit	Attr	Reset Value	Description
31:17	RO	0x0000	reserved
16:13	RO	0x0	fsm DMAC FSM present state. 4'h0: DMA_IDLE 4'h1: DMA_SUSPEND 4'h2: DESC_RD 4'h3: DESC_CHK 4'h4: DMA_RD_REQ_WAI 4'h5: DMA_WR_REQ_WAI 4'h6: DMA_RD 4'h7: DMA_WR 4'h8: DESC_CLOSE Others: Reserved
12:10	RO	0x0	eb Error bits. Indicates the type of error that caused a bus error. Valid only with fatal bus. 3'h1: Host abort received during transmission 3'h2: Host abort received during reception Others: Reserved
9	RW	0x0	ais Abnormal interrupt summary. Logical OR of the following: SDMMC_IDSTS[2] fatal bus interrupt SDMMC_IDSTS[4] du bit interrupt Only unmasked bits affect this bit. This is a sticky bit and must be cleared each time a corresponding bit that causes ais to be set is cleared. Writing a 1 clears this bit.
8	RW	0x0	nis Normal interrupt summary. Logical OR of the following: SDMMC_IDSTS[0] transmit interrupt SDMMC_IDSTS[1] receive interrupt Only unmasked bits affect this bit. This is a sticky bit and must be cleared each time a corresponding bit that causes nis to be set is cleared. Writing a 1 clears this bit.
7:6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5	RW	0x0	ces Card error summary. Indicates the status of the transaction to/from the card; also present in SDMMC_RINTSTS. Indicates the logical OR of the following bits: EBE: End Bit Error RTO: Response Timeout/Boot Ack Timeout RCRC: Response CRC SBE: Start Bit Error DRTO: Data Read Timeout/BDS timeout DCRC: Data CRC for Receive RE: Response Error Writing a 1 clears this bit. The abort condition of the IDMAC depends on the setting of this CES bit. If the CES bit is enabled, then the IDMAC aborts on a "response error"; however, it will not abort if the CES bit is cleared.
4	RW	0x0	dui Descriptor unavailable interrupt. This bit is set when the descriptor is unavailable due to OWN bit = 0 (DES0[31] =0). Writing a 1 clears this bit.
3	RO	0x0	reserved
2	RW	0x0	fbe Fatal bus error interrupt. Indicates that a bus error occurred (SDMMC_IDSTS[12:10]). When this bit is set, the DMA disables all its bus accesses. Writing a 1 clears this bit.
1	RW	0x0	ri Receive interrupt. Indicates the completion of data reception for a descriptor. Writing a 1 clears this bit.
0	RW	0x0	ti Transmit interrupt. Indicates that data transmission is finished for a descriptor. Writing 1 clears this bit.

SDMMC_IDINTEN

Address: Operational Base + offset (0x0090)

Bit	Attr	Reset Value	Description
31:10	RO	0x000000	reserved
9	RW	0x0	ai Abnormal interrupt summary enable. When set, an abnormal interrupt is enabled. This bit enables the following bits: SDMMC_IDINTEN[2] fatal bus error interrupt SDMMC_IDINTEN[4] du interrupt
8	RW	0x0	ni Normal interrupt summary enable. When set, a normal interrupt is enabled. When reset, a normal interrupt is disabled. This bit enables the following bits: SDMMC_IDINTEN[0] transmit interrupt SDMMC_IDINTEN[1] receive interrupt
7:6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5	RW	0x0	ces Card error summary interrupt enable. When set, it enables the card interrupt summary.
4	RW	0x0	du Descriptor unavailable interrupt. When set along with abnormal interrupt summary enable, the du interrupt is enabled.
3	RO	0x0	reserved
2	RW	0x0	fbe Fatal bus error enable. When set with abnormal interrupt summary enable, the fatal bus error interrupt is enabled. When reset, fatal bus error enable interrupt is disabled.
1	RW	0x0	ri Receive interrupt enable. When set with normal interrupt summary enable, receive interrupt is enabled. When reset, receive interrupt is disabled.
0	RW	0x0	ti Transmit interrupt enable. When set with normal interrupt summary enable, transmit interrupt is enabled. When reset, transmit interrupt is disabled.

SDMMC_DSCADDR

Address: Operational Base + offset (0x0094)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	hda Host descriptor address pointer. This register points to the start address of the current descriptor read by the IDMAC. Cleared on reset. Pointer updated by IDMAC during operation.

SDMMC_BUFADDR

Address: Operational Base + offset (0x0098)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	hba Host buffer address pointer. This register points to the current data buffer address being accessed by the IDMAC. Cleared on Reset. Pointer updated by IDMAC during operation.

SDMMC_CARDTHRCTL

Address: Operational Base + offset (0x0100)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	card_rd_thres Card read threshold size
15:2	RO	0x0000	reserved

Bit	Attr	Reset Value	Description
1	RW	0x0	busy_clr_int_en Busy clear interrupt. 1'b0: Busy clear interrupt disabled 1'b1: Busy clear interrupt enabled Note: The application can disable this feature if it does not want to wait for a busy clear interrupt. For example, in a multi-card scenario, the application can switch to the other card without waiting for a busy to be completed. In such cases, the application can use the polling method to determine the status of busy. By default this feature is disabled and backward-compatible to the legacy drivers where polling is used.
0	RW	0x0	card_rd_thres_en Card read threshold enable. 1'b0: Card read threshold disabled 1'b1: Card read threshold enabled. The host initiates read transfer only if zcard_rd_thres amount of space is available in receive FIFO.

SDMMC_BACKEND_POWER

Address: Operational Base + offset (0x0104)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RW	0x0	back_end_power Back end power. 1'b0: Off; Reset 1'b1: Back-end power supplied to card application

SDMMC_EMMCDDR_REG

Address: Operational Base + offset (0x010c)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RW	0x0	half_start_bit Control for start bit detection mechanism inside Host Controller based on duration of start bit. For eMMC 4.5, start bit can be: 1'b0: Full cycle (half_start_bit=0) 1'b1: Less than one full cycle (half_start_bit=1) Set half_start_bit=1 for eMMC 4.5 and above; set to 0 for SD applications.

SDMMC_RDYINT_GEN

Address: Operational Base + offset (0x0120)

Bit	Attr	Reset Value	Description
31:25	RO	0x00	reserved
24	RO	0x0	rdyint_cnt_finish Counter finish indication. When high, it indicates that the rdyint counter is finished.
23:16	RO	0xff	rdyint_cnt_status Couner status, reflect internal counter value.
15:9	RO	0x00	reserved

Bit	Attr	Reset Value	Description
8	RW	0x0	rdyint_gen_working Working indication for rdyint generator. When high, Host Controller start to count and generate one rdyint trigger. After the rdyint trigger is generated, this bit will be set to 0 by Host Controller. So software should set it to 1 before detecting next interrupt.
7:0	RW	0x00	rdyint_gen_maxval Max counter value to detect cdata_in0 high value for generating rdyint, based on internal clock frequency.

SDMMC_FIFO_BASE

Address: Operational Base + offset (0x0200)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	fifo_base_addr FIFO base address

27.5 Interface Description

27.5.1 SDMMC Interface Description

Table 27-8 SDMMC Interface Description

Module Pin	Direction	PIN Name	IOMUX Setting
sdmmc_cclk	O	SDMMC0_CLK/UART3_RTSN_M1/RISC-V_JTAG_TDO/GPIO1_B0_u	GRF_GPIO1B_IOMUX_L[3:0]=4'h1
sdmmc_ccmd	I/O	SDMMC0_CMD/UART3_CTSN_M1/RISC-V_JTAG_TDI/GPIO1_B1_u	GRF_GPIO1B_IOMUX_L[7:4]=4'h1
sdmmc_cdata0	I/O	SDMMC0_D0/TEST_CLK1_OUT/UART2_RX_M0/GPIO1_A4_u	GRF_GPIO1A_IOMUX_H[3:0]=4'h1
sdmmc_cdata1	I/O	SDMMC0_D1/TEST_CLK0_OUT/UART2_TX_M0/RISC-V_JTAG_TRSTn/GPIO1_A5_u	GRF_GPIO1A_IOMUX_H[7:4]=4'h1
sdmmc_cdata2	I/O	SDMMC0_D2/UART3_RX_M1/A7_JTAG_TCK_M0/RISC-V_JTAG_TCK/GPIO1_A6_u	GRF_GPIO1A_IOMUX_H[11:8]=4'h1
sdmmc_cdata3	I/O	SDMMC0_D3/UART3_TX_M1/A7_JTAG_TMS_M0/RISC-V_JTAG_TMS/GPIO1_A7_u	GRF_GPIO1A_IOMUX_H[15:12]=4'h1
sdmmc_cdetn	I	SDMMC0_DET/GPIO0_A3_u	GRF_GPIO0A_IOMUX_L[15:12]=4'h1
sdmmc_pwr	O	SDMMC0_PWR/UART1_RTSN_M0/PWM2_M0/GPIO0_C0_d	GRF_GPIO0C_IOMUX_L[3:0]=4'h1

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

27.5.2 SDIO Interface Description

Table 27-9 SDIO Interface Description

Module Pin	Direction	PIN Name	IOMUX Setting
sdio_cclk	O	SDMMC1_CLK/GPIO1_B2_d	GRF_GPIO1B_IOMUX_L[11:8]=4'h1
sdio_ccmd	I/O	SDMMC1_CMD/GPIO1_B3_u	GRF_GPIO1B_IOMUX_L[15:12]=4'h1

Module Pin	Direction	PIN Name	IOMUX Setting
sdio_cdata0	I/O	SDMMC1_D0/GPIO1_B4_u	GRF_GPIO1B_IOMUX_H[3:0] =4'h1
sdio_cdata1	I/O	SDMMC1_D1/GPIO1_B5_u	GRF_GPIO1B_IOMUX_H[7:4] =4'h1
sdio_cdata2	I/O	SDMMC1_D2/GPIO1_B6_u	GRF_GPIO1B_IOMUX_H[11:8] =4'h1
sdio_cdata3	I/O	SDMMC1_D3/GPIO1_B7_u	GRF_GPIO1B_IOMUX_H[15:12]=4'h1
sdio_cdetn	I	I2S2_MCLK_M0/SDMMC1_DET/SPI1_CS1n_M1/I2C5_SCL_M2/UART1_TX_M1/GPIO1_D0_d	GRF_GPIO1D_IOMUX_L[3:0] =4'h2
sdio_pwr	O	SDMMC1_PWR/ I2C5_SDA_M2/UART1_RX_M1/ GPIO1_D1_d	GRF_GPIO1D_IOMUX_L[7:4] =4'h1

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

27.5.3 EMMC Interface Description

Table 27-10 EMMC Interface Description

Module Pin	Direction	PIN Name	IOMUX Setting
emmc_cclk	O	FLASH_CLE/EMMC_CLKO/GPIO0_D7_d	GRF_GPIO0D_IOMUX_H[15:12]=4'h2
emmc_ccmd	I/O	FLASH_WRn/EMMC_CMD/GPIO0_D5_u	GRF_GPIO0D_IOMUX_H[7:4] =4'h2
emmc_cdata0	I/O	FLASH_D0/EMMC_D0/GPIO0_C4_u	GRF_GPIO0C_IOMUX_H[3:0] =4'h2
emmc_cdata1	I/O	FLASH_D1/EMMC_D1/GPIO0_C5_u	GRF_GPIO0C_IOMUX_H[7:4] =4'h2
emmc_cdata2	I/O	FLASH_D2/EMMC_D2/GPIO0_C6_u	GRF_GPIO0C_IOMUX_H[11:8] =4'h2
emmc_cdata3	I/O	FLASH_D3/EMMC_D3/GPIO0_C7_u	GRF_GPIO0C_IOMUX_H[15:12]=4'h2
emmc_cdata4	I/O	FLASH_D4/EMMC_D4/GPIO0_D0_u	GRF_GPIO0D_IOMUX_L[3:0] =4'h2
emmc_cdata5	I/O	FLASH_D5/EMMC_D5/FSPI_CS1n/GPIO0_D1_u	GRF_GPIO0D_IOMUX_L[7:4] =4'h2
emmc_cdata6	I/O	FLASH_D6/EMMC_D6/GPIO0_D2_u	GRF_GPIO0D_IOMUX_L[11:8] =4'h2
emmc_cdata7	I/O	FLASH_D7/EMMC_D7/GPIO0_D3_u	GRF_GPIO0D_IOMUX_L[15:12] =4'h2
emmc_rstn	O	FLASH_WPn/EMMC_RSTn/FSPI_CLK/GPIO1_A3_d	GRF_GPIO1A_IOMUX_L[15:12] =4'h2

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

27.6 Application Notes

27.6.1 Card-Detect and Write-Protect Mechanism

Following figure illustrates how the SD/MMC card detection and write-protect signals are connected. Most of the SD/MMC sockets have card-detect pins. When no card is present, card_detect_n is 1 due to the pull-up. When the card is inserted, the card-detect pin is shorted to ground, which makes card_detect_n go to 0. Similarly in SD cards, when the write-protect switch is toward the left, it shorts the write_protect port to ground.

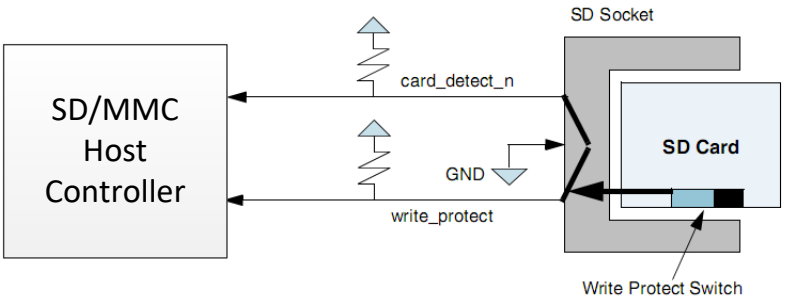


Fig. 27-10 SD/MMC Card-Detect and Write-Protect

27.6.2 SD/MMC Termination Requirement

Following Figure illustrates the SD/MMC termination requirements, which is required to pull up ccmd and cdata lines on the device bus. The recommended specification for pull-up on the ccmd line (Rcmd) is 4.7K - 100K for MMC, and 10K - 100K for an SD. The recommended pull-up on the cdata line (Rdat) is 50K - 100K.

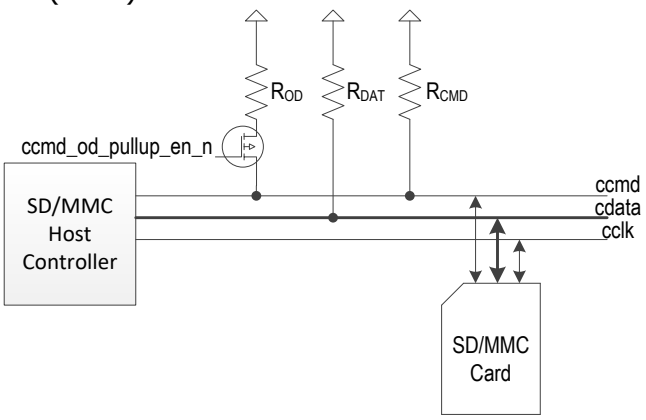


Fig. 27-11 SD/MMC Card Termination

Rcmd and Rod Calculation

The SD/MMC card enumeration happens at a very low frequency – 100-400KHz. Since the MMC bus is a shared bus between multiple cards, during enumeration open-drive mode is used to avoid bus conflict. Cards that drive 0 win over cards that drive "z". The pull-up in the command line pulls the bus to 1 when all cards drive "z". During normal data transfer, the host chooses only one card and the card driver switches to push-pull mode.

For example, if enumeration is done at 400KHz and the total bus capacitance is 200 pf, the pull-up needed during enumeration is:

2.2 RC = rise-time = 1/400KHz

R = 1/(2.2 * C * 100KHz)

= 1/(2.2 x 200 x 10**-12 x 400 x 10**3)

= 1/(17.6 x 10**-5)

= 5.68K

The ROD and RCMD should be adjusted in such a way that the effective pull-up is at the maximum 5.68K during enumeration. If there are only a few cards in the bus, a fixed RCMD resister is sufficient and there is no need for an additional ROD pull-up during enumeration. You should also ensure the effective pull-up will not violate the Iol rating of the drivers.

In SD mode, since each card has a separate bus, the capacitance is less, typically in the order of 20-30pf (host capacitance + card capacitance + trace + socket capacitance). For example, if enumeration is done at 400KHz and the total bus capacitance is 20pf, the pull-up needed during enumeration is:

2.2 RC = rise-time = 1/400KHz

R = 1/(2.2 * C * 100KHz)

= 1/(2.2 x 20 x 10**-12 x 400 x 10**3)

= 1/(1.76 x 10**-5)

= 56.8K

Therefore, a fixed 56.8K permanent Rcmd is sufficient in SD mode to enumerate the cards.

The driver of the SD/MMC on the "command" port needs to be only a push-pull driver. During enumeration, the SD/MMC emulates an open-drain driver by driving only a 0 or a"z" by controlling the ccmd_out and ccmd_out_en signals.

27.6.3 Software/Hardware Restriction

Before issuing a new data transfer command, the software should ensure that the card is not busy due to any previous data transfer command. Before changing the card clock frequency, the software must ensure that there are no data or command transfers in progress.

If the card is enumerated in SDR50, or DDR50 mode, then the application must program the use_hold_reg bit[29] in the SDMMC_CMD register to 1'b0 (phase shift of cclk_in_drv = 0) or 1'b1 (phase shift of cclk_in_drv>0). If the card is enumerated in SDR12 or SDR25 mode, the application must program the use_hold_reg bit[29] in the SDMMC_CMD register to 1'b1. This programming should be done for all data transfer commands and non-data commands that are sent to the card. When the use_hold_reg bit is programmed to 1'b0, the Host Controller bypasses the Hold Registers in the transmit path. The value of this bit should not be changed when a Command or Data Transfer is in progress. For more details on using use_hold_reg and the implementation requirements for meeting the Card input hold time, refer to "Recommended Usage" in following table.

Table 27-11 Recommended Usage of use_hold_reg

No.	Speed Mode	use_hold_reg	cclk_in (MHz)	clk_in_drv (MHz)	clk_divider	Phase shift
1	SDR104	1'b0	200	200	0	0
2	SDR104	1'b1	200	200	0	Tunable> 0
3	SDR50	1'b0	100	100	0	0
4	SDR50	1'b1	100	100	0	Tunable> 0
5	DDR50 (8bit)	1'b0	100	100	1	0
6	DDR50 (8bit)	1'b1	100	100	1	Tunable> 0
7	DDR50 (4bit)	1'b0	50	50	0	0
8	DDR50 (4bit)	1'b1	50	50	0	Tunable> 0
9	SDR25	1'b1	50	50	0	Tunable> 0
10	SDR12	1'b1	50	50	1	Tunable> 0

To avoid glitches in the card clock outputs, the software should use the following steps when changing the card clock frequency:

1) Before disable the clocks, ensure that the card is not busy due to any previous data command. To determine this, check for 0 in bit9 of SDMMC_STATUS register.
2) Update the Clock Enable register to disable all clocks. To ensure completion of any previous command before this update, send a command to the CIU to update the clock registers by setting:

- start_cmd bit
- "update clock registers only" bits
- "wait_previous data complete" bit

   Wait for the CIU to take the command by polling for 0 on the start_cmd bit.

3) Set the start_cmd bit to update the Clock Divider and/or Clock Source registers, and send a command to the CIU in order to update the clock registers; wait for the CIU to take the command.
4) Set start_cmd to update the Clock Enable register in order to enable the required clocks and send a command to the CIU to update the clock registers; wait for the CIU to take the command.

In non-DMA mode, while reading from a card, the Data Transfer Over (SDMMC_RINTSTS[3]) interrupt occurs as soon as the data transfer from the card is over. There still could be some data left in the FIFO, and the RX_WMark interrupt may or may not occur, depending on the remaining bytes in the FIFO. Software should read any remaining bytes upon seeing the

Data Transfer Over (DTO) interrupt. While using the external DMA interface for reading from a card, the DTO interrupt occurs only after all the data is flushed to memory by the DMA interface unit.
While writing to a card in external DMA mode, if an undefined-length transfer is selected by setting the Byte Count Register to 0, the DMA logic will likely request more data than it will send to the card, since it has no way of knowing at which point the software will stop the transfer. The DMA request stops as soon as the DTO is set by the CIU.
If the software issues a controller_reset command by setting control register bit[0] to 1, all the CIU state machines are reset; the FIFO is not cleared. The DMA sends all remaining bytes to the host. In addition to a card-reset, if a FIFO reset is also issued, then:

- Any pending DMA transfer on the bus completes correctly
- DMA data read is ignored
- Write data is unknown(x)

Additionally, if dma_reset is also issued, any pending DMA transfer is abruptly terminated. When the DMA is used, the DMA controller channel should also be reset and reprogrammed. If any of the previous data commands do not properly terminate, then the software should issue the FIFO reset in order to remove any residual data, if any, in the FIFO. After asserting the FIFO reset, you should wait until this bit is cleared.
One data-transfer requirement between the FIFO and host is that the number of transfers should be a multiple of the FIFO data width (32bits). For example, you want to write only 15 bytes to an SD/MMC card (SDMMC_BYTCNT), the host should write 16 bytes to the FIFO or program the DMA to do 16-byte transfers. The software can still program the Byte Count register to only 15, at which point only 15 bytes will be transferred to the card. Similarly, when 15 bytes are read from a card, the host should still read all 16 bytes from the FIFO. It is recommended that you not change the FIFO threshold register in the middle of data transfers.

27.6.4 Programming Sequence

27.6.4.1 Initialization

Following figure illustrates the initialization flow.

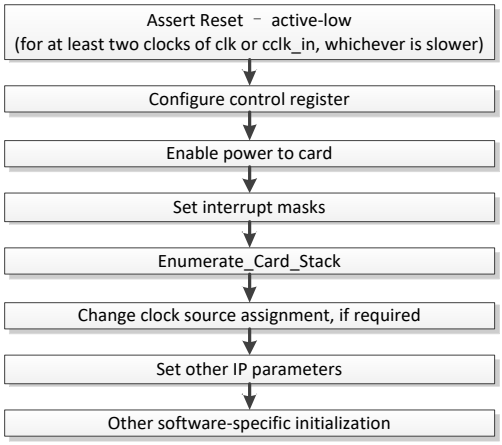


Fig. 27-12 Host Controller Initialization Sequence

Once the power and clocks are stable, reset_n should be asserted(active-low) for at least two cycles of clk or cclk_in, whichever is slower. The reset initializes the registers, ports, FIFO-pointers, DMA interface controls, and state-machines in the design. After power-on reset, the software should do the following:

1) Configure control register – For MMC mode, enable the open-drain pullup by setting enable_OD_pullup(bit24) in the SDMMC_CTRL register.
2) Enable power to cards – Before enabling the power, confirm that the voltage setting to the voltage regulators is correct. Enable power to the connected cards by setting the corresponding bit to 1 in the Power Enable register. Wait for the power ramp-up time.
3) Set masks for interrupts by clearing appropriate bits in the Interrupt Mask register. Set the global int_enable bit of the SDMMC_CTRL register. It is recommended that you write 0xffff_ffff to the Raw Interrupt register in order to clear any pending interrupts before setting the int_enable bit.
4) Enumerate card stack – Each card is enumerated according to card type; for details,

refer to "Enumerated Card Stack". For enumeration, you should restrict the clock frequency to 400KHz.

5) Changing clock source assignment – set the card frequency using the clock-divider and clock-source registers; for details, refer to "Clock Programming". MMC cards operate at a maximum of 20MHz (at maximum of 52MHz in high-speed mode). SD mode operates at a maximum of 25MHz (at maximum of 50MHz in high-speed mode).
6) Set other parameters, which normally do not need to be changed with every command, with a typical value such as timeout values in cclk_out according to SD/MMC specifications.

- ResponseTimeOut = 0x64
- DataTimeOut = highest of one of the following:
- (10*((TAAC*Fop)+(100*NSAC))
- Host FIFO read/write latency from FIFO empty/full
- Set the debounce value to 25ms(default:0x0fffff) in host clock cycle units in the SDMMC_DEBNCE register.
- FIFO threshold value in bytes in the SDMMC_FIFOTH register.

27.6.4.2 Enumerated Card Stack

The card stack does the following:

- Enumerates all connected cards
- Sets the RCA for the connected cards
- Reads card-specific information
- Stores card-specific information locally

Enumeration depends on the operating mode of the SD/MMC card; the card type is first identified and the appropriate card enumeration routine is called.

1) Check if the card is connected.
2) Clear the card type register to set the card width as a single bit. For the given card number, clear the corresponding bits in the card_type register. Clear the register bit for a 1-bit, 4-bit bus width. For example, for card number=1, clear bit 0 and bit 16 of the card_type register.
3) Set clock frequency to FOD=400KHz, maximum – Program clock divider0 (bits 0-7 in the SDMMC_CLKDIV register) value to one-half of the cclk_in frequency divided by 400KHz. For example, if cclk_in is 20MHz, then the value is 20, 000/(2*400)=25.
4) Identify the card type; that is, SD, MMC, or SDIO.
   a. Send CMD5 first. If a response is received, then the card is SDIO
   b. If not, send CMD8 with the following Argument
      Bit[31:12] = 20'h0 //reserved bits
      Bit[11:8] = 4'b0001 //VHS value
      Bit[7:0] = 8'b10101010 //Preferred Check Pattern by SD2.0
   c. If Response is received the card supports High Capacity SD2.0 then send ACMD41 with the following Argument
      Bit[31] = 1'b0; //Reserved bits
      Bit[30] = 1'b1; //High Capacity Status
      Bit[29:24] = 6'h0; //Reserved bits
      Bit[23:0] = Supported Voltage Range
   d. If Response is received for ACMD41 then the card is SD. Otherwise the card is MMC.
   e. If response is not received for initial CMD8 then card does not support High Capacity SD2.0, then issue CMD0 followed by ACMD41 with the following Argument
      Bit[31] = 1'b0; //Reserved bits
      Bit[30] = 1'b0; //High Capacity Status
      Bit[29:24] = 6'h0; //Reserved bits
      Bit[23:0] = Supported Voltage Range
5) Enumerate the card according to the card type.
6) Use a clock source with a frequency = Fod (that is, 400KHz) and use the following enumeration command sequence:

- SD card – Send CMD0, CMD8, ACMD41, CMD2, CMD3.
- MMC – Send CMD0, CMD1, CMD2, CMD3.

27.6.4.3 Clock Programming

The Host Controller supports one clock source. The clock to an individual card can be enabled or disabled. Registers that support this are:

- SDMMC_CLKDIV – Programs individual clock source frequency. SDMMC_CLKDIV limited to 0 or 1 is recommended.
- SDMMC_CLKSRC – Assign clock source for each card.
- SDMMC_CLKENA – Enables or disables clock for individual card and enables low-power mode, which automatically stops the clock to a card when the card is idle for more than 8 clocks.

The Host Controller loads each of these registers only when the start_cmd bit and the update_clk_regs_only bit in the SDMMC_CMD register are set. When a command is successfully loaded, the Host Controller clears this bit, unless the Host Controller already has another command in the queue, at which point it gives an HLE(Hardware Locked Error). Software should look for the start_cmd and the update_clk_regs_only bits, and should also set the wait_prvdata_complete bit to ensure that clock parameters do not change during data transfer. Note that even though start_cmd is set for updating clock registers, the Host Controller does not raise a command_done signal upon command completion.

The following shows how to program these registers:

1) Confirm that no card is engaged in any transaction; if there is a transaction, wait until it finishes.
2) Stop all clocks by writing 0 to the SDMMC_CLKENA register. Set the start_cmd, Update_clk_regs_only, and wait_prvdata_complete bits in the SDMMC_CMD register. Wait until start_cmd is cleared or an HLE is set; in case of an HLE, repeat the command.
3) Program the SDMMC_CLKDIV and SDMMC_CLKSRC registers, as required. Set the start_cmd, Update_clk_regs_only, and wait_prvdata_complete bits in the SDMMC_CMD register. Wait until start_cmd is cleared or an HLE is set; in case of an HLE, repeat the command.
4) Re-enable all clocks by programming the SDMMC_CLKENA register. Set the start_cmd, update_clk_regs_only, and wait_prvdata_complete bits in the SDMMC_CMD register. Wait until start_cmd is cleared or an HLE is set; in case of an HLE, repeat the command.

27.6.4.4 No-Data Command With or Without Response Sequence

To send any non-data command, the software needs to program the SDMMC_CMD register @0x2C and the SDMMC_CMDARG register @0x28 with appropriate parameters. Using these two registers, the Host Controller forms the command and sends it to the command bus. The Host Controller reflects the errors in the command response through the error bits of the SDMMC_RINTSTS register.

When a response is received – either erroneous or valid – the Host Controller sets the command_done bit in the SDMMC_RINTSTS register. A short response is copied in Response Register0, while along response is copied to all four response registers @0x30, 0x34, 0x38, and 0x3C. The Response3 register bit 31 represents the MSB, and the Response0 register bit 0 represents the LSB of a long response.

For basic commands or non-data commands, follow these steps:

1) Program the Command register @0x28 with the appropriate command argument parameter.
2) Program the Command register @0x2C with the settings in following table.

Table 27-12 Command Settings for No-Data Command

Parameter	Value	Description
Default		
start_cmd	1	-
use_hold_reg	1/0	Choose value based on speed mode being used;ref to "use_hold_reg" on SDMMC_CMD register
update_clk_regs_only	0	No clock parameters update command
data_expected	0	No data command
card number	0	Actual card number(one

Parameter	Value	Description
		controller only connect one card, the num is No. 0)
cmd_index	command-index	-
send_initialization	0	Can be 1, but only for card reset commands, such as CMD0
stop_abort_cmd	0	Can be 1 for commands to stop data transfer, such as CMD12
response_length	0	Can be 1 for R2(long) response
response_expect	1	Can be 0 for commands with no response; for example, CMD0, CMD4, CMD15, and so on
User-selectable		
wait_prvdata_complete	1	Before sending command on command line, host should wait for completion of any data command in process, if any (recommended to always set this bit, unless the current command is to query status or stop data transfer when transfer is in progress)
check_response_crc	1	If host should crosscheck CRC of response received

3) Wait for command acceptance by host. The following happens when the command is loaded into the Host Controller:
- Host Controller accepts the command for execution and clears the start_cmd bit in the SDMMC_CMD register, unless one command is in process, at which point the Host Controller can load and keep the second command in the buffer.
- If the Host Controller is unable to load the command – that is, a command is already in progress, a second command is in the buffer, and a third command is attempted – then it generates an HLE (hardware-locked error).

4) Check if there is an HLE.
5) Wait for command execution to complete. After receiving either a response from a card or response timeout, the Host Controller sets the command_done bit in the SDMMC_RINTSTS register. Software can either poll for this bit or respond to a generated interrupt.
6) Check if response_timeout error, response_CRC error, or response error is set. This can be done either by responding to an interrupt raised by these errors or by polling bits 1, 6, and 8 from the SDMMC_RINTSTS register @0x44. If no response error is received, then the response is valid. If required, the software can copy the response from the response registers @0x30-0x3C.

Software should not modify clock parameters while a command is being executed.

27.6.4.5 Data Transfer Commands

Data transfer commands transfer data between the memory card and the Host Controller. To send a data command, the Host Controller needs a command argument, total data size, and block size. Software can receive or send data through the FIFO.

Before a data transfer command, software should confirm that the card is not busy and is in a transfer state, which can be done using the CMD13 and CMD7 commands, respectively.

For the data transfer commands, it is important that the same bus width that is programmed in the card should be set in the card type register @0x18.

The Host Controller generates an interrupt for different conditions during data transfer, which are reflected in the SDMMC_RINTSTS register @0x44 as:

1) Data_Transfer_Over (bit 3) – When data transfer is over or terminated. If there is a response timeout error, then the Host Controller does not attempt any data transfer and the "Data Transfer Over" bit is never set.
2) Transmit_FIFO_Data_request (bit 4) – FIFO threshold for transmitting data was reached; software is expected to write data, if available, in FIFO.
3) Receive_FIFO_Data_request (bit 5) – FIFO threshold for receiving data was reached; software is expected to read data from FIFO.
4) Data starvation by Host timeout (bit 10) – FIFO is empty during transmission or is full during reception. Unless software writes data for empty condition or reads data for full condition, the Host Controller cannot continue with data transfer. The clock to the card has been stopped.
5) Data read timeout error (bit 9) – Card has not sent data within the timeout period.
6) Data CRC error (bit 7) – CRC error occurred during data reception.
7) Start bit error (bit 13) – Start bit was not received during data reception.
8) End bit error (bit 15) – End bit was not received during data reception or for a write operation; a CRC error is indicated by the card.

Conditions 6), 7), and 8) indicate that the received data may have errors. If there was a response timeout, then no data transfer occurred.

27.6.4.6 Single-Block or Multiple-Block Read

Steps involved in a single-block or multiple-block read are:

1) Write the data size in bytes in the SDMMC_BYTCNT register @0x20.
2) Write the block size in bytes in the SDMMC_BLKSIZ register @0x1C. The Host Controller expects data from the card in blocks of size SDMMC_BLKSIZ each.
3) Program the SDMMC_CMDARG register @0x28 with the data address of the beginning of a data read.
4) Program the Command register with the parameters listed in following table. For SD and MMC cards, use CMD17 for a single-block read and CMD18 for a multiple-block read. For SDIO cards, use CMD53 for both single-block and multiple-block transfers.

Table 27-13 Command Setting for Single or Multiple-Block Read

Parameter	Value	Description
Default		
start_cmd	1	-
use_hold_reg	1/0	Choose value based on speed mode being used;ref to "use_hold_reg" on SDMMC_CMD register
update_clk_regs_only	0	No clock parameters update command
card number	0	Actual card number(one controller only connect one card, the num is No.0)
send_initialization	0	Can be 1, but only for card reset commands, such as CMD0
stop_abort_cmd	0	Can be 1 for commands to stop data transfer, such as CMD12
send_auto_stop	0/1	-
transfer_mode	0	Block transfer
read_write	0	Read from card
data_expected	1	Data command
response_length	0	Can be 1 for R2(long) response
response_expect	1	Can be 0 for commands with no response; for example, CMD0, CMD4, CMD15, and so

Parameter	Value	Description
		on
User-selectable		
cmd_index	command-index	-
wait_prvdata_complete	1	0: Sends command immediately 1: Sends command after previous data transfer ends
check_response_crc	1	0: Host Controller should not check response CRC 1: Host Controller should check response CRC

After writing to the SDMMC_CMD register, the Host Controller starts executing the command; when the command is sent to the bus, the command_done interrupt is generated.

- Software should look for data error interrupts; that is, bits 7, 9, 13, and 15 of the SDMMC_RINTSTS register. If required, software can terminate the data transfer by sending a STOP command.
- Software should look for Receive_FIFO_Data_request and/or data starvation by host timeout conditions. In both cases, the software should read data from the FIFO and make space in the FIFO for receiving more data.
- When a Data_Transfer_Over interrupt is received, the software should read the remaining data from the FIFO.

27.6.4.7 Single-Block or Multiple-Block Write

Steps involved in a single-block or multiple-block write are:

1) Write the data size in bytes in the SDMMC_BYTCNT register @0x20.
2) Write the block size in bytes in the SDMMC_BLKSIZ register @0x1C; the Host Controller sends data in blocks of size SDMMC_BLKSIZ each.
3) Program SDMMC_CMDARG register @0x28 with the data address to which data should be written.
4) Write data in the FIFO; it is usually best to start filling data the full depth of the FIFO.
5) Program the Command register with the parameters listed in following table.

Table 27-14 Command Settings for Single or Multiple-Block Write

Parameter	Value	Description
Default		
start_cmd	1	-
use_hold_reg	1/0	Choose value based on speed mode being used;ref to "use_hold_reg" on SDMMC_CMD register
update_clk_regs_only	0	No clock parameters update command
card number	0	Actual card number(one controller only connect one card, the num is No. 0)
send_initialization	0	Can be 1, but only for card reset commands, such as CMD0
stop_abort_cmd	0	Can be 1 for commands to stop data transfer, such as CMD12
send_auto_stop	0/1	-
transfer_mode	0	Block transfer
read_write	1	Write to card
data_expected	1	Data command
response_length	0	Can be 1 for R2(long)

Parameter	Value	Description
		response
response_expect	1	Can be 0 for commands with no response; for example, CMD0, CMD4, CMD15, and so on
User-selectable		
cmd_index	command-index	-
wait_prvdata_complete	1	0: Sends command immediately 1: Sends command after previous data transfer ends
check_response_crc	1	0: Host Controller should not check response CRC 1: Host Controller should check response CRC

After writing to the SDMMC_CMD register, Host Controller starts executing a command; when the command is sent to the bus, a command_done interrupt is generated.

- Software should look for data error interrupts; that is, for bits 7, 9, and 15 of the SDMMC_RINTSTS register. If required, software can terminate the data transfer by sending the STOP command.
- Software should look for Transmit_FIFO_Data_Request and/or timeout conditions from data starvation by the host. In both cases, the software should write data into the FIFO.
- When a Data_Transfer_Over interrupt is received, the data command is over. For an open-ended block transfer, if the byte count is 0, the software must send the STOP command. If the byte count is not 0, then upon completion of a transfer of a given number of bytes, the Host Controller should send the STOP command, if necessary. Completion of the AUTO-STOP command is reflected by the Auto_command_done interrupt – bit 14 of the SDMMC_RINTSTS register. A response to AUTO_STOP is stored in SDMMC_RESP1 @0x34.

27.6.4.8 Stream Read

A stream read is like the block read mentioned in "Single-Block or Multiple-Block Read", except for the following bits in the Command register:

transfer_mode = 1; //Stream transfer
cmd_index = CMD20;

A stream transfer is allowed for only a single-bit bus width.

27.6.4.9 Stream Write

A stream write is exactly like the block write mentioned in "Single-Block or Multiple-Block Write", except for the following bits in the Command register:

transfer_mode = 1;//Stream transfer
cmd_index = CMD11;

In a stream transfer, if the byte count is 0, then the software must send the STOP command. If the byte count is not 0, then when a given number of bytes completes a transfer, the Host Controller sends the STOP command. Completion of this AUTO_STOP command is reflected by the Auto_command_done interrupt. A response to an AUTO_STOP is stored in the SDMMC_RESP1 register@0x34.

A stream transfer is allowed for only a single-bit bus width.

27.6.4.10 Sending Stop or Abort in Middle of Transfer

The STOP command can terminate a data transfer between a memory card and the Controller, while the ABORT command can terminate an I/O data transfer for only the SDIO_IOONLY and SDIO_COMBO cards.

- Send STOP command – Can be sent on the command line while a data transfer is in progress; this command can be sent at any time during a data transfer.

You can also use an additional setting for this command in order to set the Command register bits (5-0) to CMD12 and set bit 14 (stop_abort_cmd) to 1. If stop_abort_cmd is not set to 1, the Controller does not know that the user stopped a data transfer. Reset bit 13 of the Command register (wait_prvdata_complete) to 0 in order to make the Controller send

the command at once, even though there is a data transfer in progress.

- Send ABORT command – Can be used with only an SDIO_IOONLY or SDIO_COMBO card. To abort the function that is transferring data, program the function number in ASx bits (CCCR register of card, address 0x06, bits (0-2) using CMD52.

27.6.4.11 Read_Wait Sequence

Read_wait is used with only the SDIO card and can temporarily stall the data transfer—either from function or memory—and allow the host to send commands to any function within the SDIO device. The host can stall this transfer for as long as required. The Host Controller provides the facility to signal this stall transfer to the card. The steps for doing this are:

1) Check if the card supports the read_wait facility; read SRW (bit 2) of the CCCR register @0x08. If this bit is 1, then all functions in the card support the read_wait facility. Use CMD52 to read this bit.
2) If the card supports the read_wait signal, then assert it by setting the read_wait (bit 6) in the SDMMC_CTRL register @0x00.
3) Clear the read_wait bit in the SDMMC_CTRL register.

27.6.4.12 Controller/DMA/FIFO Reset Usage

- Controller reset – Resets the controller by setting the controller_reset bit (bit 0) in the SDMMC_CTRL register; this resets the CIU and state machines, and also resets the BIU-to-CIU interface. Since this reset bit is self-clearing, after issuing the reset, wait until this bit is cleared.
- FIFO reset - Resets the FIFO by setting the fifo_reset bit (bit 1) in the SDMMC_CTRL register; this resets the FIFO pointers and counters of the FIFO. Since this reset bit is self-clearing, after issuing the reset, wait until this bit is cleared.

In external DMA transfer mode, even when the FIFO pointers are reset, if there is a DMA transfer in progress, it could push or pop data to or from the FIFO; the DMA itself completes correctly. In order to clear the FIFO, the software should issue an additional FIFO reset and clear any FIFO under-run or overrun errors in the SDMMC_RAWINTS register caused by the DMA transfers after the FIFO was reset.

27.6.4.13 Card Read Threshold

When an application needs to perform a Single or Multiple Block Read command, the application must program the SDMMC_CARDTHRCTL register with the appropriate Card Read Threshold size (CardRdThreshold) and set the Card Read Threshold Enable (CardRdThrEnable) bit to 1'b1. This additional programming ensures that the Host controller sends a Read Command only if there is space equal to the Card Read Threshold available in the Rx FIFO. This in turn ensures that the card clock is not stopped in the middle a block of data being transmitted from the card. The Card Read Threshold can be set to the block size of the transfer, which guarantees that there is a minimum of one block size of space in the RxFIFO before the controller enables the card clock. The Card Read Threshold is required when the Round Trip Delay is greater than 0.5cclk_in period.

27.6.4.14 Error Handling

The Host Controller implements error checking; errors are reflected in the SDMMC_RAWINTS register@0x44 and can be communicated to the software through an interrupt, or the software can poll for these bits. Upon power-on, interrupts are disabled (int_enable in the SDMMC_CTRL register is 0), and all the interrupts are masked (bits 0-31 of the SDMMC_INTMASK register; default is 0).

Error handling:

- Response and data timeout errors – For response timeout, software can retry the command. For data timeout, the Host Controller has not received the data start bit – either for the first block or the intermediate block – within the timeout period, so software can either retry the whole data transfer again or retry from a specified block onwards. By reading the contents of the SDMMC_TCBCNT later, the software can decide how many bytes remain to be copied.
- Response errors – Set when an error is received during response reception. In this case, the response that copied in the response registers is invalid. Software can retry the command.
- Data errors – Set when error in data reception are observed; for example, data CRC,

start bit not found, end bit not found, and so on. These errors could be set for any block-first block, intermediate block, or last block. On receipt of an error, the software can issue a STOP or ABORT command and retry the command for either whole data or partial data.

- Hardware locked error – Set when the Host Controller cannot load a command issued by software. When software sets the start_cmd bit in the SDMMC_CMD register, the Host Controller tries to load the command. If the command buffer is already filled with a command, this error is raised. The software then has to reload the command.
- FIFO under-run/overrun error – If the FIFO is full and software tries to write data in the FIFO, then an overrun error is set. Conversely, if the FIFO is empty and the software tries to read data from the FIFO, an under-run error is set. Before reading or writing data in the FIFO, the software should read the fifo_empty or fifo_full bits in the Status register.
- Data starvation by host timeout – Raised when the Host Controller is waiting for software intervention to transfer the data to or from the FIFO, but the software does not transfer within the stipulated timeout period. Under this condition and when a read transfer is in process, the software should read data from the FIFO and create space for further data reception. When a transmit operation is in process, the software should fill data in the FIFO in order to start transferring data to the card.
- CRC Error on Command – If a CRC error is detected for a command, the CE-ATA device does not send a response, and a response timeout is expected from the Host Controller. The ATA layer is notified that an MMC transport layer error occurred.

Notes: During a multiple-block data transfer, if a negative CRC status is received from the device, the data path signals a data CRC error to the BIU by setting the data CRC error bit in the SDMMC_RINTSTS register. It then continues further data transmission until all the bytes are transmitted.

27.6.5 Voltage Switching

The Host Controller supports SD 3.0 Ultra High Speed (UHS-1) and is capable of voltage switching in SD-mode, which can be applied to SD High-Capacity (SDHC) and SD Extended Capacity (SDXC) cards. UHS-1 supports only 4-bit mode.

However, whether the IO voltage of 1.8v supported or not is depended on the SoC design.

SD 3.0 UHS-1 supports the following transfer speed modes for UHS-50 and/or UHS-104 cards:

- DS – default-speed up to 25MHz, 3.3V signaling
- HS – high-speed up to 50MHz, 3.3V signaling
- SDR12 – SDR up to SDR 25MHz, 1.8V signaling
- SDR25 – SDR up to 50MHz, 1.8V signaling
- SDR50 – SDR up to 100MHz, 1.8V signaling
- DDR50 – DDR up to 50MHz, 1.8V signaling

Voltage selection can be done in only SD mode. The first CMD0 selects the bus mode-either SD mode or SPI mode. The card must be in SD mode in order for 1.8V signaling mode to apply, during which time the card cannot be switched to SPI mode or 3.3V signaling without a power cycle.

If the System BIOS in an embedded system already knows that it is connected to an SD 3.0 card, then the driver programs the Controller to initiate ACMD41. The software knows from the response of ACMD41 whether or not the card supports voltage switching to 1.8V.

- If bit 32 of ACMD41 response is 1'b1: card supports voltage switching and next command-CMD11-invokes voltage switching sequence. After CMD11 is started, the software must program the IO voltage selection register based on the soc architecture.
- If bit 32 of ACMD41 response is 1'b0: card does not support voltage switching and CMD11 should not be started.

If the card and host controller accept voltage switching, then they support UHS-1 modes of data transfer. After the voltage switch to 1.8V, SDR12 is the default speed.

Since the UHS-1 can be used in only 4-bit mode, the software must start ACMD6 and change the card data width to 4-bit mode; ACMD6 is driven in any of the UHS-1 speeds. If the host wants to select the DDR mode of data transfer, then the software must program the SDMMC_DDR_REG register in the CSR space with the appropriate card number.

To choose from any of the SDR or DDR modes, appropriate values should be programmed in

the SDMMC_CLKDIV register.

27.6.5.1 Voltage Switch Operation

The Voltage Switch operation must be performed in SD mode only.

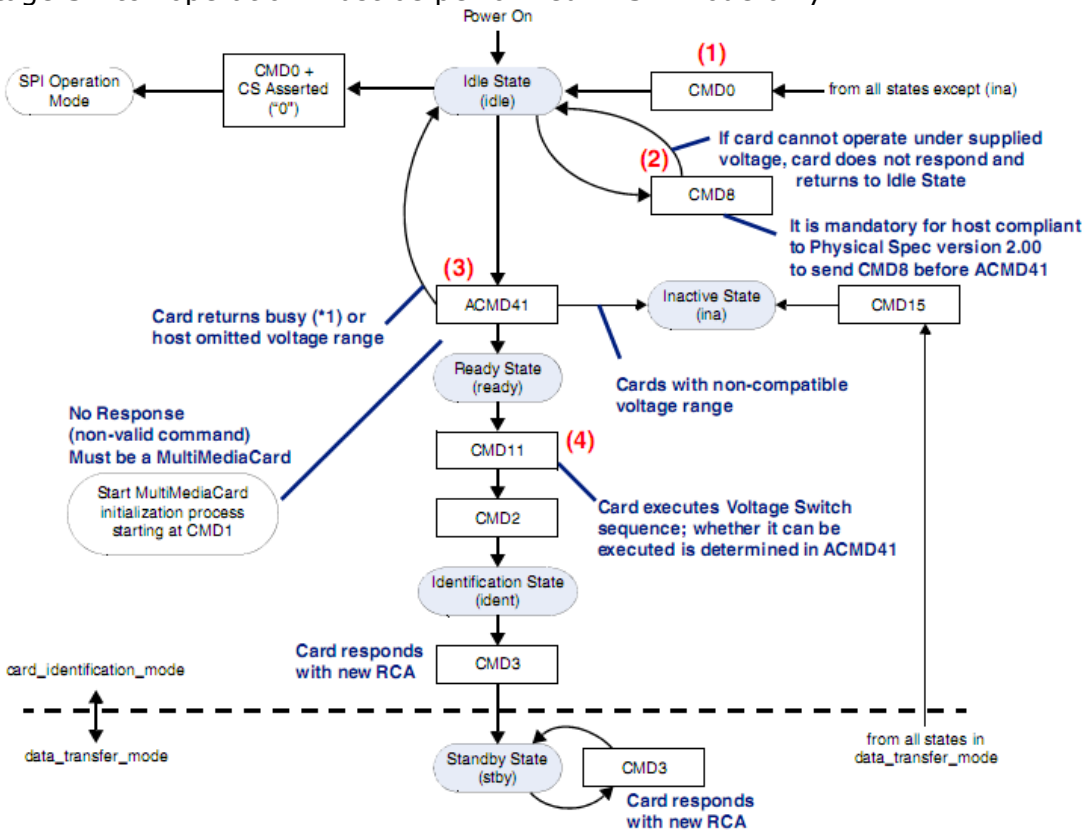


Fig. 27-13 Voltage Switching Command Flow Diagram

The following outlines the steps for the voltage switch programming sequence

1) Software Driver starts CMD0, which selects the bus mode as SD.
2) After the bus is in SD card mode, CMD8 is started in order to verify if the card is compatible with the SD Memory Card Specification, Version 2. 00. CMD8 determines if the card is capable of working within the host supply voltage specified in the VHS (19:16) field of the CMD; the card supports the current host voltage if a response to CMD8 is received.
3) ACMD 41 is started. The response to this command informs the software if the card supports voltage switching; bits 38, 36, and 32 are checked by the card argument of ACMD41; refer to following figure.

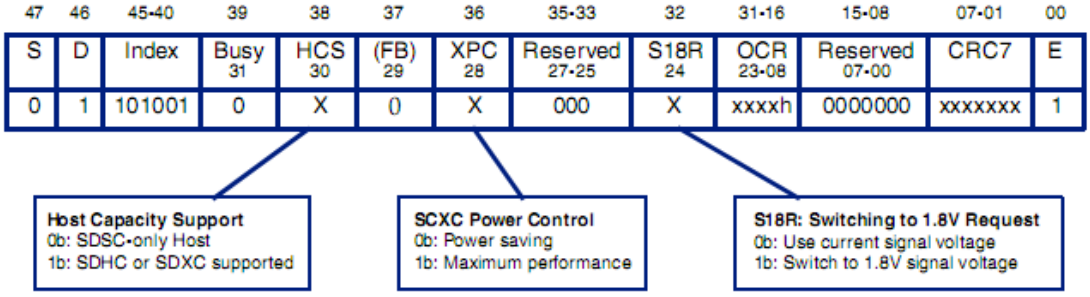


Fig. 27-14 ACMD41 Argument

- Bit 30 informs the card if host supports SDHC/SDXC or not; this bit should be set to 1'b1.
- Bit 28 can be either 1 or 0.
- Bit 24 should be set to 1'b1, indicating that the host is capable of voltage switching; refer to following figure.

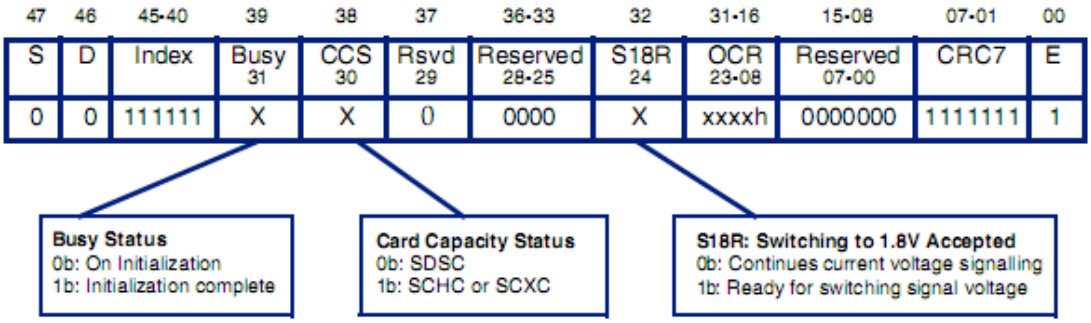


Fig. 27-15 ACMD41 Response(R3)

- Bit 30 – If set to 1'b1, card supports SDHC/SDXC; if set to 1'b0, card supports only SDSC
- Bit 24 – If set to 1'b1, card supports voltage switching and is ready for the switch
- Bit 31 – If set to 1'b1, initialization is over; if set to 1'b0, means initialization in process

4) If the card supports voltage switching, then the software must perform the steps discussed for either the "Voltage Switch Normal Scenario" or the "Voltage Switch Error Scenario".

27.6.5.2 Voltage Switch Normal Scenario

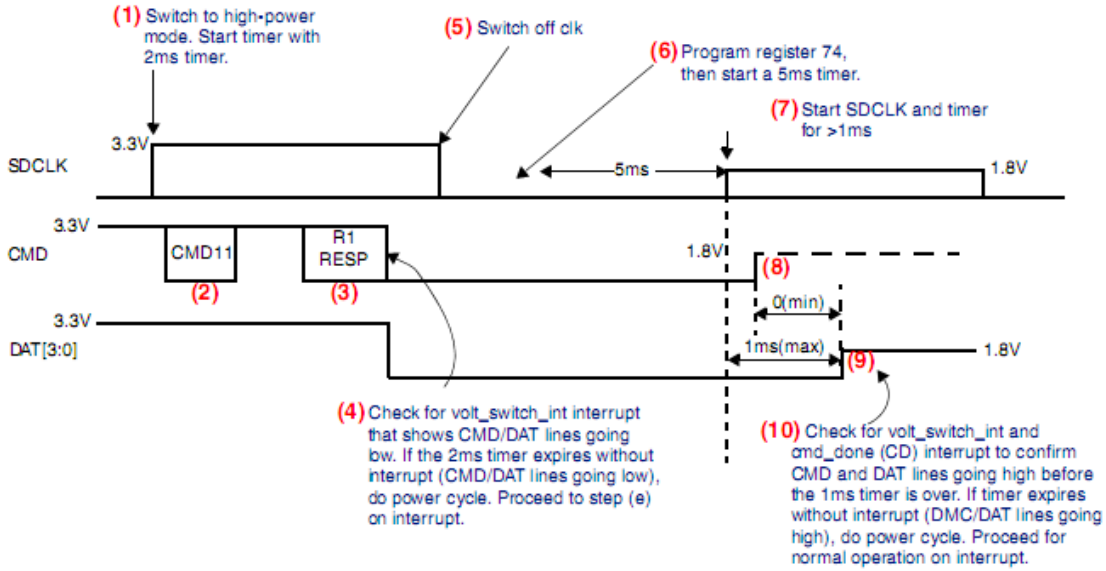


Fig. 27-16 Voltage Switch Normal Scenario

1) The host programs SDMMC_CLKENA—cclk_low_power register—with zero (0) for the corresponding card, which makes the host controller move to high-power mode. The application should start a timer with a recommended value of 2ms; this value of 2 ms is determined as below: Total cycles required for CMD11 = 48 cycles
Total cycles required for RESP R1 = 48 cycles
Maximum clock delay between MCD11 end to start of RESP1 = 60 cycles
Total = 48+48 + 60 = 160
Minimum frequency during enumeration is 100 KHz; that is, 10us
Total time = 160 * 10us = 1600us = 1. 6ms ~ 2ms
2) The host issues CMD11 to start the voltage switch sequence. Set bit 28 to 1'b1 in CMD when setting CMD11; for more information on setting bits, refer to "Boot Operation".
3) The card returns R1 response; the host controller does not generate cmd_done interrupt on receiving R1 response.
4) The card drives CMD and DAT [3:0] to low immediately after the response. The host controller generates interrupt (VOLT_SWITCH_INT) once the CMD or DAT [3:0] line goes low. The application should wait for this interrupt. If the 2ms timer expires without an interrupt (CMD/DAT lines going low), do a power cycle.

Note: Before doing a power cycle, switch off the card clock by programming SDMMC_CLKENA register

Proceed to step (5) on getting an interrupt (VOLT_SWITCH_INT).

Note: This interrupt must be cleared once this interrupt is received. Additionally, this interrupt should not be

masked during the voltage switch sequence.
If the timer expires without interrupt (CMD/DAT lines going low), perform a power cycle. Proceed to step (5) on interrupt.

1) Program the SDMMC_CLKENA register, with 0 for the corresponding card; the host stops supplying SDCLK.
2) Program Voltage register to the required values for the corresponding card. The application should start a timer > 5ms.
3) After the 5ms timer expires, the host voltage regulator is stable. Program SDMMC_CLKENA register, with 1 for the corresponding card; the host starts providing SDCLK at 1. 8V; this can be at zero time after Voltage register has been programmed. When the SDMMC_CLKENA register is programmed, the application should start another timer > 1ms.
4) By detecting SDCLK, the card drives CMD to high at 1. 8V for at least one clock and then stops driving (tri-state); CMD is triggered by the rising edge of SDCLK (SDR timing).
5) If switching to 1. 8V signaling is completed successfully, the card drives DAT [3:0] to high at 1. 8V for at least one clock and then stops driving (tri-state); DAT [3:0] is triggered by the rising edge of SDCLK (SDR timing). DAT[3:0] must be high within 1ms from the start of SDCLK.
6) The host controller generates a voltage switch interrupt (VOLT_SWITCH_INT) and a command done (CD) interrupt once the CMD and DAT[3:0] lines go high. The application should wait for this interrupt to confirm CMD and DAT lines going high before the 1ms timer is done.

If the timer expires without the voltage switch interrupt (VOLT_SWITCH_INT), a power cycle should be performed. Program the SDMMC_CLKENA register to stop the clock for the corresponding card number. Wait for the cmd_done (CD) interrupt. Proceed for normal operation on interrupt. After the sequence is completed, the host and the card start communication in SDR12 timing.

27.6.5.3 Voltage Switch Error Scenario

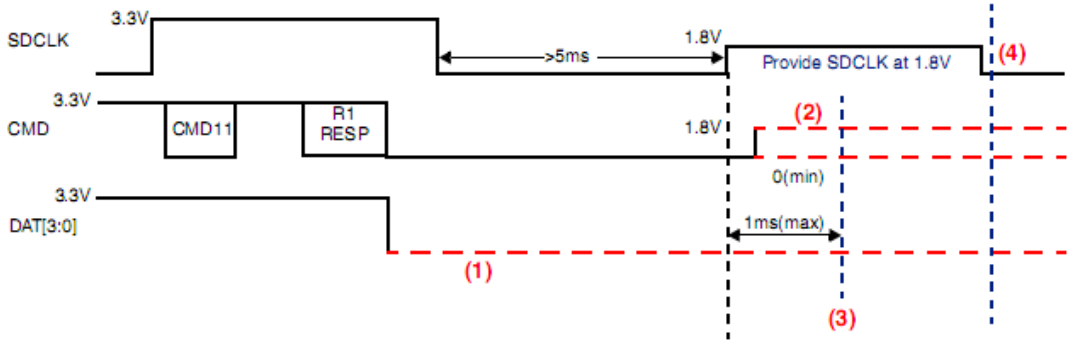


Fig. 27-17 Voltage Switch Error Scenario

1) If the interrupt (VOLT_SWITCH_INT) does not come, then the 2 ms timer should time out and a power cycle should be initiated.

Note: Before performing a power cycle, switch off the card clock by programming SDMMC_CLKENA register; no cmd_done (CD) interrupt is generated.

Additionally, if the card detects a voltage error at any point in between steps (5) and (7) in the card keeps driving DAT[3:0] to low until card power off.

2) CMD can be low or tri-state.
3) The host controller generates a voltage switch interrupt once the CMD and DAT[3:0] lines go high. The application should check for an interrupt to confirm CMD and DAT lines going high before the 1 ms timer is done.

If the 1 ms timer expires without interrupt (VOLT_SWITCH_INT) and cmd_done (CD), a power cycle should be performed. Program the SDMMC_CLKENA register to stop SDCLK of the corresponding card. Wait for the cmd_done interrupt. Proceed for normal operation on interrupt.

4) If DAT[3:0] is low, the host drives SDCLK to low and then stops supplying the card power.

Note: The card checks voltages of its own regulator output and host signals to ensure they are less than 2. 5V. Errors are indicated by (1) and (2).

- If voltage switching is accepted by the card, the default speed is SDR12.
- Command Done is given:
  - If voltage switching is properly done, CMD and DAT line goes high.
  - If switching is not complete, the 1ms timer expires, and the card clock is switched off.

Note: No other CMD should be driven before the voltage switching operation is completed and Command Done is received.

- The application should use CMD6 to check and select the particular function; the function appropriate-speed should be selected.

After the function switches, the application should program the correct value in the CLKDIV register, depending on the function chosen. Additionally, if Function 0x4 of the Access mode is chosen—that is, DDR50, then the application should also program 1'b1 in DDR_REG for the card number that has been selected for DDR50 mode.

27.6.6 DDR Operation

27.6.6.1 4-bit DDR Programming Sequence

DDR programming should be done only after the voltage switch operation has completed. The following outlines the steps for the DDR programming sequence:

1) Once the voltage switch operation is complete, the user must program voltage selection register to the required values for the corresponding card.

- To start a card to work in DDR mode, the application must program a bit of the newly defined SDMMC_UHSREG[16] register with a value of 1'b1.
- The bit that the user programs depends on which card is to be accessed in DDR mode.

2) To move back to SDR mode, a power cycle should be run on the card—putting the card in SDR12 mode—and only then should SDMMC_UHSREG[16]be set back to 1'b0 for the appropriate card.

27.6.6.2 8-bit DDR Programming Sequence

The following outlines the steps for the 8-bit DDR programming sequence:

1) The cclk_in signal should be twice the speed of the required cclk_out. Thus, if the cclk_out signal is required to be 50 MHz, the cclk_in signal should be 100 MHz.
2) The SDMMC_CLKDIV register should always be programmed with a value higher than zero (0); that is, a clock divider should always be used for 8-bit DDR mode.
3) The application must program the SDMMC_UHSREG[16] register (ddr_reg bits) by assigning it with a value of 1 for the bit corresponding to the card number; this causes the selected card to start working in DDR mode.
4) Depending on the card number, the SDMMC_CTYPE [16] bits should be set in order to make the host work in the 8-bit mode.

27.6.6.3 eMMC4.5 DDR START Bit

The eMMC4.5 changes the START bit definition in the following manner:

1) Receiver samples the START bit on the rising edge.
2) On the next rising edge after sampling the START bit, the receiver must sample the data.
3) Removes requirement of the START bit and END bit to be high for one full cycle.

Notes: The Host Controller does not support a START bit duration higher than one clock cycle. START bit durations of one or less than one clock cycle are supported and can be defined at the time of startup by programming the EMMC_DDR_REG register.

27.6.6.4 Reset Command/Moving From DDR50 to SDR12

To reset the mode of operation from DDR50 to SDR12, the following sequence of operations has to be done by the application:

1) Issue CMD0.
2) When CMD0 is received, the card changes from DDR50 to SDR12.
3) Program the SDMMC_CLKDIV register with an appropriate value.
4) Set ddr_reg to 0.

Note: The Voltage register should not be programmed to 0 while switching from DDR50 to SDR12, since the card is still operating in 1.8V mode after receiving CMD0.

27.6.7 H/W Reset Operation

When the RST_n signal goes low, the card enters a pre-idle state from any state other than the inactive state.

The following outlines the steps for the H/W reset programming sequence:

1) Program CMD12 to end any transfer in process.
2) Wait for DTO, even if no response is sent back by the card.
3) Set the following resets:
- DMA reset–SDMMC_CTRL [2] bit
- FIFO reset–SDMMC_CTRL [1] bit

Note: The above steps are required only if a transfer is in process.

4) Program the SDMMC_RSTN register with a value of 0; this can be done at any time when the card is connected to the controller. This programming asserts the RST_n signal and resets the card.
5) Wait for minimum of 1 µs or cclk_in period, whichever is greater
6) After a minimum of 1 µs, the application should program a value of 0 into the SDMMC_RSTN register. This de-asserts the RST_n signal and takes the card out of reset.
7) The application can program a new CMD only after a minimum of 200 µs after the de-assertion of the RST_n signal, as per the MMC 4.41 standard.

Note: For backward compatibility, the RST_n signal is temporarily disabled in the card by default. The host may need to set the signal as either permanently enabled or permanently disabled before it uses the card.

27.6.8 FBE Scenarios

An FBE occurs due to an AHB error response on the AHB bus. This is a system error, so the software driver should not perform any further programming to the Host. The only recovery mechanism from such scenarios is to do one of the following:

- Issue a hard reset by asserting the reset_n signal
- Do a program controller reset by writing to the SDMMC_CTRL[0] bit

27.6.8.1 FIFO Overflow and Underflow

During normal data transfer conditions, FIFO overflow and underflow will not occur. However if there is a programming error, then FIFO overflow/underflow can result. For example, consider the following scenarios.

- For transmit: PBL=4, Tx watermark = 1. For the above programming values, if the FIFO has only one location empty, it issues a dma_req to IDMAC FSM. Due to PBL value=4, the IDMAC FSM performs 4 pushes into the FIFO. This will result in a FIFO overflow interrupt.
- For receive: PBL=4, Rx watermark = 1. For the above programming values, if the FIFO has only one location filled, it issues a dma_req to IDMAC FSM. Due to PBL value=4, the IDMAC FSM performs 4 pops to the FIFO. This will result in a FIFO underflow interrupt.

The driver should ensure that the number of bytes to be transferred as indicated in the descriptor should be a multiple of 4bytes with respect to H_DATA_WIDTH=32. For example, if the SDMMC_BYTCNT=13, the number of bytes indicated in the descriptor should be 16 for H_DATA_WIDTH=32.

27.6.8.2 Programming of PBL and Watermark Levels

The DMAC performs data transfers depending on the programmed PBL and threshold values.

Table 27-15 PBL and Watermark Levels

PBL (Number of transfers)	Tx/Rx Watermark Value
1	greater than or equal to 1
4	greater than or equal to 4
8	greater than or equal to 8
16	greater than or equal to 16
32	greater than or equal to 32
64	greater than or equal to 64
128	greater than or equal to 128
256	greater than or equal to 256

27.6.9 Variable Delay Usage

The control signals for variable delay usage are shown as follows.

27.6.9.1 SDMMC Variable Delay Usage

Table 27-16 Configuration for SDMMC Variable Delay Usage

Signal Name	Source	Default	Description
init_state	CRU_SDMMC_CON0[0]	0	Soft initial state for phase shift.
drv_degree[1:0]	CRU_SDMMC_CON0[2:1]	2	Phase shift for cclk_in_drv. 0: 0-degree

Signal Name	Source	Default	Description
			1: 90-degree 2: 180-degree 3: 270-degree
drv_delaynum[7:0]	CRU_SDMMC_CON0[10:3]	0	Element number in delay line for cclk_in_drv
drv_sel	CRU_SDMMC_CON0[11]	0	cclk_in_drv source selection: 0: use clock after phase_shift 1: use clock after phase_shift and delay line
sample_degree[1:0]	CRU_SDMMC_CON1[2:1]	0	Phase shift for cclk_in_sample. 0: 0-degree 1: 90-degree 2: 180-degree 3: 270-degree
sample_delaynum[7:0]	CRU_SDMMC_CON1[10:3]	0	Element number in delay line for cclk_in_sample
sample_sel	CRU_SDMMC_CON1[11]	0	cclk_in_sample source selection: 0: use clock after phase_shift 1: use clock after phase_shift and delay line

27.6.9.2 SDIO Variable Delay Usage

Table 27-17 Configuration for SDIO Variable Delay Usage

Signal Name	Source	Default	Description
init_state	CRU_SDIO_CON0[0]	0	Soft initial state for phase shift.
drv_degree[1:0]	CRU_SDIO_CON0[2:1]	2	Phase shift for cclk_in_drv. 0: 0-degree 1: 90-degree 2: 180-degree 3: 270-degree
drv_delaynum[7:0]	CRU_SDIO_CON0[10:3]	0	Element number in delay line for cclk_in_drv
drv_sel	CRU_SDIO_CON0[11]	0	cclk_in_drv source selection: 0: use clock after phase_shift 1: use clock after phase_shift and delay line
sample_degree[1:0]	CRU_SDIO_CON1[2:1]	0	Phase shift for cclk_in_sample. 0: 0-degree 1: 90-degree 2: 180-degree 3: 270-degree
sample_delaynum[7:0]	CRU_SDIO_CON1[10:3]	0	Element number in delay line for cclk_in_sample.
sample_sel	CRU_SDIO_CON1[11]	0	cclk_in_sample source selection: 0: use clock after phase_shift 1: use clock after phase_shift and delay line

27.6.9.3 EMMC Variable Delay Usage

Table 27-18 Configuration for EMMC Variable Delay Usage

Signal Name	Source	Default	Description
init_state	CRU_EMMC_CON0[0]	0	Soft initial state for phase shift.
drv_degree[1:0]	CRU_EMMC_CON0[2:1]	2	Phase shift for cclk_in_drv. 0: 0-degree 1: 90-degree 2: 180-degree 3: 270-degree

Signal Name	Source	Default	Description
drv_delaynum [7:0]	CRU_EMMC_CON0[10:3]	0	Element number in delay line for cclk_in_drv
drv_sel	CRU_EMMC_CON0[11]	0	cclk_in_drv source selection: 0: use clock after phase_shift 1: use clock after phase_shift and delay line
sample_degree[1:0]	CRU_EMMC_CON1[2:1]	0	Phase shift for cclk_in_sample. 0: 0-degree 1: 90-degree 2: 180-degree 3: 270-degree
sample_delaynum[7:0]	CRU_EMMC_CON1[10:3]	0	Element number in delay line for cclk_in_sample
sample_sel	CRU_EMMC_CON1[11]	0	cclk_in_sample source selection: 0: use clock after phase_shift 1: use clock after phase_shift and delay line

The following outlines the steps for clock generation sequence:

1) Assert init_state to soft reset the CLKGEN.
2) Configure drv_degree/sample_degree.
3) If fine adjustment required, delay line can be used by configuring drv_delaynum/sample_delaynum and drv_sel/sample_sel.
4) Dis-assert init_state to start CLKGEN.

27.6.10 Variable Delay Tuning

Tuning is defined by SD and MMC cards to determine the correct sampling point required for the host, especially for the speed modes SDR104 and HS200 where the output delays from the cards can be up to 2 UI. Tuning is required for other speed modes-such as DDR50-even though the output delay from the card is less than one cycle.

Command for tuning is different for different cards.

- SD Memory Card:
  - CMD19 – SD card for SDR50 and SDR104 speed modes. Tuning data is defined by card specifications.
  - CMD6 – SD card for speed modes not supporting CMD19. Tuning data is the 64byte SD status.
- Multimedia Card:
  - CMD21 – MMC card for HS200 speed mode. Tuning data is defined by card specifications.
  - CMD8 – MMC card for speed modes not supporting CMD21. Tuning data is 512 byte ExtCSD data.

The following is the procedure for variable delay tuning:

1) Set a phase shift of 0-degree on cclk_in_sample.
2) Send the Tuning command to the card; the card in turn sends an R1 response on the CMD line and tuning data on the DAT line.
3) If the host sees any of the errors—start bit error, data crc error, end bit error, data read time-out, response CRC error, response error—then the sampling point is incorrect.
4) Send CMD12 to bring the host controller state machines to idle.
   - The card may treat CMD12 as an invalid command because the card has successfully sent the tuning data, and it cannot send a response.
   - The host controller may generate a response time-out interrupt that must be cleared by software.
5) Repeat steps 2) to 4) by increasing the phase shift value or delay element number on cclk_in_sample until the correct sampling point is received such that the host does not see any of the errors.
6) Mark this phase shift value as the starting point of the sampling window.
7) Repeat steps 2 to 4 by increasing the phase shift value or delay element number on cclk_in_sample until the host sees the errors starting to come again or the phase shift

value reaches 360-degree.

8) Mark the last successful phase shift value as the ending point of the sampling window.

A window is established where the tuning block is matched. For example, for a scenario where the tuning block is received correctly for a phase shift window of 90-degree and 180-degree, then an appropriate sampling point is established as 135-degree. Once a sampling point is established, no errors should be visible in the tuning block.

27.6.11 Card Detection Method

There are many methods for SDMMC/SDIO device detection.

- Method1: Using SDMMC_CDETECT register, which is value on card_detect_n input port. 0 represents presence of card.
- Method2: Using card detection unit in Host Controller, outputting host interrupt. The card detection unit looks for any changes in the card-detect signals for card insertion or card removal. It filters out the debounces associated with mechanical insertion or removal, and generates one interrupt to the host. You can program the debounce filter value in SDMMC_DEBNCE [23:0]. Following figure illustrates the timing for card-detect signals.

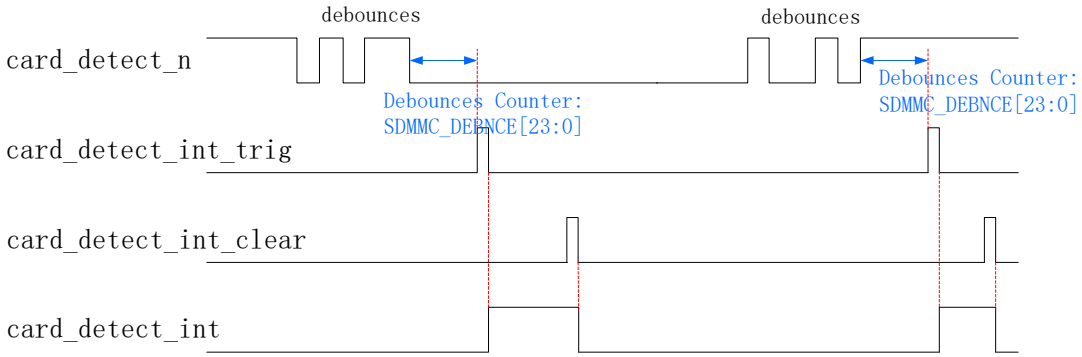


Fig. 27-18 Card Detection Method 2

- Method3: Using card detection unit in GRF, outputting sdmmc_detect_dual_edge_int connecting to IRQ[105]. Similar to Method2, except that the debounce time is configurable by PMUGRF_SD_DET_COUNT; and the insertion/removal detection interrupt can be enabled or cleared respectively. The detailed register information is:

Table 27-19 Register for SDMMC Card Detection Method 3

Signal Name	Source	Default	Description
sd_detectn_rise_edge_irq_en	PMUGRF_SD_DETECT_CON[0]	0	sdmmc detect_n signal rise edge interrupt enable. 1: enable 0: disable
sd_detect_fall_edge_detect_en	PMUGRF_SD_DETECT_CON[1]	0	sd_detect_falling_edge enable 0: disable 1: enable
sd_detect_rising_edge_dectect_status	PMUGRF_SD_DETECT_STATUS[0]	0	sd_detect_rising_edge status 0: disable 1: enable
sd_detect_fall_edge_detect_status	PMUGRF_SD_DETECT_STATUS[1]	0	sd_detect_falling_edge status 0: disable 1: enable
sd_detect_rising_edge_dectect_clr	PMUGRF_SD_DETECT_CLR[0]	0	sd_detect_rising_edge clear 0: disable 1: enable
sd_detect_fall_edge_detect_clr	PMUGRF_SD_DETECT_CLR[1]	0	sd_detect_falling_edge clear 0: disable 1: enable

- Method4: Using filtered card_detect_n with the debounce time PMUGRF_SD_DET_COUNT for interrupt source, connecting to IRQ [106] directly.

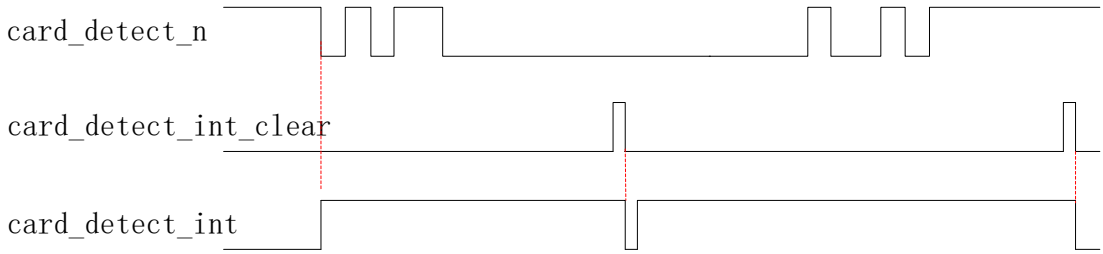


Fig. 27-19 Card Detection Method 4

27.6.12 SDMMC IOMUX With JTAG

The IO for sdmmc_cdata2/sdmmc_cdata3 is shared with jtag_tck/jtag_tms. The condition of usage for SDMMC or JTAG usage is as follows.

- If GRF_IOFUNC_SEL[3](grf_force_jtag) is equal to 1 and SD/MMC card is not detected within detection time(in the unit of XIN24M clock), the GPIOs are used for JTAG.
- Otherwise, the GPIOs' usage is defined by IOMUX configuration.

Chapter 28 USB2.0 OTG Controller

28.1 Overview

USB2.0 OTG Controller can act as static host, static device, USB2.0 A device or B device basing on the status of input ID from USB2.0 PHY. It can perform data transmission between host and device as host or device for High-Speed / Full-Speed / Low-Speed.
USB2.0 OTG controller supports the following features:

- General Features
  1. Compatible Specification
     - Universal Serial Bus Specification, Revision 2.0
     - eXtensible Host Controller Interface for Universal Serial Bus (xHCI), Revision 1.1
  2. Support Control/Bulk/Interrupt/Isochronous Transfer
  3. Descriptor caching and data pre-fetching used to improve system performance in high-latency systems
  4. Dynamic FIFO memory allocation for endpoints
  5. Keep-Alive feature in LS mode and (micro-)SOFs in HS/FS modes
  6. Low MIPS requirement
     - Driver involved only in setting up transfers and high-level error recovery
     - Hardware handles data packing and routing to a specific pipe
- Application Interface Features
  1. AHB Slave interface
  2. AXI Master interface
     - Programmable burst lengths up to 16
     - Handle fixed burst address alignment
     - Programmable number of outstanding read/write requests up to 16
- USB2.0 Device Features
  1. Up to 7 IN endpoints, including control endpoint 0
  2. Up to 6 OUT endpoints, including control endpoint 0
  3. Up to 13 endpoint transfer resources, each one for each endpoint
  4. Flexible endpoint configuration for multiple applications/USB set-configuration modes
  5. Isochronous endpoints with isochronous data in data buffers
  6. Flexible Descriptor with rich set of features to support buffer interrupt moderation, multiple transfers, isochronous, control, and scattered buffering support
- USB Class-Specific Device Features
  1. Gathering of scattered packet to support Ethernet Over USB
  2. Scheduling of multiple Ethernet packets without interrupt
  3. Variable FIFO buffer allocation for each endpoint
  4. For isochronous applications, scheduling of variable-length payloads for each microframe
  5. Microframe precise scheduling for isochronous applications
  6. Configurable endpoint type selection and dynamic FIFO allocation to facilitate multi-function/composite device implementation. During set-config or alternate-setting, device resources are reconfigured to meet the configuration or alternate setting requirements
- USB 2.0 Host Features
  1. Support up to 64 devices
  2. Support 1 interrupter
  3. Support 1 USB2.0 port
- USB 2.0 Dual-Role Device (DRD) Features
  1. Static Device Operation
  2. Static Host Operation
  3. USB2.0 OTG A device and B device basing on ID
  4. Not support USB2.0 OTG session request protocol(SRP), host negotiation protocol(HNP) and Role Swap Protocol(RSP)

28.2 Block Diagram

USB2.0 OTG Controller comprises with:

- Bus Interface/List Management: Register Interface/Data and Descriptors DMA management
- HS/FS/LS MAC : USB2.0 part logic

USB2.0 PHY: UTMI+ interface USB2.0 PHY

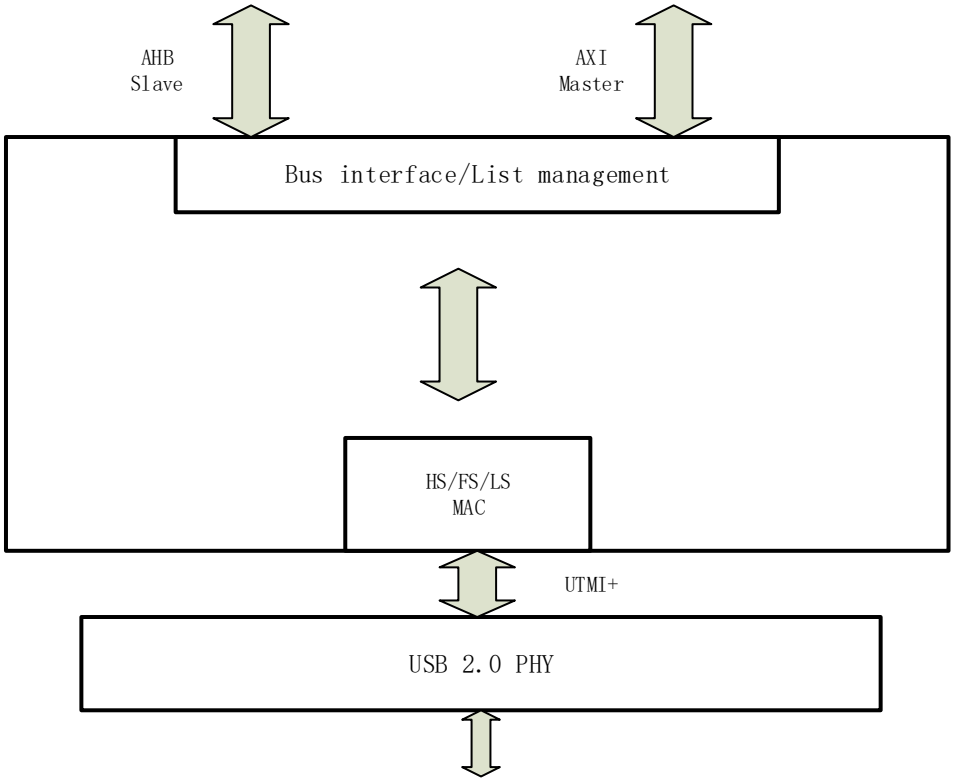


Fig. 28-1 USB2.0 OTG Block Diagram

28.3 Function Description

As a USB2.0 OTG controller, it can act as static host controller, static device controller, 2.0 OTG A device or B device basing on ID of USB2.0 PHY.

As device controller, it can work on USB2.0 speed and process USB transactions described in the descriptors (read back from external memory by AXI master) to/from UTMI+ interface PHY.

As host controller, it can work on USB2.0 speed and process USB transactions described in the descriptors (read back from external memory by AXI master) to/from UTMI+ interface.

28.4 Register Description

28.4.1 Internal Address Mapping

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

Table 28-1 USB2 Address Mapping

Offset Address Range	Register Type
0x00000 ~ 0x07FFF	xHCI Registers, see xHCI spec.
0x0C100 ~ 0x0C6FF	Global Registers
0x0C700 ~ 0x0CBFF	Device Controller Registers
0x0CC00 ~ 0x0CFFF	Unused/Reserved
0x40000 ~ 0x7FFFF	Internal RAM0 – Debug Access (256KB)

28.4.2 Registers Summary

Name	Offset	Size	Reset Value	Description
usb2otg_GSBUSCFG0	0xC100	W	0x00000001	Global SoC Bus Configuration Register 0
usb2otg_GSBUSCFG1	0xC104	W	0x00000300	Global SoC Bus Configuration Register 1
usb2otg_GRXTHRCFG	0xC10C	W	0x00000000	Global Rx Threshold Control Register
usb2otg_GCTL	0xC110	W	0x30C12004	Global Core Control Register
usb2otg_GSTS	0xC118	W	0x7E800000	Global Status Register
usb2otg_GUCTL1	0xC11C	W	0x0404018A	Global User Control Register 1
usb2otg_GSNPSID	0xC120	W	0x5533300A	Global SNPS ID Register
usb2otg_GGPIO	0xC124	W	0x00000000	Global General Purpose Input/Output Register
usb2otg_GUID	0xC128	W	0x20190520	Global User ID Register
usb2otg_GUCTL	0xC12C	W	0x02000010	Global User Control Register
usb2otg_GBUSERRADDRLO	0xC130	W	0x00000000	Global SoC Bus Error Address Register - Low
usb2otg_GBUSERRADDRHI	0xC134	W	0x00000000	Global SoC Bus Error Address Register - High
usb2otg_GHWPARAMS0	0xC140	W	0x2020400A	Global Hardware Parameters Register 0
usb2otg_GHWPARAMS1	0xC144	W	0x0120C93B	Global Hardware Parameters Register 1
usb2otg_GHWPARAMS2	0xC148	W	0x20190520	Global Hardware Parameters Register 2
usb2otg_GHWPARAMS3	0xC14C	W	0x069CD084	Global Hardware Parameters Register 3
usb2otg_GHWPARAMS4	0xC150	W	0x47822010	Global Hardware Parameters Register 4
usb2otg_GHWPARAMS5	0xC154	W	0x04204108	Global Hardware Parameters Register 5
usb2otg_GHWPARAMS6	0xC158	W	0x09D78020	Global Hardware Parameters Register 6
usb2otg_GHWPARAMS7	0xC15C	W	0x00000000	Global Hardware Parameters Register 7
usb2otg_GDBGFIFOSPACE	0xC160	W	0x000A0000	Global Debug Queue/FIFO Space Available Register
usb2otg_GDBGLNMCC	0xC168	W	0x00000000	Global Debug LNMCC Register
usb2otg_GDBGBMU	0xC16C	W	0x20300000	Global Debug BMU Register
usb2otg_GDBGLSPMUX	0xC170	W	0x003F0000	Global Debug LSP MUX Register - Device
usb2otg_GDBGLSP	0xC174	W	0x00000000	Global Debug LSP Register
usb2otg_GDBGEPINFO0	0xC178	W	0x00000801	Global Debug Endpoint Information Register 0
usb2otg_GDBGEPINFO1	0xC17C	W	0x00800000	Global Debug Endpoint Information Register 1
usb2otg_GPRTBIMAP_HSLO	0xC180	W	0x00000000	Global High-Speed Port to Bus Instance Mapping Register - Low
usb2otg_GPRTBIMAP_FSLO	0xC188	W	0x00000000	Global Full-Speed Port to Bus Instance Mapping Register - Low
usb2otg_GUSB2PHYCFG0	0xC200	W	0x00101408	Global USB2 PHY Configuration Register 0

Name	Offset	Size	Reset Value	Description
usb2otg_GTXFIFOSIZn	0xC300	W	0x04B9000A	Global Transmit FIFO Size Register n, offset (0xc300 + 4*n), n=0~6
usb2otg_GRXFIFOSIZn	0xC380	W	0x03340185	Global Receive FIFO Size Register n,
usb2otg_GEVNTADRLO0	0xC400	W	0xFF7001B0	Global Event Buffer Address (Low) Register 0
usb2otg_GEVNTADRHI0	0xC404	W	0x00000000	Global Event Buffer Address (High) Register 0
usb2otg_GEVNTSIZ0	0xC408	W	0x80000100	Global Event Buffer Size Register 0
usb2otg_GEVNTCOUNT0	0xC40C	W	0x00000004	Global Event Buffer Count Register 0
usb2otg_GHWPARAMS8	0xC600	W	0x0000047C	Global Hardware Parameters Register 8
usb2otg_GTXFIFOPRIDEV	0xC610	W	0x00000000	Global Device TX FIFO DMA Priority Register
usb2otg_GTXFIFOPRIHST	0xC618	W	0x00000000	Global Host TX FIFO DMA Priority Register
usb2otg_GRXFIFOPRIHST	0xC61C	W	0x00000000	Global Host RX FIFO DMA Priority Register
usb2otg_GFIFOPRIDBC	0xC620	W	0x00000000	Global Host Debug Capability DMA Priority Register
usb2otg_GDMAHLRATIO	0xC624	W	0x00000000	Global Host FIFO DMA High-Low Priority Ratio Register
usb2otg_GFLADJ	0xC630	W	0x00000000	Global Frame Length Adjustment Register
usb2otg_DCFG	0xC700	W	0x04080928	Device Configuration Register
usb2otg_DCTL	0xC704	W	0x80000A00	Device Control Register
usb2otg_DEVTEN	0xC708	W	0x0000121F	Device Event Enable Register
usb2otg_DSTS	0xC70C	W	0x00038118	Device Status Register
usb2otg_DGCMDPAR	0xC710	W	0x00000000	Device Generic Command Parameter Register
usb2otg_DGCMD	0xC714	W	0x00000000	Device Generic Command Register
usb2otg_DALEPENA	0xC720	W	0x00000003	Device Active USB Endpoint Enable Register
usb2otg_DEPnCMDPAR2	0xC800	W	0x00000000	Device Physical Endpoint-n Command Parameter 2 Register, offset (0xc800 + 4*n), n=0~12
usb2otg_DEPnCMDPAR1	0xC804	W	0xFF7002E0	Device Physical Endpoint-n Command Parameter 1 Register, offset (0xc804 + 4*n), n=0~12
usb2otg_DEPnCMDPAR0	0xC808	W	0x00000006	Device Physical Endpoint-n Command Parameter 0 Register, offset (0xc808 + 4*n), n=0~12
usb2otg_DEPnCMD	0xC80C	W	0x00000000	Device Physical Endpoint-n Command Register, offset (0xc80c + 4*n), n=0~12

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

28.4.3 Detail Registers Description

usb2otg_GSBUSCFG0

Address: Operational Base + offset (0xC100)

Bit	Attr	Reset Value	Description
31:28	RW	0x0	datrdreqinfo AXI-cache for Data Read (DatRdReqInfo).
27:24	RW	0x0	desrdreqinfo AXI-cache for Descriptor Read (DesRdReqInfo).
23:20	RW	0x0	datwrreqinfo AXI-cache for Data Write (DatWrReqInfo).
19:16	RW	0x0	deswrreqinfo AXI-cache for Descriptor Write (DesWrReqInfo).
15:12	RO	0x0	reserved
11	RW	0x0	datbigend This bit controls the endian mode for data accesses. 1'b0: Little-endian (default) 1'b1: Big-endian
10	RW	0x0	desbigend This bit controls the endian mode for descriptor accesses. 1'b0: Little-endian (default) 1'b1: Big-endian
9:8	RO	0x0	reserved
7	RW	0x0	incr256brstena If software set this bit to 1, the AXI master uses INCR to do the 256-beat burst.
6	RW	0x0	incr128brstena If software set this bit to 1, the AXI master uses INCR to do the 128-beat burst.
5	RW	0x0	incr64brstena If software set this bit to 1, AXI master uses INCR to do the 64-beat burst.
4	RW	0x0	incr32brstena If software set this bit to 1, the AXI master uses INCR to do the 32-beat burst.
3	RW	0x0	incr16brstena If software set this bit to 1, the AXI master uses INCR to do the 16-beat burst.
2	RW	0x0	incr8brstena If software set this bit to 1, the AXI master uses INCR to do the 8-beat burst.
1	RW	0x0	incr4brstena When this bit is enabled the controller is allowed to do bursts of beat length 1, 2, 3, and 4. It is highly recommended that this bit is enabled to prevent descriptor reads and writes from being broken up into separate transfers.

Bit	Attr	Reset Value	Description
0	RW	0x1	incrbrstena This bit determines the set of burst lengths the master interface uses. It works in conjunction with the GSBUSCFG0[7:1] enables(INCR256/128/64/32/16/8/4). 0: INCRX burst mode ARLEN/AWLEN do not use INCR. They use only the following burst lengths: 1; 4 (if GSBUSCFG0.INCR4BrstEna = 1); 8 (if GSBUSCFG0.INCR8BrstEna = 1); 16 (if GSBUSCFG0.INCR16BrstEna = 1); 32 (if GSBUSCFG0.INCR32BrstEna = 1); 64 (if GSBUSCFG0.INCR64BrstEna = 1); 128 (if GSBUSCFG0.INCR128BrstEna = 1); 256 (if GSBUSCFG0.INCR256BrstEna = 1); 1: INCR (undefined length) burst mode; ARLEN/AWLEN uses any length less than or equal to the largest-enabled burst length of INCR4/8/16/32/64/128/256. For cache line-aligned applications, this bit is typically set to 0 to ensure that the master interface uses only power-of-2 burst lengths (as enabled via GSBUSCFG0[7:0]).

usb2otg_GSBUSCFG1

Address: Operational Base + offset (0xC104)

Bit	Attr	Reset Value	Description
31:13	RO	0x00000	reserved
12	RW	0x0	en1kpage 1K Page Boundary Enable. By default (this bit is disabled) the AXI breaks transfers at the 4k page boundary. When this bit is enabled, the AXI master (DMA data) breaks transfers at the 1k page boundary.
11:8	RW	0x3	pipe_trans_limit AXI Pipelined Transfers Burst Request Limit. The field controls the number of outstanding pipelined transfer requests the AXI master pushes to the AXI slave. When the AXI master reaches this limit, it does not make any more requests on the AXI ARADDR and AWADDR buses until the associated data phases complete. This field is encoded as follows: 0: 1 request 1: 2 requests 2: 3 requests 3: 4 requests ... F: 16 requests
7:0	RO	0x00	reserved

usb2otg_GRXTHRCFG

Address: Operational Base + offset (0xC10C)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved

Bit	Attr	Reset Value	Description
29	RW	0x0	usb_rx_pkt_cntsel USB ReceivePacket Count Enable. This field enables/disables the USB reception multi-packet thresholding: 1'b0: The core can only start reception on the USB when the RX FIFO has space for at least one packet. 1'b1: Not applied. If you are using external buffer control (EBC) feature, disable this mode by setting USBRxPktCntSel to 0.
28:13	RO	0x0000	reserved
12:0	RW	0x0000	resvl_socout_spc Space reserved in rx fifo for ISOC OUT. In host mode, this field is not applicable and must be programmed to 0. In device mode, this value represents the amount of space to be reserved for ISOC OUT packets. The value to be programmed should be chosen so as to ensure that non ISOC packets are not completely dropped, If no space needs to be reserved for ISOC OUT packets, program this field to 0. This field is valid only in device mode. The maximum configurable depth of rx fifo is 8192. Therefore, this field is 13 bits wide. For HS/FS, the space reservation is the actual value.

usb2otg_GCTL

Address: Operational Base + offset (0xC110)

Bit	Attr	Reset Value	Description
31:19	RW	0x0618	pwrdnscale Not applied.
18	RW	0x0	masterfiltbypass Master Filter Bypass When this bit is set to 1'b1, all the filters are bypassed. The double synchronizers to mac_clk preceding the filters are also bypassed. For enabling the filters, this bit must be 1'b0.
17	RW	0x0	bypassetaddr Bypass SetAddress in Device Mode. When BYPSSETADDR bit is set, the device core uses the value in the DCFG[DevAddr] bits directly for comparing the device address in the tokens. For simulation, you can use this feature to avoid sending an actual SET ADDRESS control transfer on the USB, and make the device core respond to a new address. Note: You can set this bit for simulation purposes only. In the actual hardware, this bit must be set to 1'b0.
16	RW	0x1	u2rstecn Device controller on USB 2.0 reset checks for receiver termination eight times per attempt if this bit is set to zero, or only once per attempt if the bit is set to one. Note: This bit is applicable only in device mode.

Bit	Attr	Reset Value	Description
15:14	RW	0x0	frmscldwn This field scales down device view of a SOF/USOF duration. For HS mode: Value of 2'h3 implements interval to be 15.625 us Value of 2'h2 implements interval to be 31.25 us Value of 2'h1 implements interval to be 62.5 us Value of 2'h0 implements interval to be 125us For FS mode, the scale-down value is multiplied by 8. This field also scales down the MaxPacketSize of the IN and OUT bulk endpoint to allow more traffic during simulation. It can only be changed from a non-zero value during simulation. 2'h0: 1024 bytes 2'h1: 512 bytes 2'h2: 256 bytes 2'h3: 128 bytes
13:12	RW	0x2	prtcapdir PRTCAPDIR: Port Capability Direction (PrtCapDir) 2'b01: for Host configurations 2'b10: for Device configurations SW should base on IDDIG input to set usb controller as an OTG 2.0 device with A-device or B-device.
11	RW	0x0	coresoftreset Core Soft Reset (CoreSoftReset) 1'b0: No soft reset; 1'b1: Soft reset to core Clears the interrupts and all the CSRs except the following registers: GCTL; GUCTL; GSTS; GSNPSID; GGPIO; GUID; GUSB2PHYCFGn registers; GUSB3PIPECTLn registers; DCFG; DCTL; DEVTEN; DSTS.
10	RW	0x0	sofitpsync Not applied.
9	RW	0x0	u1u2_timescale Not applied.
8	RW	0x0	debugattach Debug Attach. Not applied.
7:6	RW	0x0	ramclksel RAM Clock Select (RAMClkSel) 2'b00: bus clock 2'b01: pipe clock (Only used in device mode) 2'b10: In device mode, pipe/2 clock. In Host mode, controller switches ram_clk between pipe/2 clock, mac2_clk and bus_clk based on the status of the U2 ports 2'b11: In device mode, selects mac2_clk as ram_clk (when 8-bit UTMI or ULPI used. Not supported in 16-bit UTMI mode); In Host mode, controller switches ram_clk between pipe_clk, mac2_clk and bus_clk based on the status of the U2 ports. In device mode, upon a USB reset and USB disconnect, the hardware clears these bits to 2'b00.

Bit	Attr	Reset Value	Description
5:4	RW	0x0	scaledown Scale-Down Mode (ScaleDown) When Scale-Down mode is enabled for simulation, the core uses scaled-down timing values, resulting in faster simulations. When Scale-Down mode is disabled, actual timing values are used. This is required for hardware operation. HS/FS/LS Modes: 2'b00: Disables all scale-downs. Actual timing values are used. 2'b01: Enables scale-down of all timing values except Device mode suspend and resume. These include Speed enumeration, HNP/SRP, and Host mode suspend and resume 2'b10: Enables scale-down of Device mode suspend and resume timing values only. 2'b11: Enables bit 0 and bit 1 scale-down timing values.
3	RW	0x0	disscramble Disable Scrambling (DisScramble) Transmit request to Link Partner on next transition to Recovery or Polling.
2	RW	0x1	u2exit_lfps If this bit is: 1'b0: the link treats 248ns LFPS as a valid U2 exit. 1'b1: the link waits for 8us of LFPS before it detects a valid U2 exit. This bit is added to improve interoperability with a third party host controller. This host controller in U2 state while performing receiver detection generates an LFPS glitch of about 4ms duration. This causes the device to exit from U2 state because the LFPS filter value is 248ns. With the new functionality enabled, the device can stay in U2 while ignoring this glitch from the host controller.
1	RO	0x0	gbl_hibernation_en This bit enables hibernation at the global level. If hibernation is not enabled through this bit, the PMU immediately accepts the D0->D3 and D3->D0 power state change requests, but does not save or restore any core state. In addition, the PMUs never drive the PHY interfaces and let the core continue to drive the PHY interfaces.
0	RW	0x0	dsblclkgtng Disable Clock Gating. This bit is set to 1 and the core is in Low Power mode, internal clock gating is disabled. You can set this bit to 1'b1 after Power On Reset.

usb2otg_GSTS

Address: Operational Base + offset (0xC118)

Bit	Attr	Reset Value	Description
31:20	RO	0x7e8	cbelt Current BELT Value. In Host mode, this field indicates the minimum value of all received device BELT values and the BELT value that is set by the Set Latency Tolerance Value command.
19:12	RO	0x00	reserved
11	RO	0x0	SSIC_IP Not applied.

Bit	Attr	Reset Value	Description
10	RO	0x0	OTG_IP OTG Interrupt Pending. This field indicates that there is a pending interrupt pertaining to OTG in OEVT register.
9	RO	0x0	bc_ip Battery Charger Interrupt Pending This field indicates that there is a pending interrupt pertaining to BC in BCEVT register.
8	RO	0x0	adp_ip ADP Interrupt Pending. This field indicates that there is a pending interrupt pertaining to ADP in ADPEVT register.
7	RO	0x0	host_ip Host Interrupt Pending. This field indicates that there is a pending interrupt pertaining to xHC in the Host event queue.
6	RO	0x0	device_ip Device Interrupt Pending This field indicates that there is a pending interrupt pertaining to peripheral (device) operation in the Device event queue.
5	W1C	0x0	csr_timeout When this bit is 1'b1, it indicates that the software performed a write or read to a core register that could not be completed within DWC_USB3_CSR_ACCESS_TIMEOUT bus clock cycles (default: h1FFFF).
4	W1C	0x0	buserraddrvld us Error Address Valid Indicates that the GBUSERRADDR register is valid and reports the first bus address that encounters a bus error.
3:2	RO	0x0	reserved
1:0	RO	0x0	curmod Current Mode of Operation.

usb2otg_GUCTL1

Address: Operational Base + offset (0xC11C)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved

Bit	Attr	Reset Value	Description
29	RW	0x0	filter_se0_fsls_eop 1'b0: Default behaviour, no change in Linestate check for SE0 detection in FS/LS. 1'b1: Feature enabled, FS/LS SE0 is filtered for 2 clocks for detecting EOP. This bit is applicable for FS/LS operation. If this feature is enabled, then SE0 on the linestate is validated for 2 consecutive utmi/ulpi clock edges for EOP detection. This feature is applicable only in FS in device mode and FS/LS mode of operation in host mode. Device mode: FS - If GUCTL1.FILTER_SE0_FSLS_EOP is set, then for device LPM hanshake, the core will ignore single SE0 glitch on the linestate during transmit. Only 2 or more SE0 is considered as a valid EOP on FS. Host mode: FS/LS - If GUCTL1.FILTER_SE0_FSLS_EOP is set, then the core will ignore single SE0 glitch on the linestate during transmit. Only 2 or more SE0 is considered as a valid EOP on FS/LS port. Enable this feature if the LineState has SE0 glitches during transmission. This bit is quasi-static, i.e., should not be changed during device operation.
28	RW	0x0	tx_ipgap_linecheck_dis 1'b0: Default behaviour, no change in Linestate check. 1'b1: Feature enabled, 2.0 MAC disables Linestate check during HS transmit. This bit is applicable for HS operation of u2mac. If this feature is enabled, then the 2.0 mac operating in HS ignores the UTMI/ULPI Linestate during the transmit of a token (during token-to-token and token-to-data IPGAP). When enabled, the controller implements a fixed 40-bit TxEndDelay after the packet is given on UTMI and ignores the Linestate during this time. This feature is applicable only in HS mode of operation. Device mode: If GUCTL1.TX_IPGAP_LINECHECK_DIS is set, then for device LPM handshake, the core will ignore the linestate after TX and wait for a fixed clocks (40 bit times equivalent) after transmitting ACK on utmi. Host mode: If GUCTL1.TX_IPGAP_LINECHECK_DIS is set, then the ipgap between (tkn to tkn/data) is added by 40 bit times of TXENDDELAY, and linestate is ignored during this 40 bit times delay. Enable this bit if the LineState will not reflect the expected line state (J) during transmission. This bit is quasi-static, i.e., should not be changed during device operation.
27	RW	0x0	dev_trb_out_spr_ind 1'b0: Default behaviour, no change in TRB status dword. 1'b1: Feature enabled, OUT TRB status indicates Short Packet. This bit is applicable for device mode only (and ignored in host mode). If the device application (SW/HW) wants to know if a short packet was received for an OUT in the TRB status itself, then this feature can be enabled, so that a bit is set in the TRB writeback in the buf_size dword. Bit[26] - SPR of the trbstatus, RSVD, SPR,PCM1, bufsize dword will be set during an OUT transfer TRB write back if this is the last TRB used for that transfer descriptor. This bit is quasi-static, i.e., should not be changed during device operation.
26	RW	0x1	dve_force_20clk_for_30clk Not applied.

Bit	Attr	Reset Value	Description
25	RW	0x0	p3_in_u2 Not applied.
24	RW	0x0	dev_l1_exit_by_hw 1'b0: Default behaviour, disables device L1 hardware exit logic. 1'b1: feature enabled This bit is applicable for device mode (2.0) only. This field enables device controller sending remote wakeup for L1 if the device becomes ready for sending/accepting data when in L1 state. If the host expects the device to send remote wkp signalling to resume after going into L1 in flow controlled state, then this bit can be set to send the remote wake signal automatically when the device controller becomes ready. This HW remote wake feature is applicable only to bulk and interrupt transfers, and not for Isoch/Control When control transfers are in progress, the LPM will be rejected (NYET response). Only after control transfers are completed (either with ACK/STALL), LPM will be accepted For Isoch transfers, the host needs to do the wake-up and start the transfer. Device controller will not do remote-wakeup when Isoch endpoints get ready. The device SW needs to keep the GUSB2PHYCFG[EnblSlpM] reset in order to keep the PHY clock to be running for keeping track of SOF intervals. When L1 hibernation is enabled, the controller will not do automatic exit for hibernation requests thru L1. This bit is quasi-static, i.e., should not be changed during device operation.
23:21	RW	0x0	ip_gap_add_on This register field is used to add on to the default inter packet gap setting in the USB 2.0 MAC.
20	RW	0x0	dev_lsp_tail_lock_dis 1'b0: Default behaviour, enables device lsp lock logic for tail TRB update. 1'b1: Fix disabled This is a bug fix for STAR 9000716195 that affects the CSP mode for OUT endpoints in device mode. The issue is that tail TRB index is not synchronized with the cache Scratchpad bytecount update. If the fast-forward request comes in-between the bytecount update on a newly fetched TRB and the tail-index write update in TPF, the RDP works on an incorrect tail index and misses the byte count decrement for the newly fetched TRB in the fast-forwarding process. This fix needs to be present all the times.
19	RW	0x0	nak_per_enh_fs 1'b1: Enables performance enhancement for FS async endpoints in the presence of NAKs. 1'b0: Enhancement not applied. If a periodic endpoint is present, and if a bulk endpoint which is also active is being NAKed by the device, then this could result in a decrease in performance of other Full Speed bulk endpoint which is ACked by the device. Setting this bit to 1, will enable the host controller to schedule more transactions to the async endpoints (bulk/ control) and hence will improve the performance of the bulk endpoint. This control bit should be enabled only if the existing performance with the default setting is not sufficient for your FullSpeed application. Setting this bit will only control, and is only required for Full Speed transfers.

Bit	Attr	Reset Value	Description
18	RW	0x1	nak_per_enh_hs 1'b1: Enables performance enhancement for HS async endpoints in the presence of NAKs. 1'b0: Enhancement not applied. If a periodic endpoint is present, and if a bulk endpoint which is also active is being NAKed by the device, then this could result in a decrease in performance of other High Speed bulk endpoint which is ACked by the device. Setting this bit to 1, will enable the host controller to schedule more transactions to the async endpoints (bulk/ control) and hence will improve the performance of the bulk endpoint. This control bit should be enabled only if the existing performance with the default setting is not sufficient for your HighSpeed application. Setting this bit will only control, and is only required for High Speed transfers.
17	RW	0x0	parkmode_disbale_ss Not applied.
16	RW	0x0	parkmode_disable_hs This bit is used only in host mode. When this bit is set to 1 all HS bus instances park mode are disabled. To improve performance in park mode, the xHCI scheduler queues in three requests of 4 packets each for High Speed asynchronous endpoints in a micro-frame. But if a device is slow and if it NAKs more than 3 times, then it is rescheduled only in the next micro-frame. This could decrease the performance of a slow device even further. In a few high speed devices (such as Sandisk Cruzer Blade 4GB VID: 1921, PID: 21863 and Flex Drive VID: 3744, PID: 8552) when an IN request is sent within 900ns of the ACK of the previous packet, these devices send a NAK. When connected to these devices, if required, the software can disable the park mode if you see performance drop in your system. When park mode is disabled, pipelining of multiple packet is disabled and instead one packet at a time is requested by the scheduler. This allows up to 12 NAKs in a micro-frame and improves performance of these slow devices.
15	RW	0x0	parkmod_disavale_fsls This bit is used only in host mode, and is for debug purpose only. When this bit is set to 1 all FS/LS bus instances in park mode disabled.
14:9	RO	0x00	reserved

Bit	Attr	Reset Value	Description
8	RW	0x1	l1_susp_thrld_en_for_host This bit is used only in host mode. The host controller asserts the utmi_l1_suspend_n and utmi_sleep_n output signals (see LPM Interface Signals table in the Databook) as follows: The controller asserts the utmi_l1_suspend_n signal to put the PHY into deep low-power mode in L1 when both of the following are true: The HIRD/BESL value used is greater than or equal to the value in L1_SUSP_THRLD_FOR_HOST field. The L1_SUSP_THRLD_EN_FOR_HOST bit is set to 1'b1. The controller asserts utmi_sleep_n on L1 when one of the following is true: The HIRD/BESL value used is less than the value in L1_SUSP_THRLD_FOR_HOST field. The L1_SUSP_THRLD_EN_FOR_HOST bit is set to 1'b0.
7:4	RW	0x8	l1_susp_thrld_for_host This field is effective only when the L1_SUSP_THRLD_EN_FOR_HOST bit is set to 1. For more details, refer to the description of the L1_SUSP_THRLD_EN_FOR_HOST bit.
3	RW	0x1	hc_errata_enable Not applied.
2	RW	0x0	hc_parchk_disable Host Parameter Check Disable. When this bit is set to 0 (by default), the xHC checks that the input slot/EP context fields comply to the xHCI Specification. Upon detection of a parameter error during command execution, the xHC generates an event TRB with completion code indicating PARAMETER ERROR. When the bit is set to 1, the xHC does not perform parameter checks and does not generate PARAMETER ERROR completion code.
1	RW	0x1	ovrld_l1_susp_com If this bit is set, the utmi_l1_suspend_com_n is overloaded with the utmi_sleep_n signal. This bit is usually set if the PHY stops the port clock during L1 sleep condition.
0	RW	0x0	loa_filter_en If this bit is set, the USB 2.0 port babble is checked at least three consecutive times before the port is disabled. This prevents false triggering of the babble condition when using low quality cables. Note: This bit is valid only in host mode.

usb2otg_GSNPSID

Address: Operational Base + offset (0xC120)

Bit	Attr	Reset Value	Description
31:0	RO	0x5533300a	snpsid SNPSID[31:16] indicates Core Identification Number. 0x5533 is ASCII for U3 (DWC_usb3). SNPSID[15:0] indicates the release number. Current Release is 3.00a. Software uses this register to configure release-specific features in the driver.

usb2otg_GGPIO

Address: Operational Base + offset (0xC124)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	gpo General Purpose Output This field's value is driven out on the gp_out[15:0] core output port.
15:0	RO	0x0000	gpi General Purpose Input This field's read value reflects the gp_in[15:0] core input value.

usb2otg_GUID

Address: Operational Base + offset (0xC128)

Bit	Attr	Reset Value	Description
31:0	RW	0x20190520	userid Application-programmable ID field.

usb2otg_GUCTL

Address: Operational Base + offset (0xC12C)

Bit	Attr	Reset Value	Description
31:22	RW	0x008	refclkper This field indicates in terms of nano seconds the period of ref_clk. The default value of this register is set to 'h8 (8ns/125 MHz). This field needs to be updated during power-on initialization, if GCTL.SOFITPSYNC or GFLADJ.GFLADJ_REFCLK_LPM_SEL is set to 1. The programmable maximum value is 62ns, and the minimum value is 8ns. You must use a reference clock with a period that is an integer multiple, so that ITP can meet the jitter margin of 32ns. The allowable ref_clk frequencies whose period is not integer multiples are 16/17/19.2/24/39.7MHz. This field must not be set to 0 at any time. If you never plan to use this feature, then set this field to 'h8, the default value.
21	RW	0x0	no_extr_di No Extra Delay Between SOF and the First. Some HS devices misbehave when the host sends a packet immediately after a SOF. However, adding an extra delay between a SOF and the first packet can reduce the USB data rate and performance. This bit is used to control whether the host must wait for 2 microseconds before it sends the first packet after a SOF, or not. User can set this bit to one to improve the performance if those problematic devices are not a concern in the user's host environment. 1'b0: Host waits for 2 microseconds after a SOF before it sends the first USB packet. 1'b1: Host doesn't wait after a SOF before it sends the first USB packet.
20:18	RO	0x0	reserved
17	RW	0x0	sprs_ctrl_trans_en Sparse Control Transaction Enable. Some devices are slow in responding to Control transfers. Scheduling multiple transactions in one microframe/frame can cause these devices to misbehave. If this bit is set to 1'b1, the host controller schedules transactions for a Control transfer in different microframes/frames.

Bit	Attr	Reset Value	Description
16	RW	0x0	res_bw_hs_esp Reserving 85% Bandwidth for HS Periodic EPs. By default, HC reserves 80% of the bandwidth for periodic EPs. If this bit is set, the bandwidth is relaxed to 85% to accommodate two high speed, high bandwidth ISOC EPs. USB 2.0 required 80% bandwidth allocated for ISOC traffic. If two High-bandwidth ISOC devices (HD Webcams) are connected, and if each requires 1024-bytes X 3 packets per Micro-Frame, then the bandwidth required is around 82%. If this bit is set, then it is possible to connect two Webcams of 1024bytes X 3 payload per Micro-Frame each. Otherwise, you may have to reduce the resolution of the Webcams. This bit is valid in Host and DRD configuration and is used in host mode operation only. Ignore this bit in device mode.
15	RW	0x0	cm_dev_addr Compliance Mode for Device Address. When this bit is 1'b1, Slot ID may have different value than Device Address if max_slot_enabled < 128. 1'b1: Increment Device Address on each Address Device command. 1'b0: Device Address is equal to Slot ID. The xHCI compliance requires this bit to be set to 1. The 0 mode is for debug purpose only. This allows you to easily identify a device connected to a port in the Lecroy or Eliisys trace during hardware debug. This bit is valid in Host and DRD configuration and is used in host mode operation only. Ignore this bit in device mode.
14	RW	0x0	usb_host_in_auto_retry_en Host IN Auto Retry. 1'b0: Auto Retry Disabled 1'b1: Auto Retry Enabled Note: This bit is also applicable to the device mode.
13	RW	0x0	en_overlap_chk Not applied.

Bit	Attr	Reset Value	Description
12	RW	0x0	ext_cap_suppt_en External Extended Capability Support Enable When set, this field enables extended capabilities to be implemented outside the core. When the ExtCapSupEN is set and the Debug Capability is enabled, the Next Capability pointer in Debug Capability returns 16. A read to the first DWORD of the last internal extended capability (the "xHCI Supported Protocol Capability for USB 3.0" when the Debug Capability is not enabled) returns a value of 4 in the Next Capability Pointer field. This indicates to software that there is another capability four DWORDs after this capability (for example, at address N+16 where N is the address of this DWORD). If enabled, an external address decoder that snoops the xHC slave interface must be implemented. If it sees an access to N+16 or greater, the slave access is re-routed to a piece of hardware which returns the external capability pointer register of the new capability and also handles reads/writes to this new capability and the side effects. If disabled, a read to the first DWORD of the last internal extended capability returns 0 in the 'Next Capability Pointer' field. This indicates there are no more capabilities.
11	RW	0x0	insrt_extr_fsbodi Insert Extra Delay Between FS Bulk OUT. Some FS devices are slow to receive Bulk OUT data and can get stuck when there are consecutive Bulk OUT transactions with short inter-transaction delays. This bit is used to control whether the host inserts extra delay between consecutive Bulk OUT transactions to a FS Endpoint. 1'b0: Host doesn't insert extra delay between consecutive Bulk OUT transactions to a FS Endpoint. 1'b1: Host inserts about 12us extra delay between consecutive Bulk OUT transactions to a FS Endpoint to work around the device issue. Note: Setting this bit to one will reduce the Bulk OUT transfer performance for most of the FS devices.
10:9	RW	0x0	dtct Device Timeout Coarse Tuning. This field is a Host mode parameter which determines how long the host waits for a response from device before considering a timeout. The core first checks the DTCT value. If it is 0, then the timeout value is defined by the DTFT. If it is non-zero, then it uses the following timeout values: 2'b00: 0 usec -> use DTFT value instead 2'b01: 500 usec 2'b10: 1.5 msec 2'b11: 6.5 msec

Bit	Attr	Reset Value	Description
8:0	RW	0x010	ctft Device Timeout Fine Tuning. This field is a Host mode parameter which determines how long the host waits for a response from device before considering a timeout. For the DTFT field to take effect, DTCT must be set to 2'b00. The DTFT value is the number of 125 MHz clocks * 256 to count before considering a device timeout. The minimum value of DTFT is 2. For example, if the mac3_clk is 125 MHz clk (8 ns period), this is calculated as follows: (DTFT value) * 256 * (8 ns) Quick Reference: if DTFT = 0x2, 2*256*8 = 4usec timeout if DTFT = 0x5, 5*256*8 = 10usec timeout if DTFT = 0xA, 10*256*8 = 20usec timeout if DTFT = 0x10, 16*256*8 = 32usec timeout if DTFT = 0x19, 25*256*8 = 51usec timeout if DTFT = 0x31, 49*256*8 = 100usec timeout if DTFT = 0x62, 98*256*8 = 200usec timeout

usb2otg_GBUSERRADDRLO

Address: Operational Base + offset (0xC130)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	buserraddr Bus Address - Low. This register contains the lower 32 bits of the first bus address that encountered a SoC bus error. It is valid when the GSTS.BusErrAddrVld field is 1. It can only be cleared by resetting the core. Note: Only supported in AHB and AXI configurations.

usb2otg_GBUSERRADDRHI

Address: Operational Base + offset (0xC134)

Bit	Attr	Reset Value	Description
31:0	RU	0x00000000	buserraddr Bus Address - High. This register contains the higher 32 bits of the first bus address that encountered a SoC bus error. It is valid when the GSTS.BusErrAddrVld field is 1. It can only be cleared by resetting the core. Note: Only supported in AHB and AXI configurations.

usb2otg_GHWPARAMS0

Address: Operational Base + offset (0xC140)

Bit	Attr	Reset Value	Description
31:0	RO	0x2020400a	ghwparams0 Global Hardware Parameters Register 0.

usb2otg_GHWPARAMS1

Address: Operational Base + offset (0xC144)

Bit	Attr	Reset Value	Description
31:0	RO	0x0120c93b	ghwparams1 Global Hardware Parameters Register 1.

usb2otg_GHWPARAMS2

Address: Operational Base + offset (0xC148)

Bit	Attr	Reset Value	Description
31:0	RO	0x20190520	ghwparams2 Global Hardware Parameters Register 2.

usb2otg_GHWPARAMS3

Address: Operational Base + offset (0xC14C)

Bit	Attr	Reset Value	Description
31:0	RO	0x069cd084	ghwparams3 Global Hardware Parameters Register 3.

usb2otg_GHWPARAMS4

Address: Operational Base + offset (0xC150)

Bit	Attr	Reset Value	Description
31:0	RO	0x47822010	ghwparams4 Global Hardware Parameters Register 4.

usb2otg_GHWPARAMS5

Address: Operational Base + offset (0xC154)

Bit	Attr	Reset Value	Description
31:0	RO	0x04204108	ghwparams5 Global Hardware Parameters Register 5.

usb2otg_GHWPARAMS6

Address: Operational Base + offset (0xC158)

Bit	Attr	Reset Value	Description
31:0	RO	0x09d78020	ghwparams6 Global Hardware Parameters Register 6.

usb2otg_GHWPARAMS7

Address: Operational Base + offset (0xC15C)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	ghwparams7 Global Hardware Parameters Register 7.

usb2otg_GDBGFIFOSPACE

Address: Operational Base + offset (0xC160)

Bit	Attr	Reset Value	Description
31:16	RO	0x000a	space_available Space Available.
15:9	RO	0x00	reserved
8:0	RW	0x000	fifo_queue_select FIFO/Queue Select (or) Port-Select. FIFO/Queue Select[8:5] indicates the FIFO/Queue Type. FIFO/Queue Select[4:0] indicates the FIFO/Queue Number. Port-Select[3:0] selects the port-number when accessing GDBGLTSSM register.

usb2otg_GDBGLNMCC

Address: Operational Base + offset (0xC168)

Bit	Attr	Reset Value	Description
31:9	RO	0x000000	reserved

Bit	Attr	Reset Value	Description
8:0	RO	0x000	lnmcc_berc This field indicates the bit error rate information for the port selected in the GDBGFIFOSPACE.PortSelect field. This field is for debug purposes only.

usb2otg_GDBGBMU

Address: Operational Base + offset (0xC16C)

Bit	Attr	Reset Value	Description
31:8	RW	0x203000	bmu_bcu BMU_BCU Debug information.
7:4	RO	0x0	bmu_dcu BMU_DCU Debug information.
3:0	RO	0x0	bmu_ccu BMU_CCU Debug information.

usb2otg_GDBGLSPMUX

Address: Operational Base + offset (0xC170)

Bit	Attr	Reset Value	Description
31:24	RO	0x00	reserved
23:16	RW	0x3f	logic_analyzer_trace Logic Analyzer Trace Port MUX Select. Currently only bits[21:16] are used. A value of 6'h3F drives "0"s on the logic_analyzer_trace signal. If you plan to OR (instead using a mux) this signal with other trace signals in your system to generate a common trace signal, you can use this feature.
15	RW	0x0	endbc Enable debugging of Debug capability LSP in Host mode. Use HostSelect to select DbC LSP debug information presented in the GDBGLSP register.
14	RO	0x0	reserved
13:8	RW	0x00	hostselect Host LSP Select. Selects the LSP debug information presented in the GDBGLSP register in host mode.
7:4	RW	0x0	devselect Device LSP Select. Selects the LSP debug information presented in the GDBGLSP register in device mode. Or bit[7:4] of HOSTSELECT, Selects the LSP debug information presented in the GDBGLSP register in host mode.
3:0	RW	0x0	epselect Device Endpoint Select. Selects the Endpoint debug information presented in the GDBGEPINFO registers in device mode. Or bit[3:0] of HOSTSELECT, Selects the LSP debug information presented in the GDBGLSP register in host mode.

usb2otg_GDBGLSP

Address: Operational Base + offset (0xC174)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	lspdebug LSP Debug Information.

usb2otg_GDBGEPINFO0

Address: Operational Base + offset (0xC178)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000801	epdebug Endpoint Debug Information Low 32-bit.

usb2otg_GDBGEPINFO1

Address: Operational Base + offset (0xC17C)

Bit	Attr	Reset Value	Description
31:0	RO	0x00800000	epdebug Endpoint Debug Information High 32-bit.

usb2otg_GPRTBIMAP_HSLO

Address: Operational Base + offset (0xC180)

Bit	Attr	Reset Value	Description
31:4	RO	0x0000000	reserved
3:0	RW	0x0	binum1 HS USB Instance Number for Port 1. Application-programmable ID field.

usb2otg_GPRTBIMAP_FSLO

Address: Operational Base + offset (0xC188)

Bit	Attr	Reset Value	Description
31:4	RO	0x0000000	reserved
3:0	RW	0x0	binum1 FS USB Instance Number for Port 1. Application-programmable ID field.

usb2otg_GUSB2PHYCFG0

Address: Operational Base + offset (0xC200)

Bit	Attr	Reset Value	Description
31	RW	0x0	physoftrst UTMI PHY Soft Reset. Causes the usb2phy_reset signal to be asserted to reset a UTMI PHY. Not applicable to ULPI because ULPI PHYs are reset via their FunctionControl. Reset register, and the core automatically writes to this register when the core is reset (vcc_reset_n, USBCMD.HCRST, DCTL.SoftReset, or GCTL.SoftReset).
30	RW	0x0	u2_freeclk_exists Specifies whether your USB 2.0 PHY provides a free-running PHY clock, which is active when the clock control input is active. If your USB 2.0 PHY provides a free-running PHY clock, it must be connected to the utmi_clk[0] input. The remaining utmi_clk[n] must be connected to the respective port clocks. The core uses the Port-0 clock for generating the internal mac2 clock. 1'b0: USB 2.0 free clock does not exist 1'b1: USB 2.0 free clock exists Note: When the core is configured as device-only, do not set this bit to 1.
29:25	RO	0x00	reserved

Bit	Attr	Reset Value	Description
24:22	RW	0x0	lstrd LS Turnaround Time. This field indicates the value of the Rx-to-Tx packet gap for LS devices. The encoding is as follows: 3'h0: 2 bit times 3'h1: 2.5 bit times 3'h2: 3 bit times 3'h3: 3.5 bit times 3'h4: 4 bit times 3'h5: 4.5 bit times 3'h6: 5 bit times 3'h7: 5.5 bit times Note: This field is applicable only in Host mode. For normal operation (to work with most LS devices), set the default value of this field to 3'h0 (2 bit times). The programmable LS device inter-packet gap and turnaround delays are provided to support some legacy LS devices that might require different delays than the default/fixed ones. For instance, the Open LS mouse requires 3 bit times of inter-packet gap to work correctly.
21:19	RW	0x2	lsipd LS Inter-Packet Time. This field indicates the value of Tx-to-Tx packet gap for LS devices. The encoding is as follows: 3'h0: 2 bit times 3'h1: 2.5 bit times 3'h2: 3 bit times 3'h3: 3.5 bit times 3'h4: 4 bit times 3'h5: 4.5 bit times 3'h6: 5 bit times 3'h7: 5.5 bit times Note: This field is applicable only in Host mode. For normal operation (to work with most LS devices), set the default value of this field to 3'h2 (3 bit times). The programmable LS device inter-packet gap and turnaround delays are provided to support some legacy LS devices that might require different delays than the default/fixed ones. For instance, the AOpen LS mouse requires 3 bit times of inter-packet gap to work correctly.
18:14	RO	0x00	reserved

Bit	Attr	Reset Value	Description
13:10	RW	0x5	usbtrdim USB 2.0 Turnaround Time. Sets the turnaround time in PHY clocks. Specifies the response time for a MAC request to the Packet FIFO Controller (PFC) to fetch data from the DFIFO (SPRAM). The following are the required values for the minimum SoC bus frequency of 60 MHz. USB turnaround time is a critical certification criteria when using long cables and five hub levels. The required values for this field: 4'h5: When the MAC interface is 16-bit UTMI+. 4'h9: When the MAC interface is 8-bit UTMI+/ULPI. If SoC bus clock is less than 60 MHz, and USB turnaround time is not critical, this field can be set to a larger value. Note: This field is valid only in device mode.
9	RW	0x0	xcvrdly Transceiver Delay. Enables a delay between the assertion of the UTMI/ULPI Transceiver Select signal (for HS) and the assertion of the TxValid signal during a HS Chirp. When this bit is set to 1, a delay (of approximately 2.5 us) is introduced from the time when the Transceiver Select is set to 2'b00 (HS) to the time the TxValid is driven to 0 for sending the chirp-K. This delay is required for some UTMI/ULPI PHYs. Note: If you enable the hibernation feature when the device core comes out of power-off, you must re-initialize this bit with the appropriate value because the core does not save and restore this bit value during hibernation. This bit is valid only in device mode.
8	RW	0x0	enblslpm Enable utmi_sleep_n and utmi_l1_suspend_n. The application uses this bit to control utmi_sleep_n and utmi_l1_suspend_n assertion to the PHY in the L1 state. 1'b0: utmi_sleep_n and utmi_l1_suspend_n assertion from the core is not transferred to the external PHY. 1'b1: utmi_sleep_n and utmi_l1_suspend_n assertion from the core is transferred to the external PHY. Note: This bit must be set high for Port0 if SNPS PHY is used. In Device mode - Before issuing any device endpoint command when operating in 2.0 speeds, disable this bit and enable it after the command completes. Without disabling this bit, if a command is issued when the device is in L1 state and if mac2_clk (utmi_clk/ulpi_clk) is gated off, the command will not get completed.
7	RO	0x0	physel USB 2.0 High-Speed PHY or USB 1.1 Full-Speed. 1'b0: USB 2.0 high-speed UTMI+ or ULPI PHY. 1'b1: USB 1.1 full-speed serial transceiver.

Bit	Attr	Reset Value	Description
6	RW	0x0	suspendusb20 Suspend USB2.0 HS/FS/LS PHY. When set, USB2.0 PHY enters Suspend mode if Suspend conditions are valid. For DRD/OTG configurations, it is recommended that this bit is set to 0 during coreConsultant configuration. If it is set to 1, then the application must clear this bit after power-on reset. Application needs to set it to 1 after the core initialization completes. For all other configurations, this bit can be set to 1 during core configuration. Note: In host mode, on reset, this bit is set to 1. Software can override this bit after reset. In device mode, before issuing any device endpoint command when operating in 2.0 speeds, disable this bit and enable it after the command completes. If you issue a command without disabling this bit when the device is in L2 state and if mac2_clk (utmi_clk/ulpi_clk) is gated off, the command will not get completed.
5	RO	0x0	reserved
4	RO	0x0	ulpi_utmi_sel ULPI or UTMI+ Select. 1'b0: UTMI+ Interface 1'b1: ULPI Interface
3	RW	0x1	phyif If UTMI+ is selected, the application uses this bit to configure the core to support a UTMI+ PHY with an 8- or 16-bit interface. 1'b0: 8 bits 1'b1: 16 bits
2:0	RW	0x0	tout_cal HS/FS Timeout Calibration. The number of PHY clocks, as indicated by the application in this field, is multiplied by a bit-time factor; this factor is added to the high-speed/full-speed interpacket timeout duration in the core to account for additional delays introduced by the PHY. This may be required, since the delay introduced by the PHY in generating the linestate condition may vary among PHYs. The USB standard timeout value for high-speed operation is 736 to 816 (inclusive) bit times. The USB standard timeout value for full-speed operation is 16 to 18 (inclusive) bit times. The application must program this field based on the speed of connection. The number of bit times added per PHY clock are: High-speed operation: One 30-MHz PHY clock = 16 bit times One 60-MHz PHY clock = 8 bit times Full-speed operation: One 30-MHz PHY clock = 0.4 bit times One 60-MHz PHY clock = 0.2 bit times

usb2otg_GTXFIFOSIZn

Address: Operational Base + offset (0xC300)

Bit	Attr	Reset Value	Description
31:16	RW	0x04B9	txfstaddr_n Transmit FIFOn RAM Start Address. This field contains the memory start address for TxFIFOn in 64-bit words.
15:0	RW	0x000A	txfdep_n TxFIFO Depth. This field contains the depth of TxFIFOn in 64-bit words. Minimum value: 32; Maximum value: 32,768.

usb2otg_GRXFIFOSIZn

Address: Operational Base + offset (0xC380)

Bit	Attr	Reset Value	Description
31:16	RW	0x0334	rxfstaddr_n RxFIFOn RAM Start Address. This field contains the memory start address for RxFIFOn in 64-bit words.
15:0	RW	0x0185	rxfdep_n RxFIFOn RAM Start Address. This field contains the memory start address for RxFIFOn in 64-bit words.

usb2otg_GEVNTADRLO0

Address: Operational Base + offset (0xC400)

Bit	Attr	Reset Value	Description
31:0	RW	0xFF7001B0	evntadrlo Event Buffer Address. Holds the lower 32 bits of start address of the external memory for the Event Buffer. During operation, hardware does not update this address.

usb2otg_GEVNTADRHI0

Address: Operational Base + offset (0xC404)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	evntadrhi Event Buffer Address. Holds the higher 32 bits of start address of the external memory for the Event Buffer. During operation, hardware does not update this address.

usb2otg_GEVNTSIZ0

Address: Operational Base + offset (0xC408)

Bit	Attr	Reset Value	Description
31	RW	0x1	evntintrptmask Event Interrupt Mask. When set to '1', this prevents the interrupt from being generated. However, even when the mask is set, the events are queued.
30:16	RO	0x0000	reserved
15:0	RW	0x0100	eventsiz Event Buffer Size in bytes. Holds the size of the Event Buffer in bytes; must be a multiple of four. This is programmed by software once during initialization. The minimum size of the event buffer is 32 bytes.

usb2otg_GEVNTCOUNT0

Address: Operational Base + offset (0xC40C)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15:0	RW	0x0004	evntcount Event Count. When read, returns the number of valid events in the Event Buffer (in bytes). When written, hardware decrements the count by the value written. The interrupt line remains high when count is not 0.

usb2otg_GHWPARAMS8

Address: Operational Base + offset (0xC600)

Bit	Attr	Reset Value	Description
31:0	RO	0x0000047c	ghwparams8_32_0 ghwparams8

usb2otg_GTXFIFOPRIDEV

Address: Operational Base + offset (0xC610)

Bit	Attr	Reset Value	Description
31:7	RO	0x0000000	reserved
6:0	RW	0x00	gtxfifopridev Device TxFIFO priority. This register specifies the relative DMA priority level among the Device TXFIFOs (one per IN endpoint). Each register bit[n] controls the priority (1: high, 0: low) of each TXFIFO[n]. When multiple TXFIFOs compete for DMA service at a given time (that is, multiple TXQs contain TX DMA requests and their corresponding TXFIFOs have space available), the TX DMA arbiter grants access on a packet-basis in the following manner: 1. High-priority TXFIFOs are granted access using round-robin arbitration 2. Low-priority TXFIFOs are granted access using round-robin arbitration only after the high-priority TXFIFOs have no further processing to do (that is, either the TXQs are empty or the corresponding TXFIFOs are full). For scatter-gather packets, the arbiter grants successive DMA requests to the same FIFO until the entire packet is completed. When configuring periodic IN endpoints, software must set register bit[n]=1, where n is the TXFIFO assignment. This ensures that the DMA for isochronous or interrupt IN endpoints are prioritized over bulk or control IN endpoints. This register is present only when the core is configured to operate in the device mode. The register size corresponds to the number of Device IN endpoints.

usb2otg_GTXFIFOPRIHST

Address: Operational Base + offset (0xC618)

Bit	Attr	Reset Value	Description
31:3	RO	0x00000000	reserved

Bit	Attr	Reset Value	Description
2:0	RW	0x0	gtxfifoprihst Host TxFIFO priority. This register specifies the relative DMA priority level among the Host TXFIFOs (one per USB bus instance) within the associated speed group (HS/FSLS). Each register bit[n] controls the priority (1: high, 0: low) of TXFIFO[n] within a speed group. When multiple TXFIFOs compete for DMA service at a given time (i.e., multiple TXQs contain TX DMA requests and their corresponding TXFIFOs have space available), the TX DMA arbiter grants access on a packet-basis in the following manner: 1. Among the FIFOs in the same speed group (HS/FSLS): a. High-priority TXFIFOs are granted access using round-robin arbitration b. Low-priority TXFIFOs are granted access using round-robin arbitration only after the high-priority TXFIFOs have no further processing to do (that is, either the TXQs are empty or the corresponding TXFIFOs are full). 2. The TX DMA arbiter prioritizes the HS/FSLS speed group according to the ratio programmed in the GDMAHLRATIO register. For scatter-gather packets, the arbiter grants successive DMA requests to the same FIFO until the entire packet is completed. This register is present only when the core is configured to operate in the host mode (includes DRD and OTG modes). The register size corresponds to the number of configured USB bus instances; for example, in the default configuration, there are 2 USB bus instances (1 HS, and 1 FSLS).

usb2otg_GRXFIFOPRIHST

Address: Operational Base + offset (0xC61C)

Bit	Attr	Reset Value	Description
31:3	RO	0x00000000	reserved

Bit	Attr	Reset Value	Description
2:0	RW	0x0	grxfifoprihst Host RxFIFO priority This register specifies the relative DMA priority level among the Host RXFIFOs (one per USB bus instance) within the associated speed group (HS/FSLS). Each register bit[n] controls the priority (1: high, 0: low) of RXFIFO[n] within a speed group. When multiple RXFIFOs compete for DMA service at a given time (i.e., multiple RXQs contain RX DMA requests and their corresponding RXFIFOs have data available), the RX DMA arbiter grants access on a packet-basis in the following manner: 1. Among the FIFOs in the same speed group (HS/FSLS): a. High-priority RXFIFOs are granted access using round-robin arbitration b. Low-priority RXFIFOs are granted access using round-robin arbitration only after high-priority RXFIFOs have no further processing to do (that is, either the RXQs are empty or the corresponding RXFIFOs do not have the required data). 2. The RX DMA arbiter prioritizes the HS/FSLS speed group according to the ratio programmed in the GDMAHLRATIO register. For scatter-gather packets, the arbiter grants successive DMA requests to the same FIFO until the entire packet is completed. This register is present only when the core is configured to operate in the host mode (includes DRD and OTG modes). The register size corresponds to the number of configured USB bus instances; for example, in the default configuration, there are 2 USB bus instances (1 HS, and 1 FSLS).

usb2otg_GFIFOPRIDBC

Address: Operational Base + offset (0xC620)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved
1:0	RW	0x0	gfifopridbc Host DbC DMA priority. This register specifies the relative priority of the RXFIFOs and TXFIFOs associated with the DbC mode. It overrides the priority assigned in the corresponding indexes of the Host RXFIFO and TXFIFO DMA priority registers, when the DbC mode is enabled.

usb2otg_GDMAHLRATIO

Address: Operational Base + offset (0xC624)

Bit	Attr	Reset Value	Description
31:22	RO	0x000	reserved
21:8	RW	0x0000	hstrxfifo Host RXFIFO DMA High-Low Priority.
7:5	RO	0x0	reserved
4:0	RW	0x00	hsttxfifo Host TXFIFO DMA High-Low Priority.

usb2otg_GFLADJ

Address: Operational Base + offset (0xC630)

Bit	Attr	Reset Value	Description
31	RW	0x0	gfladj_refclk_240mhzdecr_pls1 This field indicates that the decrement value that the controller applies for each ref_clk must be GFLADJ_REFCLK_240MHZ_DECR and GFLADJ_REFCLK_240MHZ_DECR +1 alternatively on each ref_clk. Set this bit to a 1 only if GFLADJ_REFCLK_LPM_SEL is set to 1 and the fractional component of 240/ref_frequency is greater than or equal to 0.5. Examples: If the ref_clk is 24 MHz then 1. GUCTL.REF_CLK_PERIOD = 41 2. GFLADJ.GFLADJ_REFCLK_240MHZ_DECR = (240/24) = 10 3. GFLADJ.GFLADJ_REFCLK_240MHZDECR_PLS1 = 0
30:24	RW	0x00	gfladj_refclk_240mhz_decr This field indicates the decrement value that the controller applies for each ref_clk in order to derive a frame timer in terms of a 240-MHz clock. This field must be programmed to a non-zero value only if GFLADJ_REFCLK_LPM_SEL is set to 1. The value is derived as follows: GFLADJ_REFCLK_240MHZ_DECR = 240/ref_clk_frequency Examples: If the ref_clk is 24 MHz then 1. GUCTL.REF_CLK_PERIOD = 41 2. GFLADJ.GFLADJ_REFCLK_240MHZ_DECR = 240/24 = 10
23	RW	0x0	gfladj_refclk_lpm_sel This bit enables the functionality of running SOF counters on the ref_clk. This bit must not be set to 1 if GCTL.SOFITPSYNC bit is set to 1. Similarly, if GFLADJ_REFCLK_LPM_SEL set to 1, GCTL.SOFITPSYNC must not be set to 1. Note that the ref_clk frequencies supported in this mode are 16/17/19.2/20/24/39.7/40 MHz. The utmi_clk[0] signal of the core must be connected to the FREECLK of the PHY. Note: If you set this bit to 1, the GUSB2PHYCFG.U2_FREECLK_EXISTS bit must be set to 0.
22	RO	0x0	reserved
21:8	RW	0x0000	gfladj_refclk_fladj This field indicates the frame length adjustment to be applied when SOF counter is running on the ref_clk. SOF interval when GLADJ.GFLADJ_REFCLK_LPM_SEL is set to 1. This field must be programmed to a non-zero value only if GFLADJ_REFCLK_LPM_SEL is set to 1 or GCTL.SOFITPSYNC is set to 1. The value is derived as follows: FLADJ_REF_CLK_FLADJ=((125000/ref_clk_period_integer)-(125000/ref_clk_period)) * ref_clk_period where: 1. The ref_clk_period_integer is the integer value of the ref_clk period got by truncating the decimal (fractional) value that is programmed in the GUCTL.REF_CLK_PERIOD field. 2. The ref_clk_period is the ref_clk period including the fractional value. Examples: If the ref_clk is 24 MHz then 1. GUCTL.REF_CLK_PERIOD = 41 2. GFLADJ.GLADJ_REFCLK_FLADJ = ((125000/41)-(125000/41.6666))*41.6666 = 2032 (ignoring the fractional value).

Bit	Attr	Reset Value	Description
7	RW	0x0	gfladj_30mhz_sdbnd_sel This field selects whether to use the input signal fladj_30mhz_reg or the GFLADJ.GFLADJ_30MHZ to adjust the frame length for the SOF. When this bit is set to: 1'b1: the controller uses the register field GFLADJ.GFLADJ_30MHZ value 1'b0: the controller uses the input signal fladj_30mhz_reg value.
6	RO	0x0	reserved
5:0	RW	0x00	gfadj_30mhz This field indicates the value that is used for frame length adjustment instead of considering from the sideband input signal fladj_30mhz_reg. This enables post-silicon frame length adjustment in case the input signal fladj_30mhz_reg is connected to a wrong value or is not valid. For details on how to set this value, refer to section 5.2.4, "Frame Length Adjustment Register (FLADJ)," of the xHCI Specification.

usb2otg_DCFG

Address: Operational Base + offset (0xC700)

Bit	Attr	Reset Value	Description
31:24	RO	0x00	reserved
23	RW	0x0	ignstrmpp This bit only affects stream-capable bulk endpoints. Not applied.
22	RW	0x1	lpmcap LPM Capable. Not applied.
21:17	RW	0x04	nump Number of Receive Buffers. Not applied.
16:12	RW	0x00	intrnum Interrupt number. Indicates interrupt/EventQ number on which non-endpoint-specific device-related interrupts (see DEVT) are generated.
11:10	RO	0x10	reserved
9:3	RW	0x25	devaddr Device Address. The application must perform the following: 1. Program this field after every SetAddress request. 2. Reset this field to zero after USB reset.
2:0	RW	0x0	devspd Device Speed. Indicates the speed at which the application requires the core to connect, or the maximum speed the application can support. However, the actual bus speed is determined only after the chirp sequence is completed, and is based on the speed of the USB host to which the core is connected. 3'b000: High-speed (USB 2.0 PHY clock is 30 MHz or 60 MHz) 3'b001: Full-speed (USB 2.0 PHY clock is 30 MHz or 60 MHz)

usb2otg_DCTL

Address: Operational Base + offset (0xC704)

Bit	Attr	Reset Value	Description
31	RW	0x1	run_stop The software writes 1 to this bit to start the device controller operation. To stop the device controller operation, the software must remove any active transfers and write 0 to this bit. When the controller is stopped, it sets the DSTS.DevCtrlHlt bit when the core is idle and the lower layer finishes the disconnect process. The Run/Stop bit must be used in following cases as specified: 1. After power-on reset and CSR initialization, the software must write 1 to this bit to start the device controller. The controller does not signal connect to the host until this bit is set. 2. The software uses this bit to control the device controller to perform a soft disconnect. When the software writes 0 to this bit, the host does not see that the device is connected. The device controller stays in the disconnected state until the software writes 1 to this bit. The minimum duration of keeping this bit cleared is specified in the Note below. If the software attempts a connect after the soft disconnect or detects a disconnect event, it must set DCTL[8:5] to 5 before reasserting the Run/Stop bit. 3. When the USB is in a lower power state and the Two Power Rails configuration is selected, software writes 0 to this bit to indicate that it is going to turn off the Core Power Rail. After the software turns on the Core Power Rail again and re-initializes the device controller, it must set this bit to start the device controller. Note: The following is the minimum duration under various conditions for which the soft disconnect (SftDiscon) bit must be set for the USB host to detect a device disconnect: 10ms: For high-speed, when the device state is Suspended, Idle, or not Idle/Suspended (performing transactions); For full-speed/low-speed, when the device state is Suspended, Idle, or not Idle/Suspended (performing transactions). To accommodate clock jitter, it is recommended that the application add extra delay to the specified minimum duration.

Bit	Attr	Reset Value	Description
30	R/WSC	0x0	csftrst Core Soft Reset. Reset all clock domains as follows: 1. This bit clears the interrupts and all the CSRs except GSTS, GSNPSID, GGPIO, GUID, GUSB2PHYCFGn registers, GUSB3PIPECTLn registers, DCFG, DCTL, DEVTEN, and DSTS registers. 2. All module state machines (except the SoC Bus Slave Unit) are reset to the IDLE state, and all the TxFIFOs and the RxFIFO are flushed. 3. Any transactions on the SoC bus Master are terminated as soon as possible, after gracefully completing the last data phase of a SoC bus transfer. Any transactions on the USB are terminated immediately. The application can write this bit at any time to reset the core. This is a self-clearing bit; the core clears this bit after all necessary logic is reset in the core, which may take several clocks depending on the core's current state. Once this bit is cleared, the software must wait at least 3 PHY clocks before accessing the PHY domain (synchronization delay). Typically, software reset is used during software development and also when you dynamically change the PHY selection bits in the USB configuration registers listed above. When you change the PHY, the corresponding clock for the PHY is selected and used in the PHY domain. Once a new clock is selected, the PHY domain must be reset for proper operation.
29	RO	0x0	reserved
28:24	RW	0x00	hirdthres HIRD Threshold. The core asserts output signals utmi_l1_suspend_n and utmi_sleep_n on the basis of this signal: The core asserts utmi_l1_suspend_n to put the PHY into Deep Low-Power mode in L1 when both of the following are true: 1. HIRD value is greater than or equal to the value in DCTL.HIRD_Thres[3:0] 2. HIRD_Thres[4] is set to 1'b1. The core asserts utmi_sleep_n on L1 when one of the following is true: 1. If the HIRD value is less than HIRD_Thres[3:0] or 2. HIRD_Thres[4] is set to 1'b0. Note: This field must be set to '0' during SuperSpeed mode of operation.

Bit	Attr	Reset Value	Description
23:20	RW	0x0	lpm_nyet_thres LPM NYET Threshold Handshake response to LPM token specified by device application. Response depends on DCFG.LPMCap. DCFG.LPMCap is 1'b0 - The core always responds with Timeout (that is, no response). DCFG.LPMCap is 1'b1 - The core responds with an ACK on successful LPM transaction, which requires that all of the following are satisfied: 1. There are no PID or CRC5 errors in both the EXT token and the LPM token (if not true, inactivity results in a timeout ERROR). 2. No data is pending in the Transmit FIFO and OUT endpoints not in flow controlled state (else NYET). 3. The BESL value in the LPM token is less than or equal to LPM_NYET_thres[3:0].
19	RW	0x0	keep_connect When 1, this bit enables the save and restore programming model by preventing the core from disconnecting from the host when DCTL.RunStop is set to 0. The device core disconnects from the host when DCTL.RunStop is set to 0. This bit indicates whether to preserve this behavior (0), or if the core must not disconnect when RunStop is set to 0 (1).
18	RW	0x0	l1_hibernation_en When this bit is set along with KeepConnect, the device core generates a Hibernation Request Event if L1 is enabled and the HIRD value in the LPM token is larger than the threshold programmed in DCTL.HIRD_Thres. The core does not exit the LPM L1 state until software writes Recovery into the DCTL.ULStChngReq field. This prevents corner cases where the device is entering hibernation at the same time the host is attempting to exit L1.
17	RW	0x0	crs Controller Restore State. This command is similar to the USBCMD.CRS bit in host mode and initiates the restore process. When software sets this bit to 1, the controller immediately sets DSTS.RSS to 1. When the controller has finished the restore process, it sets DSTS.RSS to 0. Note: When read, this field always returns 0.
16	RW	0x0	css Controller Save State. This command is similar to the USBCMD.CSS bit in host mode and initiates the save process. When software sets this bit to 1, the controller immediately sets DSTS.SSS to 1. When the controller has finished the save process, it sets DSTS.SSS to 0. Note: When read, this field always returns 0.
15:13	RO	0x0	reserved
12	RW	0x0	initu2ena Not applied.
11	RW	0x1	acceptu2ena Not applied.
10	RW	0x0	initu1ena Not applied.
9	RW	0x1	acceptu1ena Not applied.

Bit	Attr	Reset Value	Description
8:5	RW	0x0	ulstchngreq Software writes this field to issue a USB state change request. A change in this field indicates a new request to the core. If software wants to issue the same request back-to-back, it must write a 0 to this field between the two requests. The result of the state change request is reflected in the USB State in DSTS. These bits are self-cleared on the MAC Layer exiting suspended state. If software is updating other fields of the DCTL register and not intending to force any link state change, then it must write a 0 to this field. In HS/FS/LS mode: Value: Requested USB state transition 8: Remote wakeup request Others: Reserved The Remote wakeup request must be issued 2us after the device goes into suspend state (DSTS[21:18] is 3). Note: After coming out of hibernation, software must write 8 (Recovery) into this field to confirm exit from the suspended state.
4:1	RW	0x0	tstctl Test Control. 4'b000: Test mode disabled 4'b001: Test_J mode 4'b010: Test_K mode 4'b011: Test_SE0_NAK mode 4'b100: Test_Packet mode 4'b101: Test_Force_Enable Others: Reserved
0	RO	0x0	reserved

usb2otg_DEVTEN

Address: Operational Base + offset (0xC708)

Bit	Attr	Reset Value	Description
31:13	RO	0x00000	reserved
12	RW	0x1	vendevtstrcvden Vendor Device Test LMP Received Event. 1'b0: Disable this event 1'b1: Enable this event
11:10	RO	0x0	reserved
9	RW	0x1	errticerrevten Erratic Error Event Enable. 1'b0: Disable this event 1'b1: Enable this event
8	RO	0x0	reserved
7	RW	0x0	softevten Start of (u)frame Event Enable. 1'b0: Disable this event 1'b1: Enable this event
6	RW	0x0	u3l2l1_susp_en U3/L2-L1 Suspend Event Enable. 1'b0: Disable this event 1'b1: Enable this event

Bit	Attr	Reset Value	Description
5	RW	0x0	hibernation_req_evt_en Hibernation Request Event Enable. 1'b0: Disable this event 1'b1: Enable this event
4	RW	0x1	wkupevten Resume/Remote Wakeup Detected Event Enable. 1'b0: Disable this event 1'b1: Enable this event
3	RW	0x1	ulstcngen USB State Change Event Enable. 1'b0: Disable this event 1'b1: Enable this event
2	RW	0x1	connectdoneevten Connection Done Event Enable. 1'b0: Disable this event 1'b1: Enable this event
1	RW	0x1	usbrstevten USB Reset Event Enable. 1'b0: Disable this event 1'b1: Enable this event
0	RW	0x1	disconnevten Disconnect Detected Event Enable. 1'b0: Disable this event 1'b1: Enable this event

usb2otg_DSTS

Address: Operational Base + offset (0xC70C)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29	RW	0x0	dcnrd Device Controller Not Ready. The bit indicates that the core is in the process of completing the state transitions after exiting from hibernation. To complete the state transitions, it takes 256 bus clock cycles from the time DCTL[31].Run/Stop is set. During hibernation, if the UTMI/ULPI PHY is in suspended state, then the 256-bus clock cycle delay starts after the PHY exited suspended state. Software must set DCTL[31].Run/Stop to 1 and wait for this bit to be de-asserted to zero before processing DSTS.USBLnkSt.
28:26	RO	0x0	reserved
25	RW	0x0	rss Restore State Status. This bit is similar to the USBSTS.RSS in host mode. When the controller finishes the restore process, it completes the command by setting DSTS.RSS to 0.
24	RO	0x0	sss Save State Status. This bit is similar to the USBSTS.SSS in host mode. When the controller has finished the save process, it completes the command by setting DSTS.SSS to 0.

Bit	Attr	Reset Value	Description
23	RO	0x0	coreidle Core Idle. The bit indicates that the core finished transferring all RxFIFO data to system memory, writing out all completed descriptors, and all Event Counts are zero. Note: While testing for Reset values, mask out the read value. This bit represents the changing state of the core and does not hold a static value.
22	RO	0x0	devctrlhlt Device Controller Halted. This bit is set to 0 when the Run/Stop bit in the DCTL register is set to 1. The core sets this bit to 1 when, after SW sets Run/Stop to 0, the core is idle and the lower layer finishes the disconnect process. When Halted=1, the core does not generate Device events. Note: The core does not set this bit to 1 if GEVNTCOUNTn has some valid value. Software needs to acknowledge the events that are generated (by writing to GEVNTCOUNTn) while it is waiting for this bit to be set to 1.
21:18	RO	0x0	usblnkst In HS/FS/LS mode: 4'h0: On state 4'h2: Sleep (L1) state 4'h3: Suspend (L2) state 4'h4: Disconnected state (Default state) 4'h5: Early Suspend state (valid only when Hibernation is disabled, GCTL[1].GblHibernationEn = 0) 4'he: Reset (valid only when Hibernation is enabled, GCTL[1].GblHibernationEn = 1) 4'hf: Resume (valid only when Hibernation is enabled, GCTL[1].GblHibernationEn = 1) The link state Resume/Reset indicates that the core received a resume or USB reset request from the host while the link was in hibernation. Software must write 8 (Recovery) to the DCTL.ULStChngReq field to acknowledge the resume/reset request. When Hibernation is enabled, GCTL[1].GblHibernationEn = 1, this field USBLnkSt is valid only when DCTL[31].Run/Stop set to 1 and DSTS[29].DCNRD = 0.
17	RO	0x1	rx_fifo_empty Rx Fifo Empty.
16:3	RO	0x3023	reserved
2:0	RU	0x0	connectspd Connected Speed. Indicates the speed at which the DWC_usb3 core has come up after speed detection through a chirp sequence. 3'b000: High-speed (PHY clock is running at 30 or 60 MHz) 3'b001: Full-speed (PHY clock is running at 30 or 60 MHz)

usb2otg_DGCMDPAR

Address: Operational Base + offset (0xC710)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	parameter This register indicates the device command parameter. This must be programmed before or along with the device command. The available device commands are listed in DGCMD register.

usb2otg_DGCMD

Address: Operational Base + offset (0xC714)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15:12	RO	0x0	cmdstatus Command Status: 1'b1: CmdErr: Indicates that the device controller encountered an error while processing the command. 1'b0: Indicates command success
11:10	RO	0x0	reserved
9	R/WSC	0x0	cmdact Command Active. The software sets this bit to 1 to enable the device controller to execute the generic command. The device controller sets this bit to 0 after executing the command.
8	RW	0x0	cmdioc Command Interrupt on Complete. When this bit is set, the device controller issues a Generic Command Completion event after executing the command. Note that this interrupt is mapped to DCFG.IntrNum. Note: This field must not set to 1 if the DCTL.RunStop field is 0.
7:0	RW	0x00	cmdtyp Command Type. Specifies the type of command the software driver is requesting the core to perform. 8'h0: Reserved 8'h1: Set Endpoint Configuration - 64 or 96-bit Parameter 8'h2: Set Endpoint Transfer Resource Configuration - 32-bit Parameter 8'h3: Get Endpoint State - No Parameter Needed 8'h4: Clear Stall (see Set Stall) - No Parameter Needed 8'h5: Start Transfer - 64-bit Parameter 8'h6: Update Transfer - No Parameter Needed 8'h7: End Transfer - No Parameter Needed 8'h8 Start New Configuration - No Parameter Needed

usb2otg_DALEPENA

Address: Operational Base + offset (0xC720)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000003	usbactep USB Active Endpoints. This field indicates if a USB endpoint is active in the current configuration and interface. It applies to USB IN endpoints 0~15 and OUT endpoints 0~15, with one bit for each of the 32 possible endpoints. Even numbers are for USB OUT endpoints, and odd numbers are for USB IN endpoints, as follows: Bit[0]: USB EP0-OUT Bit[1]: USB EP0-IN Bit[2]: USB EP1-OUT Bit[3]: USB EP1-IN ... The entity programming this register must set bits 0 and 1 because they enable control endpoints that map to physical endpoints (resources) after USBReset. Hardware clears these bits for all endpoints (other than EP0-OUT and EP0-IN) after detecting a USB reset event. After receiving SetConfiguration and SetInterface requests, the application must program endpoint registers accordingly and set these bits.

usb2otg_DEPnCMDPAR2

Address: Operational Base + offset (0xC800)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	parameter This register indicates the physical endpoint command Parameter 2. It must be programmed before issuing the command.

usb2otg_DEPnCMDPAR1

Address: Operational Base + offset (0xC804)

Bit	Attr	Reset Value	Description
31:0	RW	0xFF7002E0	parameter This register indicates the physical endpoint command Parameter 1. It must be programmed before issuing the command.

usb2otg_DEPnCMDPAR0

Address: Operational Base + offset (0xC808)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	parameter This register indicates the physical endpoint command Parameter 0. It must be programmed before issuing the command.

usb2otg_DEPnCMD

Address: Operational Base + offset (0xC80C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	commandparam Command Parameters or Event Parameters when this register is written: For Start Transfer command: The 16-bit StreamID assigned to this transfer For Start Transfer command applied to an isochronous endpoint: StartMicroFramNum, Indicates the (micro) frame number to which the first TRB applies. For Update Transfer, End Transfer, and Start New Configuration commands: [22:16]: Transfer Resource Index (XferRscIdx). The hardware-assigned transfer resource index for the transfer, which was returned in response to the Start Transfer command. The application software-assigned transfer resource index for a Start New Configuration command. Event Parameters (EventParam), when this register is read.
15:12	RW	0x0	cmdstatus Command Completion Status. The information is in the same format as bits 15:12 of the Endpoint Command Complete event.
11	RW	0x0	hipri_forcerm HighPriority: Only valid for Start Transfer command. ForceRM: Only valid for End Transfer command. ClearPendIN: Only valid for Clear Stall command. Software sets this bit to clear any pending IN transaction (on that endpoint) stuck at the lower layers when a Clear Stall command is issued.
10	RW	0x0	cmdact Command Active. Software sets this bit to 1 to enable the device endpoint controller to execute the generic command. The device controller sets this bit to 0 when the CmdStatus field is valid and the endpoint is ready to accept another command. This does not imply that all the effects of the previously-issued command have taken place.
9	RO	0x0	reserved
8	RW	0x0	cmdioc Command Interrupt on Complete. When this bit is set, the device controller issues a generic Endpoint Command Complete event after executing the command. Note that this interrupt is mapped to DEPCFG.IntrNum. When the DEPCFG command is executed, the command interrupt on completion goes to the interrupt pointed by the DEPCFG.IntrNum in the current command. Note: This field must not set to 1 if the DCTL.RunStop field is 0.
7:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3:0	RW	0x6	cmdtyp Command Type. Specifies the type of command the software driver is requesting the core to perform. 4'h0: Reserved 4'h1: Set Endpoint Configuration - -64 or 96-bit Parameter 4'h2: Set Endpoint Transfer Resource Configuration - 32-bitparameter 4'h3: Get Endpoint State - No Parameter Needed 4'h4: Set Stall - No Parameter Needed 4'h5: Clear Stall (see Set Stall) - No Parameter Needed 4'h6: Start Transfer - 64-bit Parameter 4'h7: Update Transfer - No Parameter Needed 4'h8: End Transfer - No Parameter Needed 4'h9: Start New Configuration - No Parameter Needed

28.5 Interface Description

Table 28-2 USB2.0 PHY Interface Description

Module Pin	Direction	Pin Name	Descriptions
ID	I/O	OTG_ID	ID Pin
VBUS18	I/O	OTG_VBUS1V8	VBUS18. It is divided from VBUS on USB connector
VCCA18	I/O	USB_AVDD_1V8	1.8 V power supply
DP	I/O	OTG_DP	USB differential signal IO, positive end
DM	I/O	OTG_DM	USB differential signal IO, negative end
VBUS18	I/O	OTG_EXTR	USB2.0 PHY Host Port VBUS
AGND	I/O	VSS	Ground IO for both 3.3V,1.8V and 0.8V
VCCA33	I/O	USB_AVDD_3V3	3.3 V power supply(only Full speed driver)
VDDA	I/O	USB_AVDD_0V8	0.8 V power supply

28.6 Application Notes

28.6.1 Some Settings and Description About USB PHY

Here list some usbotg phy configure register of TOP_GRF in detail.

- GRF_USBPHY_CON0

Table 28-3 GRF_USBPHY_CON0 Description

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

Bit	Attr	Reset Value	Description
15	RW	0x1	usbotg_data_bus16_8 Software configure for OTGPHY data_bus16_8 Selects between 8 and 16-bit data transfer. 1'b1: 16-bit data path operation enabled, CLK60_30 = 30 MHz. 1'b0: 8-bit data path operation enabled, CLK60_30 = 60 MHz. This signal is sampled by USB PHY IP macro cell only when RESET becomes low. Default is 16-bit model and not suggest to configure this bit.
14	RW	0x0	usbotg_pll_en Software configure for OTGPHY pll_en PLL enable signal, internal PLL start to work when this signal is set to high, otherwise PLL is disabled and PHY IP stop working. This bit must be set to high before usb function.
13	RW	0x0	usbotg_dischrgvbus Software configure for OTGPHY dischrgvbus Discharge VBUS18. The signal enables discharging VBUS18. 1'b0: do not discharge VBUS18 through a resistor 1'b1: discharge VBUS18 through a resistor (this has to be active for at least 50ms)
12	RW	0x0	usbotg_chrgvbus Software configure for OTGPHY chrgvbus Charge VBUS18. The signal enables charging VBUS18. 1'b0: do not charge VBUS18 through a resistor 1'b1: charge VBUS18 through a resistor (this has to be active for at least 30ms)
11	RW	0x1	usbotg_idpulup Software configure for OTGPHY idpulup Enable ID pin sample. Signal that enables the sampling of the analog ID line. 1'b0: Sampling of ID pin is disabled (IDDIG force to low). 1'b1: Sampling of ID pin is enabled.
10	RW	0x1	usbotg_utmi_iddig Software configure for OTGPHY utmi_iddig Plug Indicator. Indicates whether the connected plug is a mini-A or mini-B. This is only valid when ID_PULLUP is set to 1b. It must be valid within 20ms after ID_PULLUP is set to 1b. 1'b0: connected plug is a mini-A 1'b1: connected plug is a mini-B In default iddig is controlled by hardware.
9	RW	0x0	usbotg_utmi_iddig_sel IDDIG selection. 1'b0: USBPHY output 1'b1: Software output, USBPHY_CON0[10]

Bit	Attr	Reset Value	Description
8	RW	0x0	usbotg_dmpulldown Software configure for OTGPHY dmpulldown DM Pulldown. This signal enables the 15k Ohm Pull-down resistor on the DM line. 1'b0: Pull-down resistor not connected to DM 1'b1: Pull-down resistor connected to DM In default this is controlled by hardware.
7	RW	0x0	usbotg_dppilldown Software configure for OTGPHY dppilldown DP Pulldown. This signal enables the 15k Ohm pull-down resistor on the DP line. 1'b0: Pull-down resistor not connected to DP 1'b1: Pull-down resistor connected to DP In default this is controlled by hardware.
6	RW	0x1	usbotg_termselect Software configure for OTGPHY termselect Termination Select. This signal selects between the FS and HS terminations: 1'b0: HS termination enabled 1'b1: FS termination enabled In default this is controlled by hardware.
5:4	RW	0x1	usbotg_xcvrselect Software configure for OTGPHY xcvrselect Transceiver Select. 2'b00: HS Transceiver enabled 2'b01: FS Transceiver enabled 2'b10: LS Transceiver enabled 2'b11: Reserved In default this is controlled by hardware.
3:2	RW	0x0	usbotg_opmode Software configure for OTGPHY opmode Operational Mode. These signals select between various operational modes: 2'b00: Normal operation 2'b01: Non-driving. Transmitter buffer is in high impedance and pull-up/down resistors are disconnected 2'b10: Disable bit stuffing and NRZI encoding 2'b11: Reserved In default this is controlled by hardware.

Bit	Attr	Reset Value	Description
1	RW	0x1	usbotg_suspend_n Software configure for OTGPHY suspend_n Active low, asynchronous. When enabled, drive IP into suspend mode and consume minimal current from power supply. 1'b0: IP in suspend mode, only those circuit that serve for resume function is still active 1'b1: IP in normal operation In default this is controlled by hardware.
0	RW	0x0	otgphy_input_sel OTGPHY input selection. 1'b0: Hardware input 1'b1: Software input

- GRF_USBPHY_CON1

Table 28-4 GRF_USBPHY_CON1 Description

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	usbotg_chg_rst Software configure for OTGPHY chg_rst Charge detector reset signal, active high, asynchronous. 1'b0: charge detector is normal operation 1'b1: charge detector is reset to initial
14	RW	0x0	usbotg_chg_en Software configure for OTGPHY chg_en Charge detector enable signal. 1'b0: charging detect is off, IP is configured as SDP in HOST application or is configured as standard device in DEVICE application 1'b1: charging detect is on, IP is configured as CDP in HOST application or is configured as portable device in DEVICE application
13	RW	0x1	usbhost_suspendm_byps Software configure for HOSTPHY suspendm_byps OTG_SUSPENDM linked control bypass signal. 1'b0: OTG_SUSPENDM is controlled by IDDIG(default) 1'b1: OTG_SUSPENDM is controlled by SOC Please refer to USB2HOST section.

Bit	Attr	Reset Value	Description
12	RW	0x0	usbotg_suspendm_byps Software configure for OTGPHY suspendm_byps OTG_SUSPENDM linked control bypass signal. 1'b0: OTG_SUSPENDM is controlled by IDDIG(default) 1'b1: OTG_SUSPENDM is controlled by SOC Please refer to the description in GRF_USBPHY_CON1. usbotg_suspendm in this section.
11	RW	0x0	usbotg_txbitstuff_enable Software configure for OTGPHY txbitstuff_enable TX BITSTUFF ENABLE. Indicates if the data on the DATA_OUT[15:0]lines need to be bit stuffed or not. 1'b0: Bit stuffing is disabled 1'b1: Bit stuffing is enabled This signal is only used when OP_MODE[1:0] is set to 11b.
10	RW	0x0	usbotg_suspendm Software configure for OTGPHY suspend This signal is used for VBUS negotiation in OTG application and HOST disconnect in HOST application, in DEVICE mode this signal should be tie to 0. In OTG/HOST application this signal should be connected to 1. In HOST application, IDDIG is not valid, OTG_SUSPENDM must be set to 1 and OTG_SUSPENDM_BYPS must set to 1. In DEVICE application, IDDIG is not valid, OTG_SUSPENDM must be set to 0 and OTG_SUSPENDM_BYPS must set to 1.
9	RW	0x0	usbotg_fs_se0 Software configure for OTGPHY fs_se0
8	RW	0x0	usbotg_fs_data Software configure for OTGPHY fs_data
7	RW	0x1	usbotg_fs_oe Software configure for OTGPHY fs_oe
6	RW	0x0	usbotg_fs_xver_own Software configure for OTGPHY fs_xver_own
5:3	RO	0x0	reserved
2	RW	0x0	usbotg_suspend_n_1 Software configure for OTGPHY suspend_n This is backups of USBPHY_CON0[1]

Bit	Attr	Reset Value	Description
1:0	RW	0x0	usbotg_suspend_n_sel usbotg_suspend_n selection. 2'b00: ~usbotg_utmi_suspend_com_n & ~usbotg_utmi_l1_suspend_com_n 2'b10: ~usbotg_utmi_suspend_com_n & ~usbotg_utmi_l1_suspend_com_n 2'b11: ~usbotg_uspend_n & ~usbotg_utmi_l1_suspend_n 2'b01: USBPHY_CON1[2] In default USBPHY_CON0[0]=0, then usbotg_suspend_n will control by this field.

28.6.2 OTG Reset Sequence

In USB function mode, the recommended reset sequence is in figure1-2. 'RST' is the reset signal of USB controller and 'CLK' is the main clock of USB controller. At first, 'RST' is set to 1 and USB controller starts to work; secondly, USB controller outputs 'SUSPENDM'. 'RESET' is the reset of the usb phy. Both 'RST' and 'RESET' are controlled by CRU model.
'SUSPENDM' is set to 1 firstly to enable PLL, PLL starts to work and frequency starts locking, 'RESET' is set to 0 to enable clock lock detector, and 'CLK60_30' is valid when frequency is stable. PLL lock time(Tlock) is about 500us including the time of bandgap building, PLL locking and so on, Tlock also includes the PVT range.
'CLK_480' is controlled by 'PLL_EN', and 'CLK_480' is not valid when 'PLL_EN' is 0. Because 'PLL_EN' is the reset signal of delay control cell, so there must be a rise edge of 'PLL_EN' to make delay control cell to work correctly. Tdelay is the time of 'PLL_EN' rise edge to 'CLK_480' transmitting 480MHz clock, the value is greater than 200us and less than 500us.

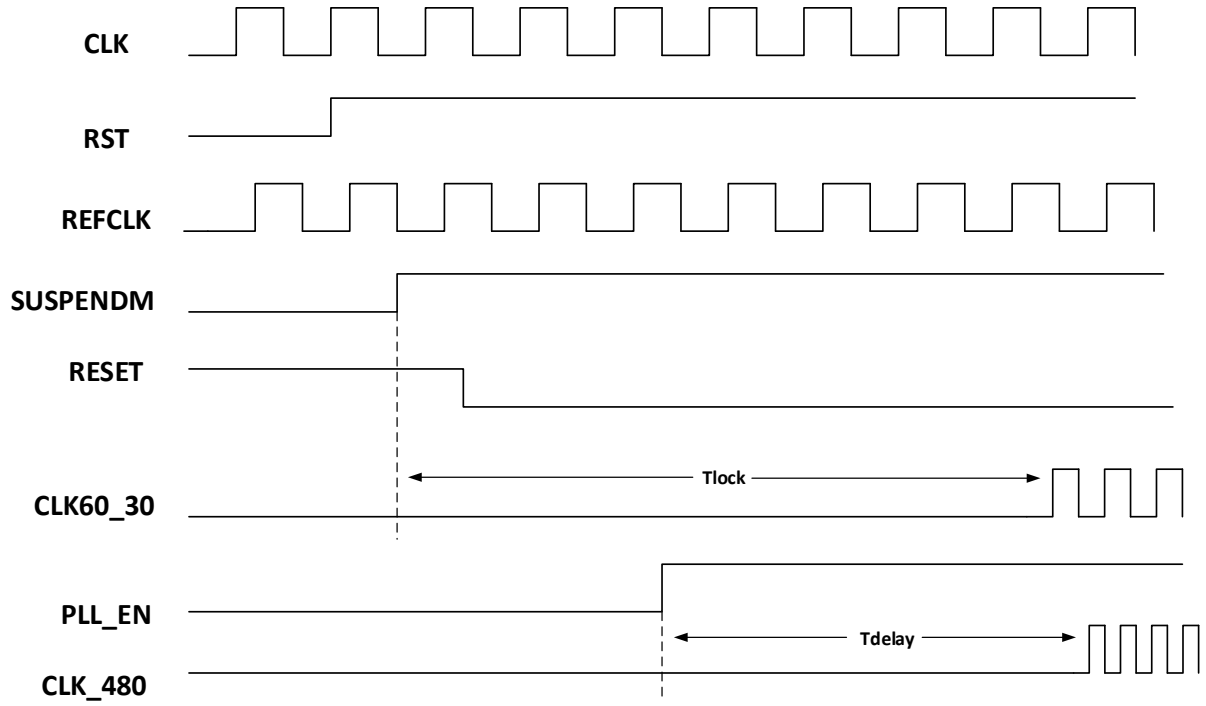


Fig. 28-2 USB2.0 OTG reset sequence

28.6.3 BchPage Application

There are three steps in charge detector sequence including data contact detection, primary detection and secondary detection. Data contact detection is judging whether DEVICE and HOST are connected or not, primary detection is judging whether the charger is connected

or not, secondary detection is judging whether the charger is DCP or CDP.
In DEVICE/HOST application, when 'CHG_EN' is set to 1, 'CHG_RST' is set to 0 to start charge detection. This two signal can be control by GRF_USBPHY_CON1. During the detection, 'RESET' must be set to 1, the PHY function is disable before 'PHY_CONNECT' changes to 1. Tdelay is greater than 1ms.

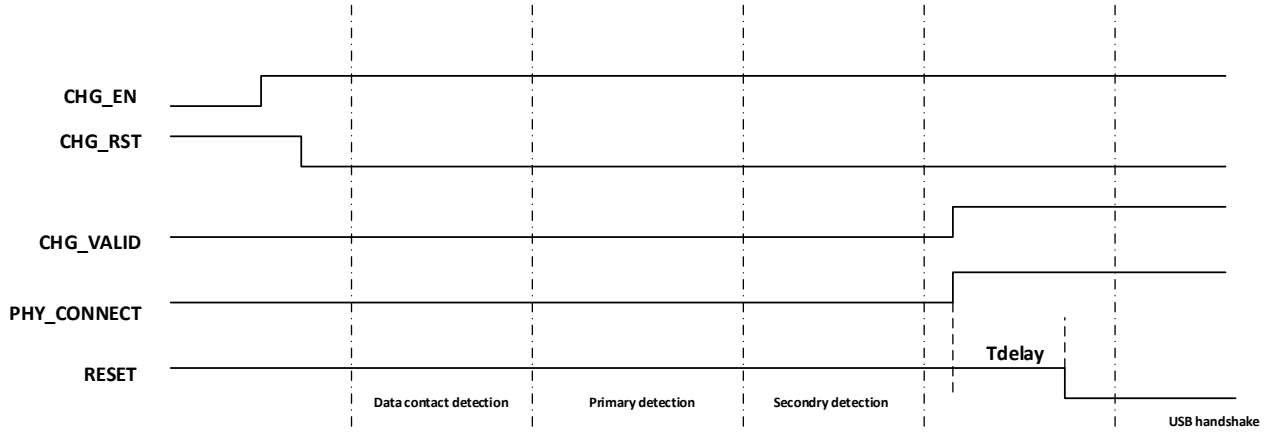


Fig. 28-3 USB2.0 OTG charge sequence

Table 1-5 and 1-6 shows the description about charge model.

Table 28-5 Charge Description in device model

Host Port	CHG_VALID	PHY_CONNECT	Description
DCP	1	0	Support charge, not connected
CDP	1	1	Support charge, connected
SDP	0	1	Not support charge, connected
ACA	Not supported	Not supported	Not supported

Table 错误!文档中没有指定样式的文字。-1 Charge Description in host model

Host Port	Device Port	CHG_VALID	PHY_CONNECT
DCP	support charge	1	0
DCP	not support charge	0	0
CDP	support charge	1	1
CDP	not support charge	0	1
SDP	-	0	1

In device application, DCP, CDP and SDP can be detected and ACA is not supported. In host application, host can be configured as DCP, CDP and SDP.

28.6.4 OTG Programming Model

When detect ID change event (see TOP GRF) or VBUS change event (see TOP GRF) after disconnect, it means a USB2 OTG A device or B device may connect, then check status of ID(see TOP GRF). If ID==0, it will work as A device, then follow host programming flow; if ID==1, it will work as B device, then it follow device programming flow.
Note: USB2.0 OTG doesn't support host/device mode swapping through HNP and RSP.
1.1.1

Chapter 29 USB2.0 Host

29.1 Overview

USB2.0 host controller supports fully USB2.0 functions with one EHCI host controller and one OHCI host controller, and each host controller has one USB port. OHCI host controller only supports full-speed and low-speed mode and is used for full-speed devices and low-speed devices. EHCI only supports high-speed mode and is used for high-speed devices. OHCI host controller and EHCI host controller shares the same USB port, EHCI host controller will auto select the owner (OHCI or EHCI) of this USB port depending on the speed mode of attached devices, when selecting OHCI as owner, OHCI host controller will serve for the attached device; when selecting EHCI as owner, EHCI host controller will serve for the attached device.

USB2.0 Host Controller supports the following features:

- Compatible Specifications
  - Universal Serial Bus Specification, Revision 2.0
  - Enhanced Host Controller Interface Specification (EHCI), Revision 1.0
  - Open Host Controller Interface Specification (OHCI), Revision 1.0a

Support High-speed (480Mbps), Full-speed (12Mbps) and Low-speed (1.5Mbps)

29.2 Block Diagram

USB2.0 Host Controller comprises with:

- EHCI Host Controller: Perform High-speed transactions
- OHCI Host Controller: Perform full/low-speed transactions
- Port Routing Control: Select EHCI Host Controller or OHCI Host Controller

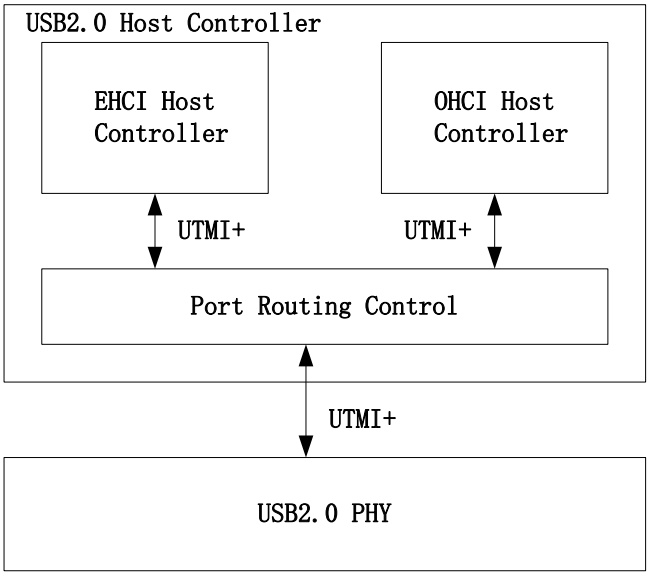


Fig. 29-1 USB2.0 Host Controller Block Diagram

29.3 Function Description

29.3.1 EHCI Host Controller

It performs descriptors and data read or write from or to system memory and packs or unpack USB transactions from or to UTMI+ interface defined in EHCI specification for high-speed data transmission.

29.3.2 OHCI Host Controller

It performs descriptors and data read/write from/to system memory and packs or un-pack USB transactions from or to UTMI+ interface defined in OHCI specification for full-speed or low-speed data transmission.

29.3.3 Port Routing Control

As part of logic in the EHCI host controller, it is used to auto-select EHCI or OHCI host controller to serve the attached device depending on the speed of the attached device.

29.4 Register Description

29.4.1 Internal Address Mapping

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

Table 29-1 USB2.0 Host Controller Address Mapping

Base Address[16]	Device	Address Length	Offset Address Range
1'b0	EHCI	64K BYTE	0x00000 ~ 0x0ffff
1'b1	OHCI	64K BYTE	0x10000 ~ 0x1ffff

EHCI and OHCI register definitions, please refer to Enhanced Host Controller Interface Specification (EHCI), Revision 1.0 and Open Host Controller Interface Specification (OHCI), Revision 1.0a.

29.5 Interface Description

Table 29-2 USB2.0 PHY Interface Description

Module Pin	Direction	Pin Name	Descriptions
DP	I/O	HOST_DP	USB differential signal IO, positive end
DM	I/O	HOST_DM	USB differential signal IO, negative end
VCCA18	I/O	HOST_AVDD_1V8	1.8V power supply
RREF	I/O	HOST_EXTR	External 200ohm Bias Resistance, this is the 200ohm external resistance pin for termination.
AGND	I/O	VSS	Ground IO for both 3.3V, 1.8V and 0.8V
ACCA33	I/O	USB_AVDD3V3	3.3 V power supply(only Full speed driver)
VDDA	I/O	USB_AVDD_0V8	0.8 V power supply

29.6 Application Notes

29.6.1 Some Settings and Description About USB PHY

Here list some usbhost phy configure register of TOP_GRF in detail.

- GRF_USBPHY_CON2

Table 29-3 GRF_USBPHY_CON2 Description

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

Bit	Attr	Reset Value	Description
15	RW	0x0	usbhost_id_pull_up Software configure for HOSTPHY id_pull_up Enable ID pin sample. Signal that enables the sampling of the analog ID line. 1'b0: Sampling of ID pin is disabled (IDDIG force to low). 1'b1: Sampling of ID pin is enabled.
14	RW	0x0	usbhost_dischrgvbus Software configure for HOSTPHY dischrgvbus Discharge VBUS18. The signal enables discharging VBUS18. 1'b0: do not discharge VBUS18 through a resistor 1'b1: discharge VBUS18 through a resistor (this has to be active for at least 50ms)
13	RW	0x0	usbhost_chrgvbus Software configure for HOSTPHY chrgvbus Charge VBUS18. The signal enables charging VBUS18. 1'b0: do not charge VBUS18 through a resistor 1'b1: charge VBUS18 through a resistor (this has to be active for at least 30ms)
12	RW	0x0	usbhost_txbitstuff_enable Software configure for HOSTPHY txbitstuff_enable TX BITSTUFF ENABLE. Indicates if the data on the DATA_OUT[15:0]lines need to be bit stuffed or not. 1'b0: Bit stuffing is disabled 1'b1: Bit stuffing is enabled This signal is only used when OP_MODE[1:0] is set to 11b.
11	RW	0x1	usbhost_otg_suspendm Software configure for HOSTPHY suspend This signal is used for VBUS negotiation in OTG application and HOST disconnect in HOST application, in DEVICE mode this signal should be tie to 0. In OTG/HOST application this signal should be connected to 1. In HOST application, IDDIG is not valid, OTG_SUSPENDM must be set to 1 and OTG_SUSPENDM_BYPS must set to 1. In DEVICE application, IDDIG is not valid, OTG_SUSPENDM must be set to 0 and OTG_SUSPENDM_BYPS must set to 1.
10	RW	0x1	usbhost_data_bus16_8 Software configure for HOSTPHY data_bus16_8 Selects between 8 and 16-bit data transfer. 1'b1: 16-bit data path operation enabled, CLK60_30 = 30 MHz. 1'b0: 8-bit data path operation enabled, CLK60_30 = 60 MHz. This signal is sampled by USB PHY IP macro cell only when RESET becomes low. Default is 16-bit model and not suggest to configure this bit.

Bit	Attr	Reset Value	Description
9	RW	0x0	usbhost_pll_en Software configure for HOSTPHY pll_en PLL enable signal, internal PLL start to work when this signal is set to high, otherwise PLL is disabled and PHY IP stop working. This bit must be set to high before usb function.
8	RW	0x1	usbhost_chg_rst Software configure for HOSTPHY chg_rst Charge detector reset signal, active high, asynchronous. 1'b0: charge detector is normal operation 1'b1: charge detector is reset to initial
7	RW	0x0	usbhost_chg_en Software configure for HOSTPHY chg_en Charge detector enable signal. 1'b0: charging detect is off, IP is configured as SDP in HOST application or is configured as standard device in DEVICE application 1'b1: charging detect is on, IP is configured as CDP in HOST application or is configured as portable device in DEVICE application
6	RW	0x1	usbhost_termselect Software configure for HOSTPHY termselect Termination Select. This signal selects between the FS and HS terminations: 1'b0: HS termination enabled 1'b1: FS termination enabled In default this is controlled by hardware.
5:4	RW	0x1	usbhost_xcvrselect Software configure for HOSTPHY xcvrselect Transceiver Select. 2'b00: HS Transceiver enabled 2'b01: FS Transceiver enabled 2'b10: LS Transceiver enabled 2'b11: Reserved In default this is controlled by hardware.
3:2	RW	0x0	usbhost_opmode Software configure for HOSTPHY opmode Operational Mode. These signals select between various operational modes: 2'b00: Normal operation 2'b01: Non-driving. Transmitter buffer is in high impedance and pull-up/down resistors are disconnected 2'b10: Disable bit stuffing and NRZI encoding 2'b11: Reserved In default this is controlled by hardware.

Bit	Attr	Reset Value	Description
1	RW	0x0	usbhost_suspend_n Software configure for HOSTPHY suspend_n Active low, asynchronous. When enabled, drive IP into suspend mode and consume minimal current from power supply. 1'b0: IP in suspend mode, only those circuit that serve for resume function is still active 1'b1: IP in normal operation In default this is controlled by hardware.
0	RW	0x0	hostphy_input_sel HOSTPHY input selection. 1'b0: Hardware input 1'b1: Software input

29.6.2 USB Host Reset Sequence

In USB function mode, the recommended reset sequence is in figure1-2. 'RST' is the reset signal of USB controller and 'CLK' is the main clock of USB controller. At first, 'RST' is set to 1 and USB controller starts to work; secondly, USB controller outputs 'SUSPENDM'. 'RESET' is the reset of the usb phy. Both 'RST' and 'RESET' are controlled by CRU model.
'SUSPENDM' is set to 1 firstly to enable PLL, PLL starts to work and frequency starts locking, 'RESET' is set to 0 to enable clock lock detector, and 'CLK60_30' is valid when frequency is stable. PLL lock time(Tlock) is about 500us including the time of bandgap building, PLL locking and so on, Tlock also includes the PVT range.
'CLK_480' is controlled by 'PLL_EN', and 'CLK_480' is not valid when 'PLL_EN' is 0. Because 'PLL_EN' is the reset signal of delay control cell, so there must be a rise edge of 'PLL_EN' to make delay control cell to work correctly. Tdelay is the time of 'PLL_EN' rise edge to 'CLK_480' transmitting 480MHz clock, the value is greater than 200us and less than 500us.

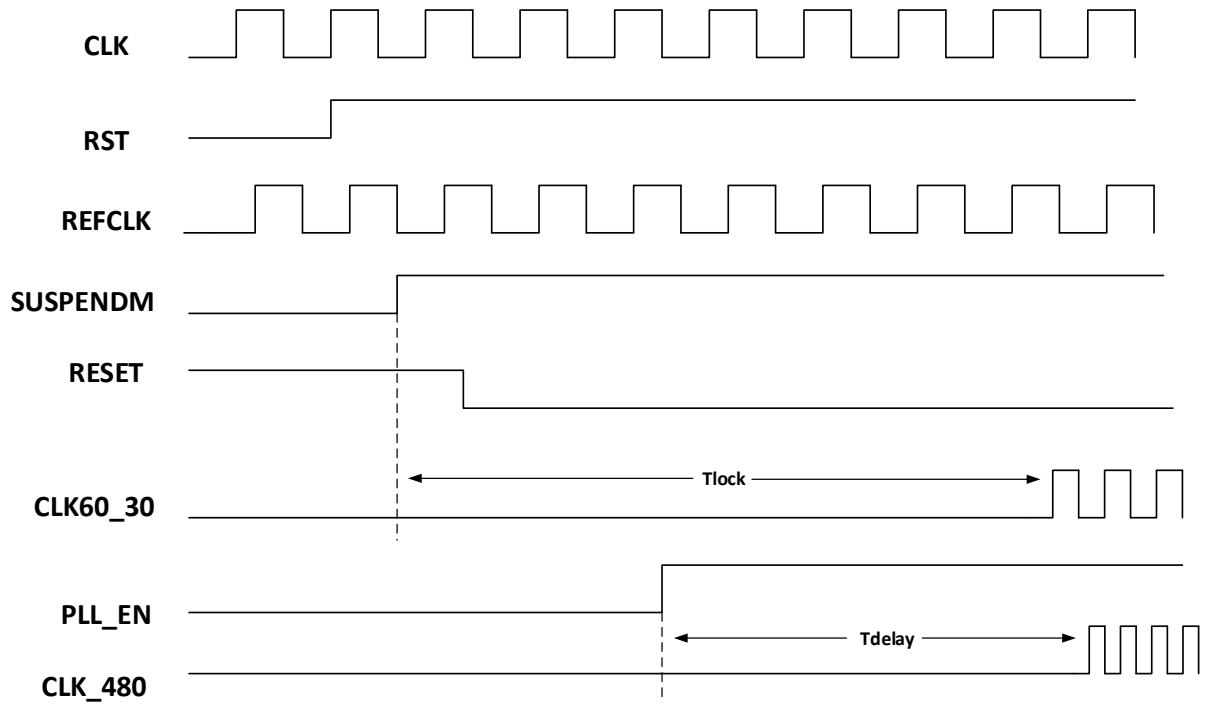


Fig. 29-2 USB2.0 Host reset sequence

29.6.3 Host Charge Detect Sequence

There are three steps in charge detector sequence including data contact detection, primary detection and secondary detection. Data contact detection is judging DEVICE are connected

or not, primary detection is judging whether the charger is connected or not, secondary detection is judging whether the charger is DCP or CDP.
When 'CHG_EN' is set to 1, 'CHG_RST' is set to 0 to start charge detection. This two signal can be control by GRF_USBPHY_CON2. During the detection, 'RESET' must be set to 1, the PHY function is disable before 'PHY_CONNECT' changes to 1. Tdelay is greater than 1ms.

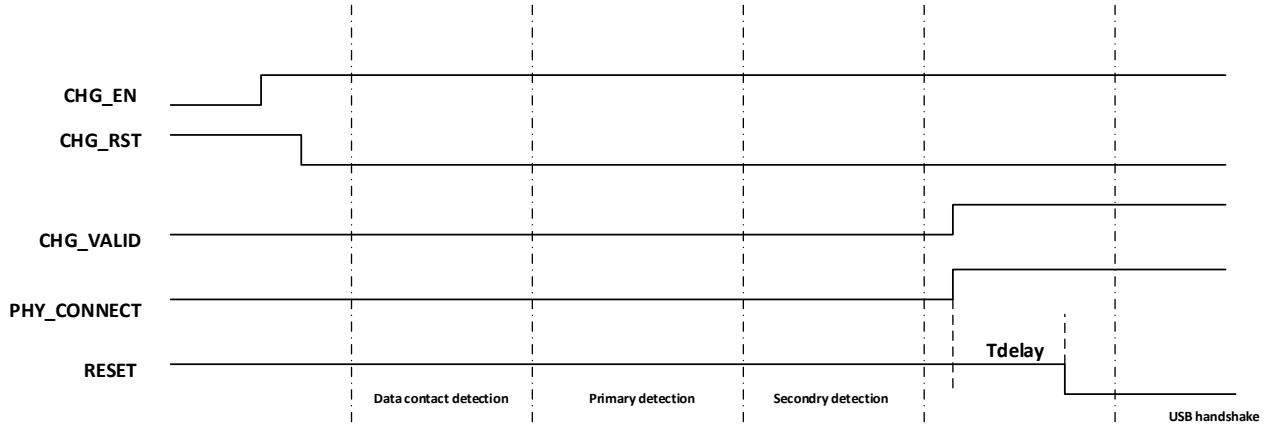


Fig. 29-3 USB2.0 Host charge sequence

Table 1-4 and the description about charge model. Host can be configured as DCP, CDP and SDP.

Table 29-4 Charge Description in host model

Host Port	Device Port	CHG_VALID	PHY_CONNECT
DCP	support charge	1	0
DCP	not support charge	0	0
CDP	support charge	1	1
CDP	not support charge	0	1
SDP	-	0	1

29.6.4 Program Flow

Please refer to Enhanced Host Controller Interface Specification (EHCI), Revision 1.0 and Open Host Controller Interface Specification (OHCI), Revision 1.0a.

Chapter 30 Gigabit Media Access Controller (GMAC)

30.1 Overview

The Ethernet Quality-of-Service controller(EQOS is commonly referred to as GMAC in this document) provides a complete Ethernet interface from processor to a Reduced Media Independent Interface (RMII) and Reduced Gigabit Media Independent Interface (RGMII) compliant Ethernet PHY. The GMAC includes a DMA controller. The DMA controller efficiently moves packet data from microprocessor's RAM, formats the data for an IEEE 802.3-2002 compliant packet and transmits the data to an Ethernet Physical Interface (PHY). It also efficiently moves packet data from RXFIFO to microprocessor's RAM.

30.1.1 MAC Features

30.1.1.1 MAC Tx and Rx Common Features

- Supports 10/100/1000-Mbps data transfer rates with the RGMII interfaces
- Supports 10/100-Mbps data transfer rates with the RMII interfaces
- Half-duplex operation:
  - CSMA/CD Protocol support
  - Flow control using backpressure support (based on implementation-specific white papers and UNH Ethernet Clause 4 MAC Test Suite - Annex D)
  - Packet bursting and packet extension in 1000 Mbps half-duplex operation
- Standard IEEE 802.3az-2010 for Energy Efficient Ethernet in Reduced Gigabit Media Independent Interface (RGMII) PHYs
- 64-bit data transfer interface on the application side
- Full-duplex flow control operations (IEEE 802.3x Pause packets and Priority flow control)
- network statistics with RMON or MIB Counters (RFC2819/RFC2665)
- Support Ethernet packet timestamping as described in IEEE 1588-2002 and IEEE 1588-2008 (64-bit timestamps given in the Tx or Rx status of PTP packet). Both one-step and two-step timestamping is supported in TX direction
- Media clock generation and recovery
- MDIO (Clause 22 and Clause 45) master interface for PHY device configuration and management

30.1.1.2 MAC Tx Features

- Preamble and start of packet data (SFD) insertion
- Separate 32-bit status for each packet transmitted from the application
- Automatic CRC and pad generation controllable on a per-packet basis
- Programmable packet length to support Standard or Jumbo Ethernet packets with up to 16 KB of size
- Programmable Inter Packet Gap (40–96 bit times in steps of 8)
- IEEE 802.3x Flow Control automatic transmission of zero-quanta Pause packet when flow control input transitions from assertion to de-assertion (in full-duplex mode)
- Frame Preemption for MAC Tx

30.1.1.3 MAC Rx Features

- Automatic Pad and CRC Stripping options
- Option to disable Automatic CRC checking
- Preamble and SFD deletion
- Separate 112-bit or 128-bit status
- Programmable watchdog timeout limit
- IEEE 802.1Q VLAN tag detection and option to delete the VLAN tags in received packets
- Optional module to detect remote wake-up packets and AMD magic packets
- Optional forwarding of received Pause packets to the application (in full-duplex mode)
- Frame Preemption for MAC Rx

30.1.2 MTL Features

30.1.2.1 MTL Tx and Rx Common Features

- 32-bit, 64-bit, or 128-bit Transaction Layer block (bridges the application and the MAC)
- Data transfers executed using simple FIFO protocol
- Synchronization for all clocks in the design (Transmit, Receive, and Application clocks)

- Optimization for packet-oriented transfers with packets delimiters
- Option to have dual-port RAM based asynchronous FIFO controllers or Single-port RAM based synchronous FIFO controllers
- RAM memory instantiation outside the top-level module to facilitate memory testing or instantiation
- Programmable burst length, up to half the size of the MTL Rx queue or Tx queue size, to support burst data transfer in the EQOS-MTL configuration
- Programmable threshold capability for each queue (default of 64 bytes)
- Optional Debug and slave mode operation on Queue 0 (default queue)

30.1.2.2 MTL Tx Features

- TX FIFO sizes on transmission is 16 KB
- Store-and-Forward mechanism or threshold mode (cut-through) for transmission to the MAC
- Programmable queue size in configurations with multiple queues. Each queue size can be programmed in terms of 256 bytes
- Automatic retransmission of collision packets in half-duplex mode
- Discard packets on late collision, excessive collisions, excessive deferral, and under-run conditions with appropriate status
- Disabling of Data Memory RAM chip-select when inactive to reduce power consumption
- Optional module to calculate and insert IPv4 header checksum and TCP, UDP, or ICMP checksum
- Programmable interrupt options for different operational conditions
- Statistics by generating pulses for packets dropped (because of underflow) in the Tx FIFO

30.1.2.3 MTL Rx Features

- Rx queue sizes in the Receive path is 32 KB
- Insertion of Rx Status vectors into the Rx queue after the EOP transfer (in Threshold mode) and before SOP (in Store-and-Forward mode) in EQOS-MTL configuration
- Programmable Rx queue threshold (default fixed at 64 bytes) in Threshold (or cut-through) mode
- Option to filter all error packets on reception and not forward them to the application in the store-and-forward mode
- Option to forward the undersized good packets
- Statistics by generating pulses for packets dropped (because of overflow) in the Rx FIFO
- Automatic generation of Pause packet control or backpressure signal to the MAC based on the Rx Queue fill level
- Option to replicate received multicast packets for transfer by multiple Rx DMA channels
- Option to have a programmable lookup table based flexible Parser for filtering and steering the Rx packets

30.1.3 DMA Block Features

- 64-bit data transfers
- Separate DMA channel in the Transmit path for each queue in MTL
- Single or multiple DMA channels for any number of queues in MTL Receive path
- Fully synchronous design operating on a single application clock (except for CSR module, when a separate CSR clock is configured)
- Optimization for packet-oriented DMA transfers with packet delimiters
- Byte-aligned addressing for data buffer support
- Dual-buffer (ring) descriptor support
- Descriptor architecture to allow large blocks of data transfer with minimum CPU intervention (each descriptor can transfer up to 32 KB of data)
- Comprehensive status reporting for normal operation and transfers with errors
- Individual programmable burst length for Tx DMA and Rx DMA engines for optimal host bus utilization
- Programmable interrupt options for different operational conditions
- Per-packet Transmit or Receive Complete Interrupt control
- Round-robin or fixed-priority arbitration between the Receive and Transmit engines
- Start and Stop modes

- Separate ports for host CSR access and host data interface
- support for TCP Segmentation Offload (TSO) and UDP Segmentation Offload (USO)
- Selectable number of Tx DMA channels with TSO/USO feature enabled
- Time-sensitive conditional packet fetching from system memory by comparing the Slot Time or IEEE 1588 time information provided in the descriptor (useful for AV applications)
- Programmable control for Transmit Descriptor posted writes to improve the throughput
- Sideband signals to control starting and stopping of DMA channels

30.2 Block Diagram

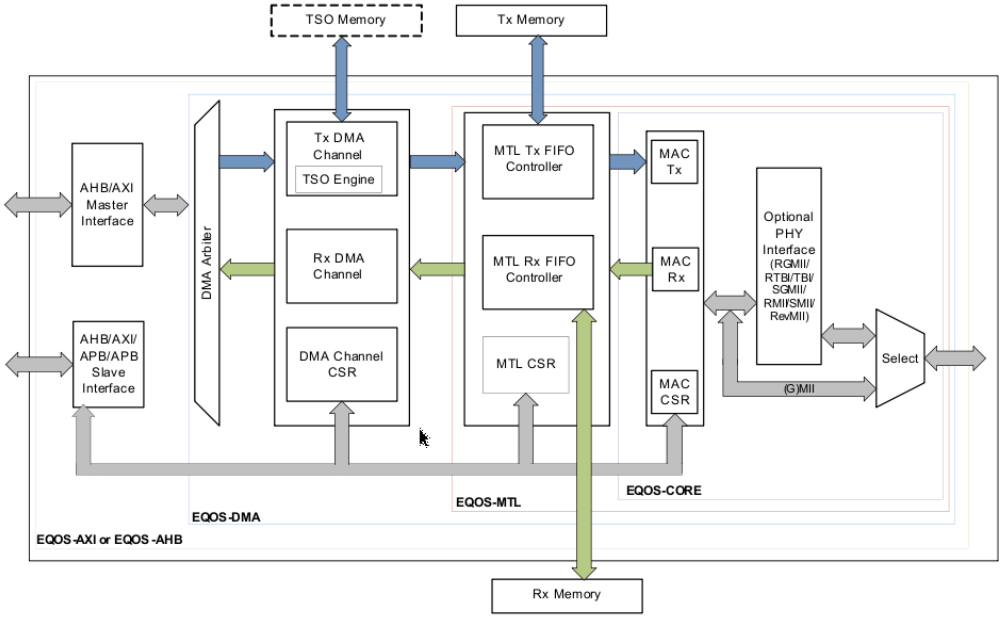


Fig. 30-1 GMAC Block Diagram

The GMAC is broken up into multiple separate functional units. These blocks are interconnected in the MAC module. The block diagram shows the general flow of data and control signals between these blocks.

The GMAC transfers data to system memory through the AXI master interface. The host CPU uses the APB Slave interface to access the GMAC subsystem's control and status registers (CSRs).

The GMAC supports the PHY interfaces of reduced GMII (RGMII) and reduced MII (RMII).

The Transmit FIFO (Tx FIFO) buffers data read from system memory by the DMA before transmission by the GMAC Core. Similarly, the Receive FIFO (Rx FIFO) stores the Ethernet frames received from the line until they are transferred to system memory by the DMA. These are asynchronous FIFOs, as they also transfer the data between the application clock and the GMAC line clocks.

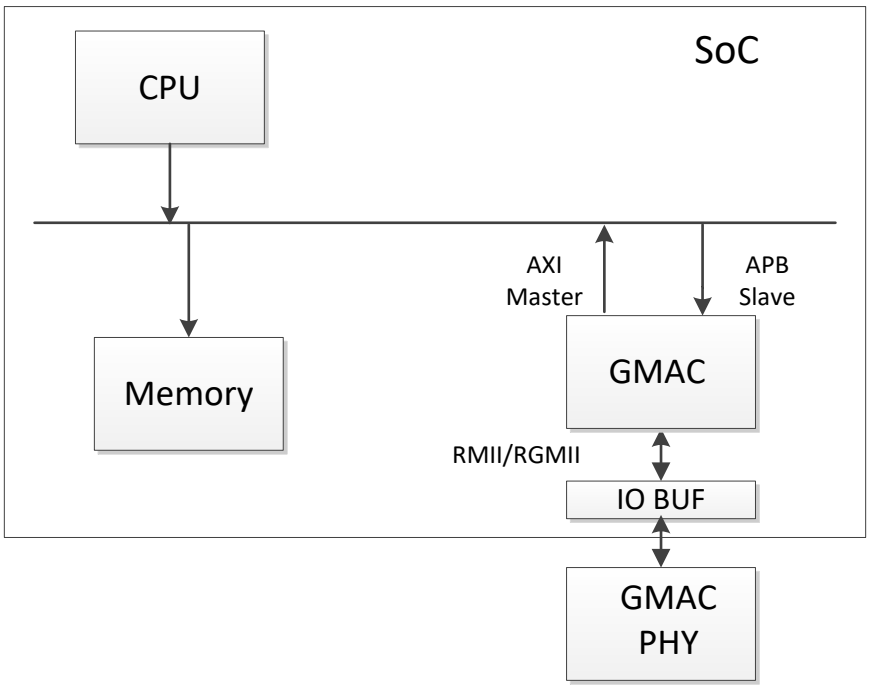


Fig. 30-2 GMAC in SOC

GMAC Supports 10/100/1000-Mbps data transfer rates with the RGMII interfaces and Supports 10/100-Mbps data transfer rates with the RMII interfaces.

30.3 Function Description

30.3.1 Frame Structure

Data frames transmitted shall have the frame format shown in Fig.1-3.

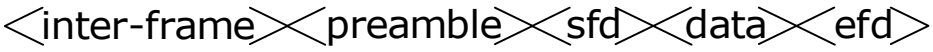


Fig. 30-3 Frame Format

The preamble <preamble> begins a frame transmission. The bit value of the preamble field consists of 7 octets with the following bit values:
10101010 10101010 10101010 10101010 10101010 10101010 10101010
The SFD (start frame delimiter) <sfd> indicates the start of a frame and follows the preamble. The bit value is 10101011.The data in a well formed frame shall consist of N octet's data.

30.3.1.1 RMII Interface timing diagram

The Reduced Media Independent Interface (RMII) specification reduces the pin count between Ethernet PHYs and Switch ASICs (only in 10/100 mode). According to the IEEE 802.3u standard, an MII contains 16 pins for data and control. In devices incorporating multiple MAC or PHY interfaces (such as switches), the number of pins adds significant cost with increase in port count. The RMII specification addresses this problem by reducing the pin count to 7 for each port - a 62.5% decrease in pin count.
The RMII module is instantiated between the GMAC and the PHY. This helps translation of the MAC's MII into the RMII. The RMII block has the following characteristics:
Supports 10-Mbps and 100-Mbps operating rates. It does not support 1000-Mbps operation.
Two clock references are sourced externally or CRU, providing independent, 2-bit wide transmit and receive paths.

30.3.1.2 Transmit Bit Ordering

Each nibble from the MII must be transmitted on the RMII a di-bit at a time with the order of di-bit transmission shown in Fig.1-4. The lower order bits (D1 and D0) are transmitted first followed by higher order bits (D2 and D3).

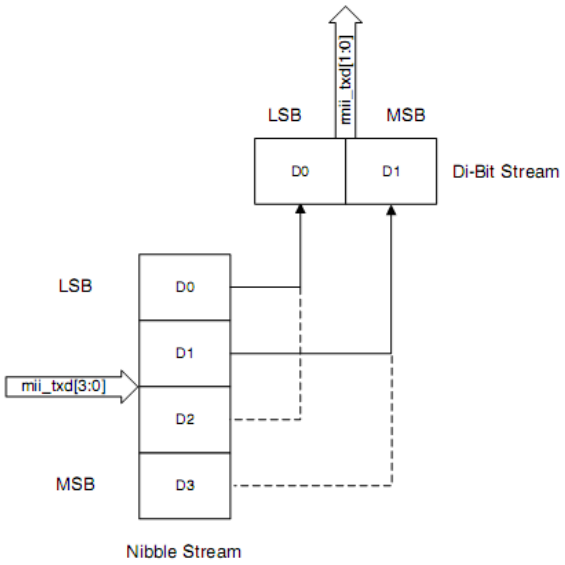


Fig. 30-4 RMII Transmission Bit Ordering

30.3.1.3 RMII Transmit Timing Diagrams

Fig.1-5 through 1-8 show MII-to-RMII transaction timing. The clk_rmii_i (REF_CLK) frequency is 50MHz in RMII interface. In 10Mb/s mode, as the REF_CLK frequency is 10 times as the data rate, the value on rmii_txd_o[1:0] (TXD[1:0]) shall be valid such that TXD[1:0] may be sampled every 10th cycle, regard-less of the starting cycle within the group and yield the correct frame data.

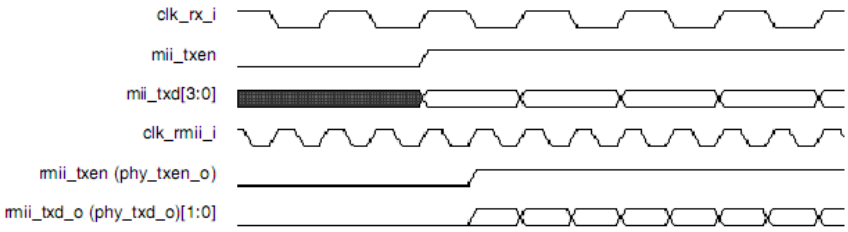


Fig. 30-5 Start of MII and RMII Transmission in 100-Mbps Mode

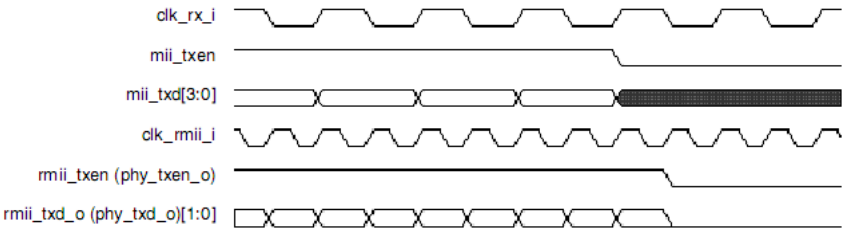


Fig. 30-6 End of MII and RMII Transmission in 100-Mbps Mode

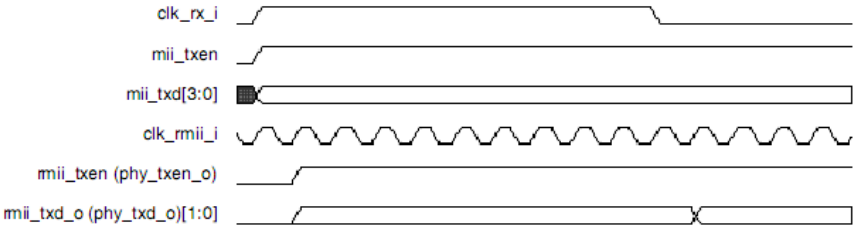


Fig. 30-7 Start of MII and RMII Transmission in 10-Mbps Mode

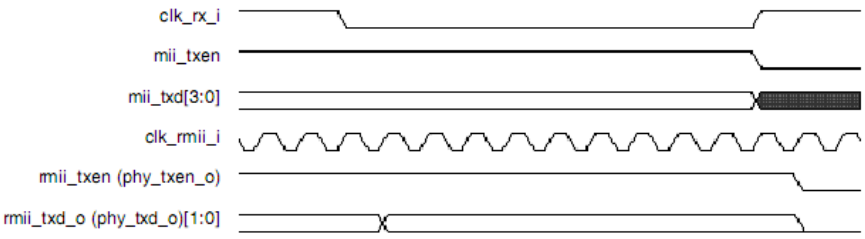


Fig. 30-8 End of MII and RMII Transmission in 10-Mbps Mode

30.3.1.4 Receive Bit Ordering

Each nibble is transmitted to the MII from the di-bit received from the RMII in the nibble transmission order shown in Fig.1-9. The lower order bits (D0 and D1) are received first, followed by the higher order bits (D2 and D3).

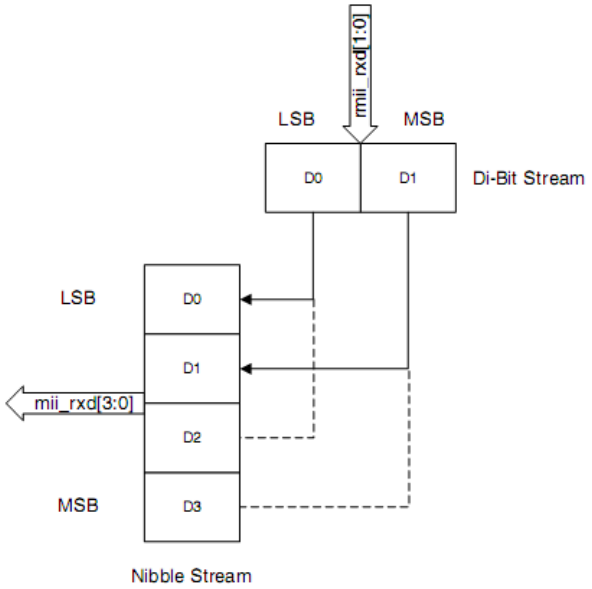


Fig. 30-9 RMII Receive Bit Ordering

30.3.1.5 RGMII Interface

The Reduced Gigabit Media Independent Interface (RGMII) specification reduces the pin count of the interconnection between the GMAC 10/100/1000 controller and the PHY for GMII and MII interfaces. To achieve this, the data path and control signals are reduced and multiplexed together with both the edges of the transmission and receive clocks. For gigabit operation the clocks operate at 125 MHz; for 10/100 operation, the clock rates are 2.5 MHz/25 MHz.

In the GMAC 10/100/1000 controller, the RGMII module is instantiated between the GMAC core's GMII and the PHY to translate the control and data signals between the GMII and RGMII protocols.

The RGMII block has the following characteristics:

- Supports 10-Mbps, 100-Mbps, and 1000-Mbps operation rates
- For the RGMII block, no extra clock is required because both the edges of the incoming clocks are used
- The RGMII block extracts the in-band (link speed, duplex mode and link status) status signals from the PHY and provides them to the GMAC core logic for link detection

30.3.2 Station Management Agent

The application can access the PHY registers through the Station Management Agent (SMA) module. SMA is a two-wire Station Management interface (MIM).

For MIM accesses, the maximum operating frequency of the MDC (gmii_mdc_o) is 2.5 MHz, as specified in the IEEE 802.3. In the GMAC core, the gmii_mdc_o clock is derived from the application clock or clk_csr_i, using a divider-counter. The divide factor depends on the clock range setting (CR field) in the MAC_MDIO_Address register Select the clock divide factor as mentioned in the description of CR field of MAC_MDIO_Address register, to meet IEEE specifications. However, if your system supports higher clock frequencies on the MIM interface, there is a provision to select a different divider.

The MDIO frame structure is as follows:

Table 30-1 MDIO Clause 45 Frame Structure

Field	Description
IDLE	The mdio line is in tri-state; there is no clock on gmii_mdc_o signal.
PREAMBLE	32 continuous bits of value 1
START	Start of packet is 2'b00
OPCODE	■ 2'b00 ■ 2'b01 ■ 2'b10 ■ 2'b11
PHY ADDR	5-bit address select for one of 32 PHYs
DEV ADDR	5-bit address select for one of 32 devices
TA	Turnaround ■ 2'bZ0: Read and post-read increment address ■ 2'b10: Write and address MDIO accesses Where Z is the tri-state level
DATA/ADDRESS	16-bit value: For an address cycle (OPCODE = 2'b00), this frame contains the address of the register to be accessed on the next cycle. For the data cycle of a write frame, this field contains the data to be written to the register. For read or post-read increment address frames, this field contains the contents of the register read from the PHY. ■ In address and data write cycles, the GMAC drives the MDIO line during the transfer of these 16 bits. ■ In read and post-read increment address cycles, the PHY drives the MDIO line during the transfer of these 16 bits.

The frame structure for Clause 22 frames is also supported. The C45E bit in the MAC_MDIO_Address register can be programmed to enable Clause 22 or Clause 45 mode of operation. Table.1-2 shows the Clause 22 frame format.

Table 30-2 MDIO Clause 22 Frame Structure

Field	Description
IDLE	The mdio line is in tri-state; there is no clock on gmii_mdc_o signal.
PREAMBLE	32 continuous bits of value 1
START	Start of packet is 2'b01
OPCODE	■ 2'b01 : Write ■ 2'b10 : Read
PHY ADDR	5-bit address select for one of 32 PHYs
DEV ADDR	5-bit address to select the register within each MMD
TA	Turnaround ■ 2'bZ0: Read and post-read increment address ■ 2'b10: Write and address MDIO accesses Where Z is the tri-state level
DATA/ADDRESS	Any 16-bit value: ■ In a write operation, the GMAC drives MDIO ■ In read operation, the PHY drives MDIO

In addition to normal read and write operations, the SMA also supports post-read increment address while operating in Clause 45 mode.

30.3.3 Power Management and Energy Efficient Ethernet

The GMAC supports the following techniques to save power.

- Magic Packet
- Remote Wakeup
- Energy Efficient Ethernet

The Magic Packet and Remove Wakeup techniques are used to save power in the host system when it is idle (Sleep mode) and has to be woken up only at the reception of specific packets from the Ethernet network. In the Sleep mode, the power to the host logic along with majority of the GMAC (except the MAC receiver logic), can be shut down. On receiving the specific packets from the network, the MAC provides the trigger to restore the power to the host system and come back to normal state.

The Energy Efficient Ethernet (EEE) mode is compliant with the IEEE 802.3az-2010 standard. It is primarily targeted to save power in the Ethernet port when there is no traffic on the line. In this mode, the host indicates to the far-end that it does not have any packets to transmit for near future and puts the transmitter port (MAC Controller, PCS and PHY layers) into low-power mode. Similarly, the Receiver port can also be put into low-power mode when the far-end host indicates that it does not have any traffic to transfer. This allows significant saving of power in the Ethernet port (mainly in the PHY) with intermittent and bursty traffic profile. The triggering of entry and exit out of the EEE mode is controlled by the MAC and is supported within the GMAC.

Simultaneous operation of the EEE mode along with any or both the other power saving modes is also supported in GMAC.

30.3.4 IEEE 1588 Timestamp Support

The IEEE 1588 defines a Precision Time Protocol (PTP) which enables precise synchronization of time in measurement and control systems. This protocol enables heterogeneous systems that include clocks of varying inherent precision, resolution, and stability to synchronize. The protocol supports system-wide synchronization accuracy in the sub-microsecond range with minimal network and local clock computing resources.

The GMAC supports the IEEE 1588-2002 (version 1) and IEEE 1588-2008 (version 2). The IEEE 1588-2002 supports PTP transported over UDP/IP and IEEE 1588-2008 supports PTP transported over Ethernet. The GMAC provides programmable support for both standards.

The controller supports the following features:

- Provides an option to take snapshot of all packets or only PTP type packets
- Provides an option to take snapshot of only event messages
- Provides an option to take the snapshot based on the clock type: ordinary, boundary, end-to-end transparent, and peer-to-peer transparent
- Provides an option to select the node to be a master or slave for ordinary and boundary clock
- Identifies the PTP message type, version, and PTP payload in packets sent directly over Ethernet and sends the status
- Provides an option to measure sub-second time in digital or binary format

30.3.5 TCP/IP Segmentation Offload (TSO) Engine

The TCP Segmentation Offload (TSO) engine is useful in offloading the TCP segmentation functions to the hardware.

It also supports UDP Segmentation Offload (USO) in which the UDP payload is segmented in the hardware. There are no sequencing/ordering controls available or updated in the segments, so it can be used in point to point applications in which out of order packets are not expected by the receiver. The description and flow of TSO mentioned in this section is same as USO, any deviation is provided as notes.

In the TCP segmentation offload (TSO) feature, the DMA splits a large TCP packet into multiple small packets and passes these packets to the MTL Tx Queue.

30.3.6 MAC Management Counters

The GMAC supports storing the statistics about the received and transmitted packets in registers that are accessible through the application.

The counters in the MAC Management Counters (MMC) module can be viewed as an extension of the register address space of the CSR module. The MMC module maintains a set of registers for gathering statistics on the received and transmitted packets. The register set includes a control register for controlling the behavior of the registers, two status registers containing interrupts generated (receive and transmit), and Interrupt Enable registers (receive and transmit). These registers are accessible from the Application through the MAC Control Interface (MCI). Each register is 32-bits wide. The write data is qualified with the corresponding mci_be_i signals. Therefore, non-32-bit accesses are allowed as long as the address is word-aligned. The MMCs are accessed using transactions, in the same way the CSR address space is accessed.
The MMC counters are free running. There is no separate enable for the counters to start. If a particular MMC counter is present in the RTL, it starts counting when corresponding packet is received or transmitted. The Receive MMC counters are updated for packets that are passed by the Address Filter (AFM) block. The statistics of packets, dropped by the AFM module, are not updated unless they are runt packets of less than 6 bytes (DA bytes are not received fully). To get statistics of all packets, set Bit 0 in the “MAC_Packet_Filter” register.
The MMC module gathers statistics on encapsulated IPv4, IPv6, TCP, UDP, or ICMP payloads in received Ethernet packets.

30.4 Register Description

30.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
GMAC_MAC_Configuration	0x0000	W	0x00000000	The MAC Configuration Register establishes the operating mode of the MAC
GMAC_MAC_Ext_Configuration	0x0004	W	0x00000000	The MAC Extended Configuration Register establishes the operating mode of the MAC
GMAC_MAC_Packet_Filter	0x0008	W	0x00000000	The MAC Packet Filter register contains the filter controls for receiving packets
GMAC_MAC_Watchdog_Timeout	0x000c	W	0x00000000	The Watchdog Timeout register controls the watchdog timeout for received packets
GMAC_MAC_Hash_Table_Reg0	0x0010	W	0x00000000	The Hash Table Register 0 contains the first 32 bits of the hash table
GMAC_MAC_Hash_Table_Reg1	0x0014	W	0x00000000	The Hash Table Register 1 contains the second 32 bits of the hash table
GMAC_MAC_VLAN_Tag	0x0050	W	0x00000000	The VLAN Tag register identifies the IEEE 802.1Q VLAN type packets
GMAC_MAC_Q0_Tx_Flow_Ctrl	0x0070	W	0x00000000	The Flow Control register controls the generation and reception of the Control (Pause Command) packets by the Flow control module of the MAC
GMAC_MAC_Rx_Flow_Ctrl	0x0090	W	0x00000000	The Receive Flow Control register controls the pausing of MAC Transmit based on the received Pause packet
GMAC_MAC_Interrupt_Status	0x00b0	W	0x00000000	The Interrupt Status register contains the status of interrupts

Name	Offset	Size	Reset Value	Description
GMAC_MAC_Interrupt_Enable	0x00b4	W	0x00000000	The Interrupt Enable register contains the masks for generating the interrupts
GMAC_MAC_Rx_Tx_Status	0x00b8	W	0x00000000	The Receive Transmit Status register contains the Receive and Transmit Error status
GMAC_MAC_PMT_Control_Status	0x00c0	W	0x00000000	The PMT Control and Status Register
GMAC_MAC_RWK_Packet_Filter	0x00c4	W	0x00000000	The Remote Wakeup Filter registers are implemented as 8, 16, or 32 indirect access registers (wkuppktfilter_reg#i) based on whether 4, 8, or 16 Remote Wakeup Filters are selected in the configuration and accessed by application through MAC_RWK_Packet_Filter register
GMAC_RWK_Filter0_Byte_Mask	0x10c0	W	0x00000000	RWK Filter0 Byte Mask
GMAC_RWK_Filter1_Byte_Mask	0x10c4	W	0x00000000	RWK Filter1 Byte Mask
GMAC_RWK_Filter2_Byte_Mask	0x10c8	W	0x00000000	RWK Filter2 Byte Mask
GMAC_RWK_Filter3_Byte_Mask	0x10cc	W	0x00000000	RWK Filter3 Byte Mask
GMAC_RWK_Filter01_CRC	0x10d0	W	0x00000000	RWK Filter 0/1 CRC-16
GMAC_RWK_Filter23_CRC	0x10d4	W	0x00000000	RWK Filter 2/3 CRC-16
GMAC_RWK_Filter_Offset	0x10d8	W	0x00000000	RWK Filter Offset
GMAC_RWK_Filter_Command	0x10dc	W	0x00000000	RWK Filter Command
GMAC_MAC_LPI_Control_Status	0x00d0	W	0x00000000	The LPI Control and Status Register controls the LPI functions and provides the LPI interrupt status. The status bits are cleared when this register is read
GMAC_MAC_LPI_Timers_Control	0x00d4	W	0x03e80000	The LPI Timers Control register controls the timeout values in the LPI states.
GMAC_MAC_LPI_Entry_Timer	0x00d8	W	0x00000000	This register controls the Tx LPI entry timer
GMAC_MAC_1US_Tic_Counter	0x00dc	W	0x00000063	This register controls the generation of the Reference time (1 microsecond tic)
GMAC_MAC_PHYIF_Control_Status	0x00f8	W	0x00000000	PHY Interface Control and Status Register
GMAC_MAC_Version	0x0110	W	0x00003051	The version register identifies the version of the DWC_ether_qos

Name	Offset	Size	Reset Value	Description
GMAC_MAC_Debug	0x0114	W	0x00000000	The Debug register provides the debug status of various MAC blocks
GMAC_MAC_HW_Feature0	0x011c	W	0x060171e7	This register indicates the presence of first set of the optional features or functions
GMAC_MAC_HW_Feature1	0x0120	W	0x010c01c8	This register indicates the presence of second set of the optional features or functions
GMAC_MAC_HW_Feature2	0x0124	W	0x10000000	This register indicates the presence of third set of the optional features or functions
GMAC_MAC_HW_Feature3	0x0128	W	0x00000000	This register indicates the presence of fourth set the optional features or functions
GMAC_MAC_MDIO_Address	0x0200	W	0x00000000	The MDIO Address register controls the management cycles to external PHY through a management interface
GMAC_MAC_MDIO_Data	0x0204	W	0x00000000	The MDIO Data register stores the Write data to be written to the PHY register located at the address specified in MAC_MDIO_Address
GMAC_MAC_CSR_SW_Ctrl	0x0230	W	0x00000000	This register contains SW programmable controls for changing the CSR access response and status bits clearing
GMAC_MAC_Address0_High	0x0300	W	0x8000ffff	The MAC Address0 High register holds the upper 16 bits of the first 6-byte MAC address of the station
GMAC_MAC_Address0_Low	0x0304	W	0xffffffff	The MAC Address0 Low register holds the lower 32 bits of the 6-byte first MAC address of the station
GMAC_MMC_Control	0x0700	W	0x00000000	This register establishes the operating mode of MMC
GMAC_MMC_Rx_Interrupt	0x0704	W	0x00000000	Maintains the interrupts generated from all Receive statistics counters
GMAC_MMC_Tx_Interrupt	0x0708	W	0x00000000	Maintains the interrupts generated from all Transmit statistics counters
GMAC_MMC_Rx_Interrupt_Mask	0x070c	W	0x00000000	This register maintains the masks for interrupts generated from all Receive statistics counters
GMAC_MMC_Tx_Interrupt_Mask	0x0710	W	0x00000000	This register maintains the masks for interrupts generated from all Transmit statistics counters

Name	Offset	Size	Reset Value	Description
GMAC_Tx_Octet_Count_Good_Bad	0x0714	W	0x00000000	This register provides the number of bytes transmitted by the DWC_ether_qos, exclusive of preamble and retried bytes, in good and bad packets
GMAC_Tx_Packet_Count_Good_Bad	0x0718	W	0x00000000	This register provides the number of good and bad packets, exclusive of retried packets
GMAC_Tx_Underflow_Error_Packets	0x0748	W	0x00000000	This register provides the number of packets aborted because of packets underflow error
GMAC_Tx_Carrier_Error_Packets	0x0760	W	0x00000000	This register provides the number of packets aborted because of carrier sense error (no carrier or loss of carrier)
GMAC_Tx_Octet_Count_Good	0x0764	W	0x00000000	This register provides the number of bytes exclusive of preamble, only in good packets
GMAC_Tx_Packet_Count_Good	0x0768	W	0x00000000	This register provides the number of good packets transmitted by DWC_ether_qos
GMAC_Rx_Packets_Count_Good_Bad	0x0780	W	0x00000000	This register provides the number of good and bad packets received by DWC_ether_qos
GMAC_Rx_Octet_Count_Good_Bad	0x0784	W	0x00000000	This register provides the number of bytes received by DWC_ether_qos, exclusive of preamble, in good and bad packets
GMAC_Rx_Octet_Count_Good	0x0788	W	0x00000000	This register provides the number of bytes received by DWC_ether_qos, exclusive of preamble, only in good packets
GMAC_Rx_Multicast_Packets_Good	0x0790	W	0x00000000	This register provides the number of good multicast packets received by
GMAC_Rx_CRC_Error_Packets	0x0794	W	0x00000000	This register provides the number of packets received by DWC_ether_qos with CRC error
GMAC_Rx_Length_Error_Packets	0x07c8	W	0x00000000	This register provides the number of packets received by DWC_ether_qos with length error (Length Type field not equal to packet size), for all packets with valid length field
GMAC_Rx_FIFO_Overflow_Packets	0x07d4	W	0x00000000	This register provides the number of missed received packets because of FIFO overflow
GMAC_MMC_IPC_Rx_Interrupt_Mask	0x0800	W	0x00000000	This register maintains the mask for the interrupt generated from the receive IPC statistic counters

Name	Offset	Size	Reset Value	Description
GMAC_MMC_IPC_Rx_Interrupt	0x0808	W	0x00000000	This register maintains the interrupt that the receive IPC statistic counters generate
GMAC_RxIPv4_Good_Packets	0x0810	W	0x00000000	This register provides the number of good IPv4 datagrams received by DWC_ether_qos with the TCP, UDP, or ICMP payload
GMAC_RxIPv4_Header_Error_Packets	0x0814	W	0x00000000	This register provides the number of IPv4 datagrams received by DWC_ether_qos with header (checksum, length, or version mismatch) errors
GMAC_RxIPv6_Good_Packets	0x0824	W	0x00000000	This register provides the number of good IPv6 datagrams received by DWC_ether_qos
GMAC_RxIPv6_Header_Error_Packets	0x0828	W	0x00000000	This register provides the number of IPv6 datagrams received by DWC_ether_qos with header (length or version mismatch) errors
GMAC_RxUDP_Error_Packets	0x0834	W	0x00000000	This register provides the number of good IP datagrams received by DWC_ether_qos whose UDP payload has a checksum error
GMAC_RxTCP_Error_Packets	0x083c	W	0x00000000	This register provides the number of good IP datagrams received by DWC_ether_qos whose TCP payload has a checksum error
GMAC_RxICMP_Error_Packets	0x0844	W	0x00000000	This register provides the number of good IP datagrams received by DWC_ether_qos whose ICMP payload has a checksum error
GMAC_RxIPv4_Header_Error_Octets	0x0854	W	0x00000000	This register provides the number of bytes received by DWC_ether_qos in IPv4 datagrams with header errors (checksum, length, version mismatch)
GMAC_RxIPv6_Header_Error_Octets	0x0868	W	0x00000000	This register provides the number of bytes received by DWC_ether_qos in IPv6 datagrams with header errors (length, version mismatch)
GMAC_RxUDP_Error_Octets	0x0874	W	0x00000000	This register provides the number of bytes received by DWC_ether_qos in a UDP segment that had checksum errors

Name	Offset	Size	Reset Value	Description
GMAC_RxTCP_Error_Octets	0x087c	W	0x00000000	This register provides the number of bytes received by DWC_ether_qos in a TCP segment that had checksum errors
GMAC_RxICMP_Error_Octets	0x0884	W	0x00000000	This register provides the number of bytes received by DWC_ether_qos in a good ICMP segment
GMAC_MAC_Timestamp_Control	0x0b00	W	0x00002000	This register controls the operation of the System Time generator and processing of PTP packets for timestamping in the Receiver
GMAC_MAC_Sub_Second_Increment	0x0b04	W	0x00000000	Specifies the value to be added to the internal system time register every cycle of clk_ptp_ref_i clock
GMAC_MAC_System_Time_Secs	0x0b08	W	0x00000000	The System Time Nanoseconds register, along with System Time Seconds register, indicates the current value of the system time maintained by the MAC
GMAC_MAC_System_Time_NS	0x0b0c	W	0x00000000	The System Time Nanoseconds register, along with System Time Seconds register, indicates the current value of the system time maintained by the MAC
GMAC_MAC_Sys_Time_Secs_Update	0x0b10	W	0x00000000	The System Time Seconds Update register, along with the System Time Nanoseconds Update register, initializes or updates the system time maintained by the MAC
GMAC_MAC_Sys_Time_NS_Update	0x0b14	W	0x00000000	MAC System Time Nanoseconds Update register
GMAC_MAC_Timestamp_Addend	0x0b18	W	0x00000000	Timestamp Addend register. This register value is used only when the system time is configured for Fine Update mode (TSCFUPDT bit in the MAC_Timestamp_Control register)
GMAC_MAC_Timestamp_Status	0x0b20	W	0x00000000	Timestamp Status register. All bits except Bits[27:25] gets cleared when the application reads this register
GMAC_MAC_Tx_TS_Status_NS	0x0b30	W	0x00000000	This register contains the nanosecond part of timestamp captured for Transmit packets when Tx status is disabled

Name	Offset	Size	Reset Value	Description
GMAC_MAC_Tx_TS_Status_Secs	0x0b34	W	0x00000000	The register contains the higher 32 bits of the timestamp (in seconds) captured when a PTP packet is transmitted
GMAC_MAC_Auxiliary_Control	0x0b40	W	0x00000000	The Auxiliary Timestamp Control register controls the Auxiliary Timestamp snapshot
GMAC_MAC_Auxiliary_TS_NS	0x0b48	W	0x00000000	The Auxiliary Timestamp Nanoseconds register, along with MAC_Auxiliary_Timestamp_Seconds, gives the 64-bit timestamp stored as auxiliary snapshot
GMAC_MAC_Auxiliary_TS_Secs	0x0b4c	W	0x00000000	The Auxiliary Timestamp - Seconds register contains the lower 32 bits of the Seconds field of the auxiliary timestamp register
GMAC_MAC_TS_Ingress_Corr_NS	0x0b58	W	0x00000000	This register contains the correction value in nanoseconds to be used with the captured timestamp value in the ingress path
GMAC_MAC_TS_Egress_Corr_NS	0x0b5c	W	0x00000000	This register contains the correction value in nanoseconds to be used with the captured timestamp value in the egress path
GMAC_MAC_TS_Ingress_Latency	0x0b68	W	0x00000000	This register holds the Ingress MAC latency
GMAC_MAC_TS_Egress_Latency	0x0b6c	W	0x00000000	This register holds the Egress MAC latency
GMAC_MAC_PPS_Control	0x0b70	W	0x00000000	PPS Control register
GMAC_MTL_DBG_CTL	0x0c08	W	0x00000000	The FIFO Debug Access Control and Status register controls the operation mode of FIFO debug access
GMAC_MTL_DBG_STS	0x0c0c	W	0x00000018	The FIFO Debug Status register contains the status of FIFO debug access
GMAC_MTL_FIFO_Debug_Data	0x0c10	W	0x00000000	The FIFO Debug Data register contains the data to be written to or read from the FIFOs
GMAC_MTL_Interrupt_Status	0x0c20	W	0x00000000	The software driver (application) reads this register during interrupt service routine or polling to determine the interrupt status of MTL queues and the MAC
GMAC_MTL_TxQ0_Operation_Mode	0x0d00	W	0x003f0008	The Queue 0 Transmit Operation Mode register establishes the Transmit queue operating modes and commands

Name	Offset	Size	Reset Value	Description
GMAC_MTL_TxQ0_Underflow	0x0d04	W	0x00000000	The Queue 0 Underflow Counter register contains the counter for packets aborted because of Transmit queue underflow and packets missed because of Receive queue packet flush
GMAC_MTL_TxQ0_Debug	0x0d08	W	0x00000000	The Queue 0 Transmit Debug register gives the debug status of various blocks related to the Transmit queue
GMAC_MTL_Q0_Interrupt_Ctrl_Status	0x0d2c	W	0x00000000	This register contains the interrupt enable and status bits for the queue 0 interrupts
GMAC_MTL_RxQ0_Operation_Mode	0x0d30	W	0x07f00000	The Queue 0 Receive Operation Mode register establishes the Receive queue operating modes and command
GMAC_MTL_RxQ0_Miss_Pkt_Ovf_Cnt	0x0d34	W	0x00000000	The Queue 0 Missed Packet and Overflow Counter register contains the counter for packets missed because of Receive queue packet flush and packets discarded because of Receive queue overflow
GMAC_MTL_RxQ0_Debug	0x0d38	W	0x00000010	The Queue 0 Receive Debug register gives the debug status of various blocks related to the Receive queue
GMAC_DMA_Mode	0x1000	W	0x00000001	The Bus Mode register establishes the bus operating modes for the DMA
GMAC_DMA_SysBus_Mode	0x1004	W	0x01010000	The System Bus mode register controls the behavior of the AHB or AXI master
GMAC_DMA_Interrupt_Status	0x1008	W	0x00000000	The application reads this Interrupt Status register during interrupt service routine or polling to determine the interrupt status of DMA channels, MTL queues, and the MAC
GMAC_DMA_Debug_Status0	0x100c	W	0x00000000	The Debug Status 0 register gives the Receive and Transmit process status for DMA Channel 0-Channel 2 for debugging purpose
GMAC_AXI_LPI_Entry_Interval	0x1040	W	0x00000000	This register is used to control the AXI LPI entry interval

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

30.4.2 Detail Registers Description

GMAC_MAC_Configuration

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31	RW	0x0	ARPEN ARP Offload Enable When this bit is set, the MAC can recognize an incoming ARP request packet and schedules the ARP packet for transmission. It forwards the ARP packet to the application and also indicate the events in the RxStatus. When this bit is reset, the MAC receiver does not recognize any ARP packet and indicates them as Type frame in the RxStatus. This bit is available only when the Enable IPv4 ARP Offload is selected. Values: 1'b0: ARP Offload is disabled 1'b1: ARP Offload is enabled
30:28	RO	0x0	reserved
27	RW	0x0	IPC Checksum Offload When set, this bit enables the IPv4 header checksum checking and IPv4 or IPv6 TCP, UDP, or ICMP payload checksum checking. When this bit is reset, the COE function in the receiver is disabled.The Layer 3 and Layer 4 Packet Filter and Enable Split Header features automatically selects the IPC Full Checksum Offload Engine on the Receive side. When any of these features are enabled, you must set the IPC bit. Values: 1'b0: IP header/payload checksum checking is disabled 1'b1: IP header/payload checksum checking is enabled
26:24	RW	0x0	IPG Inter-Packet Gap These bits control the minimum IPG between packets during transmission. This range of minimum IPG is valid in full-duplex mode.In the half-duplex mode, the minimum IPG can be configured only for 64-bit times (IPG = 100). Lower values are not considered. When a JAM pattern is being transmitted because of backpressure activation, the MAC does not consider the minimum IPG.The above function (IPG less than 96 bit times) is valid only when EIPGEN bit in MAC_Ext_Configuration register is reset. When EIPGEN is set, then the minimum IPG (greater than 96 bit times) is controlled as per the description given in EIPG field in MAC_Ext_Configuration register. Values: 3'b000: 96 bit times IPG 3'b001: 88 bit times IPG 3'b010: 80 bit times IPG 3'b011: 72 bit times IPG 3'b100: 64 bit times IPG 3'b101: 56 bit times IPG 3'b110: 48 bit times IPG 3'b111: 40 bit times IPG

Bit	Attr	Reset Value	Description
23	RW	0x0	GPSLCE Giant Packet Size Limit Control Enable When this bit is set, the MAC considers the value in GPSL field in MAC_Ext_Configuration register to declare a received packet as Giant packet. This field must be programmed to more than 1,518 bytes. Otherwise, the MAC considers 1,518 bytes as giant packet limit. When this bit is reset, the MAC considers a received packet as Giant packet when its size is greater than 1,518 bytes (1522 bytes for tagged packet).The watchdog timeout limit, Jumbo Packet Enable and 2K Packet Enable have higher precedence over this bit, that is the MAC considers a received packet as Giant packet when its size is greater than 9,018 bytes (9,022 bytes for tagged packet) with Jumbo Packet Enabled and greater than 2,000 bytes with 2K Packet Enabled. The watchdog timeout, if enabled, terminates the received packet when watchdog limit is reached. Therefore, the programmed giant packet limit should be less than the watchdog limit to get the giant packet status. Values: 1'b0: Giant Packet Size Limit Control is disabled 1'b1: Giant Packet Size Limit Control is enabled
22	RW	0x0	S2KP IEEE 802.3as Support for 2K Packets When this bit is set, the MAC considers all packets with up to 2,000 bytes length as normal packets. When the JE bit is not set, the MAC considers all received packets of size more than 2K bytes as Giant packets.When this bit is reset and the JE bit is not set, the MAC considers all received packets of size more than 1,518 bytes (1,522 bytes for tagged) as giant packets. Note: When the JE bit is set, setting this bit has no effect on the giant packet status. Values: 1'b0: Support upto 2K packet is disabled 1'b1: Support upto 2K packet is Enabled
21	RW	0x0	CST CRC stripping for Type packets When this bit is set, the last four bytes (FCS) of all packets of Ether type (type field greater than 1,536) are stripped and dropped before forwarding the packet to the application. Note: For information about how the settings of the ACS bit and this bit impact the packet length, see the Table, Packet Length based on the CST and ACS Bits. Values: 1'b0: CRC stripping for Type packets is disabled 1'b1: CRC stripping for Type packets is enabled

Bit	Attr	Reset Value	Description
20	RW	0x0	ACS Automatic Pad or CRC Stripping When this bit is set, the MAC strips the Pad or FCS field on the incoming packets only if the value of the length field is less than 1,536 bytes. All received packets with length field greater than or equal to 1,536 bytes are passed to the application without stripping the Pad or FCS field.When this bit is reset, the MAC passes all incoming packets to the application, without any modification. Note: For information about how the settings of CST bit and this bit impact the packet length, see the Table, Packet Length based on the CST and ACS Bit . Values: 1'b0: Automatic Pad or CRC Stripping is disabled 1'b1: Automatic Pad or CRC Stripping is enabled
19	RW	0x0	WD Watchdog Disable When this bit is set, the MAC disables the watchdog timer on the receiver. The MAC can receive packets of up to 16,383 bytes.When this bit is reset, the MAC does not allow more than 2,048 bytes (10,240 if JE is set high) of the packet being received. The MAC cuts off any bytes received after 2,048 bytes. Values: 1'b0: Watchdog is enabled 1'b1: Watchdog is disabled
18	RW	0x0	BE Packet Burst Enable When this bit is set, the MAC allows packet bursting during transmission in the GMII half-duplex mode. Values: 1'b0: Packet Burst is disabled 1'b1: Packet Burst is enabled
17	RW	0x0	JD Jabber Disable When this bit is set, the MAC disables the jabber timer on the transmitter. The MAC can transfer packets of up to 16,383 bytes. When this bit is reset, if the application sends more than 2,048 bytes of data (10,240 if JE is set high) during transmission, the MAC does not send rest of the bytes in that packet. Values: 1'b0: Jabber is enabled 1'b1: Jabber is disabled
16	RW	0x0	JE Jumbo Packet Enable When this bit is set, the MAC allows jumbo packets of 9,018 bytes (9,022 bytes for VLAN tagged packets) without reporting a giant packet error in the Rx packet status. Values: 1'b0: Jumbo packet is disabled 1'b1: Jumbo packet is enabled

Bit	Attr	Reset Value	Description
15	RW	0x0	PS Port Select This bit selects the Ethernet line speed. This bit, along with Bit 14, selects the exact line speed. In the 10/100 Mbps-only (always 1) or 1000 Mbps-only (always 0) configurations, this bit is read-only (RO) with appropriate value. In default 10/100/1000 Mbps configurations, this bit is read-write (R/W). The mac_speed_o[1] signal reflects the value of this bit. Values: 1'b0: For 1000 or 2500 Mbps operations 1'b1: For 10 or 100 Mbps operations
14	RW	0x0	FES Speed This bit selects the speed mode. The mac_speed_o[0] signal reflects the value of this bit. Values: 1'b0: 10 Mbps when PS bit is 1 and 1 Gbps when PS bit is 0 1'b1: 100 Mbps when PS bit is 1 and 2.5 Gbps when PS bit is 0
13	RW	0x0	DM Duplex Mode When this bit is set, the MAC operates in the full-duplex mode in which it can transmit and receive simultaneously. This bit is RO with default value of 1'b1 in the full-duplex-only configurations. Values: 1'b0: Half-duplex mode 1'b1: Full-duplex mode
12	RW	0x0	LM Loopback Mode When this bit is set, the MAC operates in the loopback mode at GMII or MII. The (G)MII Rx clock input (clk_rx_i) is required for the loopback to work properly. This is because the Tx clock is not internally looped back. Values: 1'b0: Loopback is disabled 1'b1: Loopback is enabled
11	RW	0x0	ECRSFD Enable Carrier Sense Before Transmission in Full-Duplex Mode When this bit is set, the MAC transmitter checks the CRS signal before packet transmission in the full-duplex mode. The MAC starts the transmission only when the CRS signal is low.When this bit is reset, the MAC transmitter ignores the status of the CRS signal. Values: 1'b0: ECRSFD is disabled 1'b1: ECRSFD is enabled
10	RW	0x0	DO Disable Receive Own When this bit is set, the MAC disables the reception of packets when the gmii_txen_o is asserted in the half-duplex mode. When this bit is reset, the MAC receives all packets given by the PHY.This bit is not applicable in the full-duplex mode. Values: 1'b0: Enable Receive Own 1'b1: Disable Receive Own

Bit	Attr	Reset Value	Description
9	RW	0x0	DCRS Disable Carrier Sense During Transmission When this bit is set, the MAC transmitter ignores the (G)MII CRS signal during packet transmission in the half-duplex mode. As a result, no errors are generated because of Loss of Carrier or No Carrier during transmission. When this bit is reset, the MAC transmitter generates errors because of Carrier Sense. The MAC can even abort the transmission. Values: 1'b0: Enable Carrier Sense During Transmission 1'b1: Disable Carrier Sense During Transmission
8	RW	0x0	DR Disable Retry When this bit is set, the MAC attempts only one transmission. When a collision occurs on the GMII or MII interface, the MAC ignores the current packet transmission and reports a Packet Abort with excessive collision error in the Tx packet status.When this bit is reset, the MAC retries based on the settings of the BL field. This bit is applicable only in the half-duplex mode. Values: 1'b0: Enable Retry 1'b1: Disable Retry
7	RO	0x0	reserved
6:5	RW	0x0	BL Back-Off Limit The back-off limit determines the random integer number (r) of slot time delays (4,096 bit times for 1000/2500 Mbps; 512 bit times for 10/100 Mbps) for which the MAC waits before rescheduling a transmission attempt during retries after a collision. n = retransmission attempt. The random integer r takes the value in the range 0 <= r < 2^k This bit is applicable only in the half-duplex mode. Values: 2'b00: k = min(n,10) 2'b01: k = min(n,8) 2'b10: k = min(n,4) 2'b11: k = min(n,1)

Bit	Attr	Reset Value	Description
4	RW	0x0	DC Deferral Check When this bit is set, the deferral check function is enabled in the MAC. The MAC issues a Packet Abort status, along with the excessive deferral error bit set in the Tx packet status, when the Tx state machine is deferred for more than 24,288 bit times in 10 or 100 Mbps mode. If the MAC is configured for 1000/2500 Mbps operation, the threshold for deferral is 155,680 bits times. Deferral begins when the transmitter is ready to transmit, but it is prevented because of an active carrier sense signal (CRS) on GMII or MII. The defer time is not cumulative. For example, if the transmitter defers for 10,000 bit times because the CRS signal is active and the CRS signal becomes inactive, the transmitter transmits and collision happens. Because of collision, the transmitter needs to back off and then defer again after back off completion. In such a scenario, the deferral timer is reset to 0, and it is restarted. When this bit is reset, the deferral check function is disabled and the MAC defers until the CRS signal goes inactive. This bit is applicable only in the half-duplex mode. Values: 1'b0: Deferral check function is disabled 1'b1: Deferral check function is enabled
3:2	RW	0x0	PRELEN Preamble Length for Transmit packets These bits control the number of preamble bytes that are added to the beginning of every Tx packet. The preamble reduction occurs only when the MAC is operating in the full-duplex mode. Values: 2'b00: 7 bytes of preamble 2'b01: 5 bytes of preamble 2'b10: 3 bytes of preamble 2'b11: Reserved
1	RW	0x0	TE Transmitter Enable When this bit is set, the Tx state machine of the MAC is enabled for transmission on the GMII or MII interface. When this bit is reset, the MAC Tx state machine is disabled after it completes the transmission of the current packet. The Tx state machine does not transmit any more packets. Values: 1'b0: Transmitter is disabled 1'b1: Transmitter is enabled
0	RW	0x0	RE Receiver Enable When this bit is set, the Rx state machine of the MAC is enabled for receiving packets from the GMII or MII interface. When this bit is reset, the MAC Rx state machine is disabled after it completes the reception of the current packet. The Rx state machine does not receive any more packets from the GMII or MII interface. Values: 1'b0: Receiver is disabled 1'b1: Receiver is enabled

GMAC_MAC_Ext_Configuration

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:25	RW	0x00	EIPG Extended Inter-Packet Gap The value in this field is applicable when the EIPGEN bit is set. This field (as Most Significant bits), along with IPG field in MAC_Configuration register, gives the minimum IPG greater than 96 bit times in steps of 8 bit times: EIPG, IPG 8'h00 - 104 bit times 8'h01 - 112 bit times 8'h02 - 120 bit times ---------- 8'hFF - 2144 bit times
24	RW	0x0	EIPGEN Extended Inter-Packet Gap Enable When this bit is set, the MAC interprets EIPG field and IPG field in MAC_Configuration register together as minimum IPG greater than 96 bit times in steps of 8 bit times. When this bit is reset, the MAC ignores EIPG field and interprets IPG field in MAC_Configuration register as minimum IPG less than or equal to 96 bit times in steps of 8 bit times. Note: The extended Inter-Packet Gap feature must be enabled when operating in Full-Duplex mode only. There may be undesirable effects on back-pressure function and frame transmission if it is enabled in Half-Duplex mode. Values: 1'b0: Extended Inter-Packet Gap is disabled 1'b1: Extended Inter-Packet Gap is enabled
23:19	RO	0x00	reserved
18	RW	0x0	USP Unicast Slow Protocol Packet Detect When this bit is set, the MAC detects the Slow Protocol packets with unicast address of the station specified in the MAC_Address0_High and MAC_Address0_Low registers. The MAC also detects the Slow Protocol packets with the Slow Protocols multicast address (01-80-C2-00-00-02). When this bit is reset, the MAC detects only Slow Protocol packets with the Slow Protocol multicast address specified in the IEEE 802.3-2015, Section 5. Values: 1'b0: Unicast Slow Protocol Packet Detection is disabled 1'b1: Unicast Slow Protocol Packet Detection is enabled
17	RW	0x0	SPEN Slow Protocol Detection Enable When this bit is set, MAC processes the Slow Protocol packets (Ether Type 0x8809) and provides the Rx status. The MAC discards the Slow Protocol packets with invalid sub-types. When this bit is reset, the MAC forwards all error-free Slow Protocol packets to the application. The MAC considers such packets as normal Type packets. Values: 1'b0: Slow Protocol Detection is disabled 1'b1: Slow Protocol Detection is enabled

Bit	Attr	Reset Value	Description
16	RW	0x0	DCRCC Disable CRC Checking for Received Packets When this bit is set, the MAC receiver does not check the CRC field in the received packets. When this bit is reset, the MAC receiver always checks the CRC field in the received packets. Values: 1'b0: CRC Checking is enabled 1'b1: CRC Checking is disabled
15:14	RO	0x0	reserved
13:0	RW	0x0000	GPSL Giant Packet Size Limit If the received packet size is greater than the value programmed in this field in units of bytes, the MAC declares the received packet as Giant packet. The value programmed in this field must be greater than or equal to 1,518 bytes. Any other programmed value is considered as 1,518 bytes. For VLAN tagged packets, the MAC adds 4 bytes to the programmed value. When the Enable Double VLAN Processing option is selected, the MAC adds 8 bytes to the programmed value for double VLAN tagged packets. The value in this field is applicable when the GPSLCE bit is set in MAC_Configuration register.

GMAC_MAC_Packet_Filter

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:17	RO	0x0000	reserved
16	RW	0x0	VTFE VLAN Tag Filter Enable When this bit is set, the MAC drops the VLAN tagged packets that do not match the VLAN Tag. When this bit is reset, the MAC forwards all packets irrespective of the match status of the VLAN Tag. Values: 1'b0: VLAN Tag Filter is disabled 1'b1: VLAN Tag Filter is enabled
15:11	RO	0x00	reserved
10	RW	0x0	HPF Hash or Perfect Filter When this bit is set, the address filter passes a packet if it matches either the perfect filtering or hash filtering as set by the HMC or HUC bit. When this bit is reset and the HUC or HMC bit is set, the packet is passed only if it matches the Hash filter. Values: 1'b0: Hash or Perfect Filter is disabled 2'b1: Hash or Perfect Filter is enabled
9:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7:6	RW	0x0	PCF Pass Control Packets These bits control the forwarding of all control packets (including unicast and multicast Pause packets). Values: 2'b00: MAC filters all control packets from reaching the application 2'b01: MAC forwards all control packets except Pause packets to the application even if they fail the Address filter 2'b10: MAC forwards all control packets to the application even if they fail the Address filter 2'b11: MAC forwards the control packets that pass the Address filter
5	RW	0x0	DBF Disable Broadcast Packets When this bit is set, the AFM module blocks all incoming broadcast packets. In addition, it overrides all other filter settings. When this bit is reset, the AFM module passes all received broadcast packets. Values: 1'b0: Enable Broadcast Packets 1'b1: Disable Broadcast Packets
4	RW	0x0	PM Pass All Multicast When this bit is set, it indicates that all received packets with a multicast destination address (first bit in the destination address field is '1') are passed. When this bit is reset, filtering of multicast packet depends on HMC bit. Values: 1'b0: Pass All Multicast is disabled 1'b1: Pass All Multicast is enabled
3	RW	0x0	DAIF DA Inverse Filtering When this bit is set, the Address Check block operates in inverse filtering mode for the DA address comparison for both unicast and multicast packets. When this bit is reset, normal filtering of packets is performed. Values: 1'b0: DA Inverse Filtering is disabled 1'b1: DA Inverse Filtering is enabled
2	RW	0x0	HMC Hash Multicast When this bit is set, the MAC performs the destination address filtering of received multicast packets according to the hash table. When this bit is reset, the MAC performs the perfect destination address filtering for multicast packets, that is, it compares the DA field with the values programmed in DA registers. Values: 1'b0: Hash Multicast is disabled 1'b1: Hash Multicast is enabled

Bit	Attr	Reset Value	Description
1	RW	0x0	HUC Hash Unicast When this bit is set, the MAC performs the destination address filtering of unicast packets according to the hash table. When this bit is reset, the MAC performs a perfect destination address filtering for unicast packets, that is, it compares the DA field with the values programmed in DA registers. Values: 1'b0: Hash Unicast is disabled 1'b1: Hash Unicast is enabled
0	RW	0x0	PR Promiscuous Mode When this bit is set, the Address Filtering module passes all incoming packets irrespective of the destination or source address. The SA or DA Filter Fails status bits of the Rx Status Word are always cleared when PR is set. Values: 1'b0: Promiscuous Mode is disabled 1'b1: Promiscuous Mode is enabled

GMAC_MAC_Watchdog_Timeout

Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:9	RO	0x000000	reserved
8	RW	0x0	PWE Programmable Watchdog Enable When this bit is set and the WD bit of the MAC_Configuration register is reset, the WTO field is used as watchdog timeout for a received packet. When this bit is cleared, the watchdog timeout for a received packet is controlled by setting of WD and JE bits in MAC_Configuration register. Values: 1'b0: Programmable Watchdog is disabled 1'b1: Programmable Watchdog is enabled
7:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3:0	RW	0x0	WTO Watchdog Timeout When the PWE bit is set and the WD bit of the MAC_Configuration register is reset, this field is used as watchdog timeout for a received packet. If the length of a received packet exceeds the value of this field, such packet is terminated and declared as an error packet. Note: When the PWE bit is set, the value in this field should be more than 1,522 (0x05F2). Otherwise, the IEEE 802.3-specified valid tagged packets are declared as error packets and then dropped. Values: 4'b0000: 2 KB 4'b0001: 3 KB 4'b0010: 4 KB 4'b0011: 5 KB 4'b0100: 6 KB 4'b0101: 7 KB 4'b0110: 8 KB 4'b0111: 9 KB 4'b1000: 10 KB 4'b1001: 11 KB 4'b1010: 12 KB 4'b1011: 13 KB 4'b1100: 14 KB 4'b1101: 15 KB 4'b1110: 16383 Bytes 4'b1111: Reserved

GMAC_MAC_Hash_Table_Reg0

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	HT31T0 MAC Hash Table First 32 Bits This field contains the first 32 Bits [31:0] of the Hash table. The Hash table is used for group address filtering. For hash filtering, the content of the destination address in the incoming packet is passed through the CRC logic and the upper six (seven in 128-bit Hash or eight in 256-bit Hash) bits of the CRC register are used to index the content of the Hash table. The most significant bits determines the register to be used (Hash Table Register X), and the least significant five bits determine the bit within the register. For example, a hash value of 6'b100000 (in 64-bit Hash) selects Bit 0 of the Hash Table Register 1, a value of 7b'1110000 (in 128-bit Hash) selects Bit 16 of the Hash Table Register 3 and a value of 8b'10111111 (in 256-bit Hash) selects Bit 31 of the Hash Table Register 5. The hash value of the destination address is calculated in the following way: 1. Calculate the 32-bit CRC for the DA (See IEEE 802.3, Section 3.2.8 for the steps to calculate CRC32). 2. Perform bitwise reversal for the value obtained in Step 1. 3. Take the upper 6 (or 7 or 8) bits from the value obtained in Step 2. If the corresponding bit value of the register is 1'b1, the packet is accepted. Otherwise, it is rejected. If the PM bit is set in MAC_Packet_Filter, all multicast packets are accepted regardless of the multicast hash values. If the Hash Table register is configured to be double-synchronized to the (G)MII clock domain, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the Hash Table Register X registers are written.If double-synchronization is enabled, consecutive writes to this register should be performed after at least four clock cycles in the destination clock domain.

GMAC_MAC_Hash_Table_Reg1

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	HT63T32 MAC Hash Table Second 32 Bits This field contains the second 32 Bits [63:32] of the Hash table.

GMAC_MAC_VLAN_Tag

Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:25	RO	0x00	reserved
24	RW	0x0	EVLRXS Enable VLAN Tag in Rx status When this bit is set, MAC provides the outer VLAN Tag in the Rx status. When this bit is reset, the MAC does not provide the outer VLAN Tag in Rx status. Values: 1'b0: VLAN Tag in Rx status is disabled 1'b1: VLAN Tag in Rx status is enabled
23	RO	0x0	reserved

Bit	Attr	Reset Value	Description
22:21	RW	0x0	EVLS Enable VLAN Tag Stripping on Receive This field indicates the stripping operation on the outer VLAN Tag in received packet. Values: 2'b00: Do not strip 2'b01: Strip if VLAN filter passes 2'b10: Strip if VLAN filter fails 2'b11: Always strip
20	RW	0x0	DOVLTC Disable VLAN Type Check When this bit is set, the MAC does not check whether the VLAN Tag specified by the ERIVLT bit is of type S-VLAN or C-VLAN. When this bit is reset, the MAC filters or matches the VLAN Tag specified by the ERIVLT bit only when VLAN Tag type is similar to the one specified by the ERSVLM bit. Values: 1'b0: VLAN Type Check is enabled 1'b1: VLAN Type Check is disabled
19	RW	0x0	ERSVLM Enable Receive S-VLAN Match When this bit is set, the MAC receiver enables filtering or matching for S-VLAN (Type = 0x88A8) packets. When this bit is reset, the MAC receiver enables filtering or matching for C-VLAN (Type = 0x8100) packets. The ERIVLT bit determines the VLAN tag position considered for filtering or matching. Values: 1'b0: Receive S-VLAN Match is disabled 1'b1: Receive S-VLAN Match is enabled
18	RW	0x0	ESVL Enable S-VLAN When this bit is set, the MAC transmitter and receiver consider the S-VLAN packets (Type = 0x88A8) as valid VLAN tagged packets. Values: 1'b0: S-VLAN is disabled 1'b1: S-VLAN is enabled
17	RW	0x0	VTIM VLAN Tag Inverse Match Enable When this bit is set, this bit enables the VLAN Tag inverse matching. The packets without matching VLAN Tag are marked as matched. When reset, this bit enables the VLAN Tag perfect matching. The packets with matched VLAN Tag are marked as matched. Values: 1'b0: VLAN Tag Inverse Match is disabled 1'b1: VLAN Tag Inverse Match is enabled

Bit	Attr	Reset Value	Description
16	RW	0x0	ETV Enable 12-Bit VLAN Tag Comparison When this bit is set, a 12-bit VLAN identifier is used for comparing and filtering instead of the complete 16-bit VLAN tag. Bits[11:0] of VLAN tag are compared with the corresponding field in the received VLAN-tagged packet. Similarly, when enabled, only 12 bits of the VLAN tag in the received packet are used for hash-based VLAN filtering. When this bit is reset, all 16 bits of the 15th and 16th bytes of the received VLAN packet are used for comparison and VLAN hash filtering. Values: 1'b0: 12-Bit VLAN Tag Comparison is disabled 1'b1: 12-Bit VLAN Tag Comparison is enabled
15:0	RW	0x0000	VL VLAN Tag Identifier for Receive Packets This field contains the 802.1Q VLAN tag to identify the VLAN packets. This VLAN tag identifier is compared to the 15th and 16th bytes of the packets being received for VLAN packets. The following list describes the bits of this field: 1. Bits[15:13]: User Priority 2. Bit 12: Canonical Format Indicator (CFI) or Drop Eligible Indicator (DEI) 3. Bits[11:0]: VLAN Identifier (VID) field of VLAN tag When the ETV bit is set, only the VID is used for comparison. If this field ([11:0] if ETV is set) is all zeros, the MAC does not check the 15th and 16th bytes for VLAN tag comparison and declares all packets with Type field value of 0x8100 or 0x88a8 as VLAN packets.

GMAC_MAC_Q0_Tx_Flow_Ctrl

Address: Operational Base + offset (0x0070)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	PT Pause Time This field holds the value to be used in the Pause Time field in the Tx control packet. If the Pause Time bits are configured to be double-synchronized to the (G)MII clock domain, consecutive writes to this register should be performed only after at least four clock cycles in the destination clock domain.
15:8	RO	0x00	reserved
7	RW	0x0	DZPQ Disable Zero-Quanta Pause When this bit is set, it disables the automatic generation of the zero-quanta Pause packets on de-assertion of the flow-control signal from the FIFO layer (MTL or external sideband flow control signal sbd_flowctrl_i or mti_flowctrl_i). When this bit is reset, normal operation with automatic zero-quanta Pause packet generation is enabled. Values: 1'b0: Zero-Quanta Pause packet generation is enabled 1'b1: Zero-Quanta Pause packet generation is disabled

Bit	Attr	Reset Value	Description
6:4	RW	0x0	PLT Pause Low Threshold This field configures the threshold of the Pause timer at which the input flow control signal mti_flowctrl_i (or sbd_flowctrl_i) is checked for automatic retransmission of the Pause packet. The threshold values should be always less than the Pause Time configured in Bits[31:16]. For example, if PT = 100H (256 slot times), and PLT = 001, a second Pause packet is automatically transmitted if the mti_flowctrl_i signal is asserted at 228 (256-28) slot times after the first Pause packet is transmitted. The following list provides the threshold values for different values. The slot time is defined as the time taken to transmit 512 bits (64 bytes) on the GMII or MII interface. This (approximate) computation is based on the packet size (64, 1518, 2000, 9018, 16384, or 32768) + 2 Pause Packet Size + IPG in Slot Times. Values: 3'b000: Pause Time minus 4 Slot Times (PT -4 slot times) 3'b001: Pause Time minus 28 Slot Times (PT -28 slot times) 3'b010: Pause Time minus 36 Slot Times (PT -36 slot times) 3'b011: Pause Time minus 144 Slot Times (PT -144 slot times) 3'b100: Pause Time minus 256 Slot Times (PT -256 slot times) 3'b101: Pause Time minus 512 Slot Times (PT -512 slot times) 3'b110: Reserved
3:2	RO	0x0	reserved
1	RW	0x0	TFE Transmit Flow Control Enable Full-Duplex Mode: In the full-duplex mode, when this bit is set, the MAC enables the flow control operation to Tx Pause packets. When this bit is reset, the flow control operation in the MAC is disabled, and the MAC does not transmit any Pause packets. Half-Duplex Mode: In the half-duplex mode, when this bit is set, the MAC enables the backpressure operation. When this bit is reset, the backpressure feature is disabled. Values: 1'b0: Transmit Flow Control is disabled 1'b1: Transmit Flow Control is enabled

Bit	Attr	Reset Value	Description
0	RW	0x0	FCB_BPA Flow Control Busy or Backpressure Activate This bit initiates a Pause packet in the full-duplex mode and activates the backpressure function in the half-duplex mode if the TFE bit is set. Full-Duplex Mode: In the full-duplex mode, this bit should be read as 1'b0 before writing to this register. To initiate a Pause packet, the application must set this bit to 1'b1. During Control packet transfer, this bit continues to be set to indicate that a packet transmission is in progress. When Pause packet transmission is complete, the MAC resets this bit to 1'b0. You should not write to this register until this bit is cleared. Half-Duplex Mode: When this bit is set (and TFE bit is set) in the half-duplex mode, the MAC asserts the backpressure. During backpressure, when the MAC receives a new packet, the transmitter starts sending a JAM pattern resulting in a collision. This control register bit is logically ORed with the mti_flowctrl_i input signal for the backpressure function. When the MAC is configured for the full-duplex mode, the BPA is automatically disabled. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. Values: 1'b0: Flow Control Busy or Backpressure Activate is disabled 1'b1: Flow Control Busy or Backpressure Activate is enabled

GMAC_MAC_Rx_Flow_Ctrl

Address: Operational Base + offset (0x0090)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved
1	RW	0x0	UP Unicast Pause Packet Detect A pause packet is processed when it has the unique multicast address specified in the IEEE 802.3. When this bit is set, the MAC can also detect Pause packets with unicast address of the station. This unicast address should be as specified in MAC_Address0_High and MAC_Address0_Low. When this bit is reset, the MAC only detects Pause packets with unique multicast address. Note: The MAC does not process a Pause packet if the multicast address is different from the unique multicast address. This is also applicable to the received PFC packet when the Priority Flow Control (PFC) is enabled. The unique multicast address (0x01_80_C2_00_00_01) is as specified in IEEE 802.1 Qbb-2011. Values: 1'b0: Unicast Pause Packet Detect disabled 1'b1: Unicast Pause Packet Detect enabled

Bit	Attr	Reset Value	Description
0	RW	0x0	RFE Receive Flow Control Enable When this bit is set and the MAC is operating in full-duplex mode, the MAC decodes the received Pause packet and disables its transmitter for a specified (Pause) time. When this bit is reset or the MAC is operating in half-duplex mode, the decode function of the Pause packet is disabled. When PFC is enabled, flow control is enabled for PFC packets. The MAC decodes the received PFC packet and disables the Transmit queue, with matching priorities, for a duration of received Pause time. Values: 1'b0: Receive Flow Control is disabled 1'b1: Receive Flow Control is enabled

GMAC_MAC_Interrupt_Status

Address: Operational Base + offset (0x00b0)

Bit	Attr	Reset Value	Description
31:19	RO	0x0000	reserved
18	RO	0x0	MDIOIS MDIO Interrupt Status This bit indicates an interrupt event after the completion of MDIO operation. To reset this bit, the application has to read this bit/Write 1 to this bit when RCWE bit of MAC_CSR_SW_Ctrl register is set. Access restriction applies. Clears on read (or write of 1 when RCWE bit in MAC_CSR_SW_Ctrl register is set). Self-set to 1 on internal event. Values: 1'b0: MDIO Interrupt status not active 1'b1: MDIO Interrupt status active
17:15	RO	0x0	reserved
14	RO	0x0	RXSTSIS Receive Status Interrupt This bit indicates the status of received packets. This bit is set when the RWT bit is set in the MAC_Rx_Tx_Status register. This bit is cleared when the corresponding interrupt source bit is read (or corresponding interrupt source bit is written to 1 when RCWE bit of MAC_CSR_SW_Ctrl register is set) in the MAC_Rx_Tx_Status register. Values: 1'b0: Receive Interrupt status not active 1'b1: Receive Interrupt status active

Bit	Attr	Reset Value	Description
13	RO	0x0	TXSTSIS Transmit Status Interrupt This bit indicates the status of transmitted packets. This bit is set when any of the following bits is set in the MAC_Rx_Tx_Status register: 1. Excessive Collision (EXCOL) 2. Late Collision (LCOL) 3. Excessive Deferral (EXDEF) 4. Loss of Carrier (LCARR) 5. No Carrier (NCARR) 6. Jabber Timeout (TJT) This bit is cleared when the corresponding interrupt source bit is read (or corresponding interrupt source bit is written to 1 when RCWE bit of MAC_CSR_SW_Ctrl register is set) in the MAC_Rx_Tx_Status register. Values: 1'b0: Transmit Interrupt status not active 1'b1: Transmit Interrupt status active
12	RO	0x0	TSIS Timestamp Interrupt Status If the Timestamp feature is enabled, this bit is set when any of the following conditions is true: 1. The system time value is equal to or exceeds the value specified in the Target Time High and Low registers. 2. There is an overflow in the Seconds register. 3. The Target Time Error occurred, that is, programmed target time already elapsed. If the Auxiliary Snapshot feature is enabled, this bit is set when the auxiliary snapshot trigger is asserted. In configurations other than EQOS_CORE, when drop transmit status is enabled in MTL, this bit is set when the captured transmit timestamp is updated in the MAC_Tx_Timestamp_Status_Nanoseconds and Mac_TxTimestamp_Status_Seconds registers. When PTP offload feature is enabled, this bit is set when the captured transmit timestamp is updated in the MAC_Tx_Timestamp_Status_Nanoseconds and MAC_Tx_Timestamp_Status_Seconds registers, for PTO generated Delay Request and Pdelay request packets. This bit is cleared when the corresponding interrupt source bit is read (or corresponding interrupt source bit is written to 1 when RCWE bit of MAC_CSR_SW_Ctrl register is set) in the MAC_Timestamp_Status register. Values: 1'b0: Timestamp Interrupt status not active 1'b1: Timestamp Interrupt status active
11	RO	0x0	MMCRXIPIS MMC Receive Checksum Offload Interrupt Status This bit is set high when an interrupt is generated in the MMC Receive Checksum Offload Interrupt Register. This bit is cleared when all bits in this interrupt register are cleared. This bit is valid only when you select the Enable MAC Management Counters (MMC) and Enable Receive TCP/IP Checksum Check options. Values: 1'b0: MMC Receive Checksum Offload Interrupt status not active 1'b1: MMC Receive Checksum Offload Interrupt status active

Bit	Attr	Reset Value	Description
10	RO	0x0	MMCTXIS MMC Transmit Interrupt Status This bit is set high when an interrupt is generated in the MMC Transmit Interrupt Register. This bit is cleared when all bits in this interrupt register are cleared. This bit is valid only when you select the Enable MAC Management Counters (MMC) option. Values: 1'b0: MMC Transmit Interrupt status not active 1'b1: MMC Transmit Interrupt status active
9	RO	0x0	MMCRXIS MMC Receive Interrupt Status This bit is set high when an interrupt is generated in the MMC Receive Interrupt Register. This bit is cleared when all bits in this interrupt register are cleared. This bit is valid only when you select the Enable MAC Management Counters (MMC) option. Values: 1'b0: MMC Receive Interrupt status not active 1'b1: MMC Receive Interrupt status active
8	RO	0x0	MMCIS MMC Interrupt Status This bit is set high when Bit 11, Bit 10, or Bit 9 is set high. This bit is cleared only when all these bits are low. This bit is valid only when you select the Enable MAC Management Counters (MMC) option. Values: 1'b0: MMC Interrupt status not active 1'b1: MMC Interrupt status active
7:6	RO	0x0	reserved
5	RO	0x0	LPIIS LPI Interrupt Status When the Energy Efficient Ethernet feature is enabled, this bit is set for any LPI state entry or exit in the MAC Transmitter or Receiver. This bit is cleared when the corresponding interrupt source bit of MAC_LPI_Control_Status register is read (or corresponding interrupt source bit of MAC_LPI_Control_Status register is written to 1 when RCWE bit of MAC_CSR_SW_Ctrl register is set). Values: 1'b0: LPI Interrupt status not active 1'b1: LPI Interrupt status active
4	RO	0x0	PMTIS PMT Interrupt Status This bit is set when a Magic packet or Wake-on-LAN packet is received in the power-down mode (RWKPRCVD and MGKPRCVD bits in MAC_PMT_Control_Status register). This bit is cleared when corresponding interrupt source bit are cleared because of a Read operation to the MAC_PMT_Control_Status register (or corresponding interrupt source bit of MAC_PMT_Control_Status register is written to 1 when RCWE bit of MAC_CSR_SW_Ctrl register is set). This bit is valid only when you select the Enable Power Management option. Values: 1'b0: PMT Interrupt status not active 1'b1: PMT Interrupt status active

Bit	Attr	Reset Value	Description
3	RO	0x0	PHYIS PHY Interrupt This bit is set when rising edge is detected on the phy_intr_i input. This bit is cleared when this register is read (or this bit is written to 1 when RCWE bit of MAC_CSR_SW_Ctrl register is set). Values: 1'b0: PHY Interrupt not detected 1'b1: PHY Interrupt detected
2:1	RO	0x0	reserved
0	RO	0x0	RGSMIIIS RGMII or SMII Interrupt Status This bit is set because of any change in value of the Link Status of RGMII or SMII interface (LNKSTS bit in MAC_PHYIF_Control_Status register). This bit is cleared when the MAC_PHYIF_Control_Status register is read (or LNKSTS bit of MAC_PHYIF_Control_Status register is written to 1 when RCWE bit of MAC_CSR_SW_Ctrl register is set). This bit is valid only when you select the optional RGMII or SMII PHY interface. Values: 1'b0: RGMII or SMII Interrupt Status is not active 1'b1: RGMII or SMII Interrupt Status is active

GMAC_MAC_Interrupt_Enable

Address: Operational Base + offset (0x00b4)

Bit	Attr	Reset Value	Description
31:19	RO	0x0000	reserved
18	RW	0x0	MDIOIE MDIO Interrupt Enable When this bit is set, it enables the assertion of the interrupt when MDIOIS field is set in the MAC_Interrupt_Status register. Values: 1'b0: MDIO Interrupt is disabled 1'b1: MDIO Interrupt is enabled
17:15	RO	0x0	reserved
14	RW	0x0	RXSTSIE Receive Status Interrupt Enable When this bit is set, it enables the assertion of the interrupt signal because of the setting of RXSTSIS bit in the MAC_Interrupt_Status register. Values: 1'b0: Receive Status Interrupt is disabled 1'b1: Receive Status Interrupt is enabled
13	RW	0x0	TXSTSIE Transmit Status Interrupt Enable When this bit is set, it enables the assertion of the interrupt signal because of the setting of TXSTSIS bit in the MAC_Interrupt_Status register. Values: 1'b0: Timestamp Status Interrupt is disabled 1'b1: Timestamp Status Interrupt is enabled

Bit	Attr	Reset Value	Description
12	RW	0x0	TSIE Timestamp Interrupt Enable When this bit is set, it enables the assertion of the interrupt signal because of the setting of TSIS bit in MAC_Interrupt_Status register. Values: 1'b0: Timestamp Interrupt is disabled 1'b1: Timestamp Interrupt is enabled
11:6	RO	0x00	reserved
5	RW	0x0	LPIIE LPI Interrupt Enable When this bit is set, it enables the assertion of the interrupt signal because of the setting of LPIIS bit in MAC_Interrupt_Status register. Values: 1'b0: LPI Interrupt is disabled 1'b1: LPI Interrupt is enabled
4	RW	0x0	PMTIE PMT Interrupt Enable When this bit is set, it enables the assertion of the interrupt signal because of the setting of PMTIS bit in MAC_Interrupt_Status register. Values: 1'b0: PMT Interrupt is disabled 1'b1: PMT Interrupt is enabled
3	RW	0x0	PHYIE PHY Interrupt Enable When this bit is set, it enables the assertion of the interrupt signal because of the setting of PHYIS bit in MAC_Interrupt_Status register. Values: 1'b0: PHY Interrupt is disabled 1'b1: PHY Interrupt is enabled
2:1	RO	0x0	reserved
0	RW	0x0	RGSMIIIE RGMII or SMII Interrupt Enable When this bit is set, it enables the assertion of the interrupt signal because of the setting of RGSMIIIS bit in MAC_Interrupt_Status register. Values: 1'b0: RGMII or SMII Interrupt is disabled 1'b1: RGMII or SMII Interrupt is enabled

GMAC_MAC_Rx_Tx_Status

Address: Operational Base + offset (0x00b8)

Bit	Attr	Reset Value	Description
31:9	RO	0x000000	reserved

Bit	Attr	Reset Value	Description
8	RO	0x0	RWT Receive Watchdog Timeout This bit is set when a packet with length greater than 2,048 bytes is received (10, 240 bytes when Jumbo Packet mode is enabled) and the WD bit is reset in the MAC_Configuration register. This bit is set when a packet with length greater than 16,383 bytes is received and the WD bit is set in the MAC_Configuration register. Access restriction applies. Clears on read (or write of 1 when RCWE bit in MAC_CSR_SW_Ctrl register is set). Self-set to 1 on internal event. Values: 1'b0: No receive watchdog timeout 1'b1: Receive watchdog timed out
7:6	RO	0x0	reserved
5	RO	0x0	EXCOL Excessive Collisions When the DTXSTS bit is set in the MTL_Operation_Mode register, this bit indicates that the transmission aborted after 16 successive collisions while attempting to transmit the current packet. If the DR bit is set in the MAC_Configuration register, this bit is set after the first collision and the packet transmission is aborted. Access restriction applies. Clears on read (or write of 1 when RCWE bit in MAC_CSR_SW_Ctrl register is set). Self-set to 1 on internal event. Values: 1'b0: No collision 1'b1: Excessive collision is sensed
4	RO	0x0	LCOL Late Collision When the DTXSTS bit is set in the MTL_Operation_Mode register, this bit indicates that the packet transmission aborted because a collision occurred after the collision window (64 bytes including Preamble in MII mode; 512 bytes including Preamble and Carrier Extension in GMII mode). This bit is not valid if the Underflow error occurs. Access restriction applies. Clears on read (or write of 1 when RCWE bit in MAC_CSR_SW_Ctrl register is set). Self-set to 1 on internal event. Values: 1'b0: No collision 1'b1: Late collision is sensed
3	RO	0x0	EXDEF Excessive Deferral When the DTXSTS bit is set in the MTL_Operation_Mode register and the DC bit is set in the MAC_Configuration register, this bit indicates that the transmission ended because of excessive deferral of over 24,288 bit times (155,680 in 1000/2500 Mbps mode or when Jumbo packet is enabled). Access restriction applies. Clears on read (or write of 1 when RCWE bit in MAC_CSR_SW_Ctrl register is set). Self-set to 1 on internal event. Values: 1'b0: No Excessive deferral 1'b1: Excessive deferral

Bit	Attr	Reset Value	Description
2	RO	0x0	LCARR Loss of Carrier When the DTXSTS bit is set in the MTL_Operation_Mode register, this bit indicates that the loss of carrier occurred during packet transmission, that is, the phy_crs_i signal was inactive for one or more transmission clock periods during packet transmission. This bit is valid only for packets transmitted without collision. Access restriction applies. Clears on read (or write of 1 when RCWE bit in MAC_CSR_SW_Ctrl register is set). Self-set to 1 on internal event. Values: 1'b0: Carrier is present 1'b1: Loss of carrier
1	RO	0x0	NCARR No Carrier When the DTXSTS bit is set in the MTL_Operation_Mode register, this bit indicates that the carrier signal from the PHY is not present at the end of preamble transmission. Access restriction applies. Clears on read (or write of 1 when RCWE bit in MAC_CSR_SW_Ctrl register is set). Self-set to 1 on internal event. Values: 1'b0: Carrier is present 1'b1: No carrier
0	RO	0x0	TJT Transmit Jabber Timeout This bit indicates that the Transmit Jabber Timer expired which happens when the packet size exceeds 2,048 bytes (10,240 bytes when the Jumbo packet is enabled) and JD bit is reset in the MAC_Configuration register. This bit is set when the packet size exceeds 16,383 bytes and the JD bit is set in the MAC_Configuration register. Access restriction applies. Clears on read (or write of 1 when RCWE bit in MAC_CSR_SW_Ctrl register is set). Self-set to 1 on internal event. Values: 1'b0: No Transmit Jabber Timeout 1'b1: Transmit Jabber Timeout occurr

GMAC_MAC_PMT_Control_Status

Address: Operational Base + offset (0x00c0)

Bit	Attr	Reset Value	Description
31	RW	0x0	RWKFILTRST Remote Wake-Up Packet Filter Register Pointer Reset When this bit is set, the remote wake-up packet filter register pointer is reset to 3'b000. It is automatically cleared after 1 clock cycle. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. Values: 1'b0: Remote Wake-Up Packet Filter Register Pointer is not Reset 1'b1: Remote Wake-Up Packet Filter Register Pointer is Reset
30:29	RO	0x0	reserved

Bit	Attr	Reset Value	Description
28:24	RO	0x00	RWKPTR Remote Wake-up FIFO Pointer This field gives the current value (0 to 7, 15, or 31 when 4, 8, or 16 Remote Wake-up Packet Filters are selected) of the Remote Wake-up Packet Filter register pointer. When the value of this pointer is equal to maximum for the selected number of Remote Wake-up Packet Filters, the contents of the Remote Wake-up Packet Filter Register are transferred to the clk_rx_i domain when a Write occurs to that register.
23:11	RO	0x0000	reserved
10	RW	0x0	RWKPFE Remote Wake-up Packet Forwarding Enable When this bit is set along with RWKPKTEN, the MAC receiver drops all received frames until it receives the expected Wake-up frame. All frames after that event including the received wake-up frame are forwarded to application. This bit is then self-cleared on receiving the wake-up packet. The application can also clear this bit before the expected wake-up frame is received. In such cases, the MAC reverts to the default behavior where packets received are forwarded to the application. This bit must only be set when RWKPKTEN is set high and PWRDWN is set low. The setting of this bit has no effect when PWRDWN is set high. Note: If Magic Packet Enable and Wake-Up Frame Enable are both set along with setting of this bit and Magic Packet is received prior to wake-up frame, this bit is self-cleared on receiving Magic Packet, the received Magic packet is dropped, and all frames after received Magic Packet are forwarded to application. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. Values: 1'b0: Remote Wake-up Packet Forwarding is disabled 1'b1: Remote Wake-up Packet Forwarding is enabled
9	RW	0x0	GLBLUCAST Global Unicast When this bit set, any unicast packet filtered by the MAC (DAF) address recognition is detected as a remote wake-up packet. Values: 1'b0: Global unicast is disabled 1'b1: Global unicast is enabled
8:7	RO	0x0	reserved
6	RO	0x0	RWKPRCVD Remote Wake-Up Packet Received When this bit is set, it indicates that the power management event is generated because of the reception of a remote wake-up packet. This bit is cleared when this register is read. Access restriction applies. Clears on read (or write of 1 when RCWE bit in MAC_CSR_SW_Ctrl register is set). Self-set to 1 on internal event. Values: 1'b0: Remote wake-up packet is received 1'b1: Remote wake-up packet is received

Bit	Attr	Reset Value	Description
5	RO	0x0	MGKPRCVD Magic Packet Received When this bit is set, it indicates that the power management event is generated because of the reception of a magic packet. This bit is cleared when this register is read. Access restriction applies. Clears on read (or write of 1 when RCWE bit in MAC_CSR_SW_Ctrl register is set). Self-set to 1 on internal event. Values: 1'b0: No Magic packet is received 1'b1: Magic packet is received
4:3	RO	0x0	reserved
2	RW	0x0	RWKPKTEN Remote Wake-Up Packet Enable When this bit is set, a power management event is generated when the MAC receives a remote wake-up packet. Values: 1'b0: Remote wake-up packet is disabled 1'b1: Remote wake-up packet is enabled
1	RW	0x0	MGKPKTEN Magic Packet Enable When this bit is set, a power management event is generated when the MAC receives a magic packet. Values: 1'b0: Magic Packet is disabled 1'b1: Magic Packet is enabled
0	RW	0x0	PWRDWN Power Down When this bit is set, the MAC receiver drops all received packets until it receives the expected magic packet or remote wake-up packet. This bit is then self-cleared and the power-down mode is disabled. The software can clear this bit before the expected magic packet or remote wake-up packet is received. The packets received by the MAC after this bit is cleared are forwarded to the application. This bit must only be set when the Magic Packet Enable, Global Unicast, or Remote Wake-Up Packet Enable bit is set high. Note: You can gate-off the CSR clock during the power-down mode. However, when the CSR clock is gated-off, you cannot perform any read or write operations on this register. Therefore, the Software cannot clear this bit. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. Values: 1'b0: Power down is disabled 1'b1: Power down is enabled

GMAC_MAC_RWK_Packet_Filter

Address: Operational Base + offset (0x00c4)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	WKUPFRMFTR(cont.) (4) If filters chained by And_Previous bit setting have complementary programming, then a frame may never pass the AND chained filter. For example, if Filter 2 And_Previous bit is set (bit 1 in Filter 2 command is set), Filter 1 Address_Type bit is set (bit 3 in Filter 1 command is set) indicating multicast detection and Filter 2 Address_Type bit is reset (bit 3 in Filter 2 command is reset) indicating unicast detection or vice versa, a remote wakeup frame does not pass the AND chained filter as a remote wakeup frame cannot be of both unicast and multicast address type. 4. Bit 0 is the enable for filter i. If Bit 0 is not set, filter i is disabled. Filter i Byte Mask: The filter i byte mask register defines the bytes of the packet that are examined by filter i (0, 1, 2, 3, .., 15) to determine whether or not a packet is a wake-up packet. 1. The MSB (31st bit) must be zero. 2. Bit j[30:0] is the byte mask. 3. If Bit j (byte number) of the byte mask is set, the CRC block processes the Filter i Offset + j of the incoming packet; otherwise Filter i Offset + j is ignored. Filter i Offset: The filter i offset register defines the offset (within the packet) from which the filter i examines the packets. 1. This 8-bit pattern-offset is the offset for the filter i first byte to be examined. 2. The minimum allowed offset is 12, which refers to the 13th byte of the packet. 3. The offset value 0 refers to the first byte of the packet. Filter i CRC-16: The filter i CRC-16 register contains the CRC-16 value calculated from the pattern and the byte mask programmed in the Remote Wakeup filter register. 1. The 16-bit CRC calculation uses the following polynomial: G(x) = x^16 + x^15 + x^2 + 1 2. Each mask, used in the hash function calculation, is compared with a 16-bit value associated with that mask. Each filter has the following: (1)32-bit Mask: Each bit in this mask corresponds to one byte in the detected packet. If the bit is 1, the corresponding byte is taken into the CRC16 calculation. (2) 8-bit Offset Pointer: Specifies the byte to start the CRC-16 computation. The pointer and the mask are used together to locate the bytes to be used in the CRC-16 calculations.

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	WKUPFRMFTR RWK Packet Filter This field contains the various controls of RWK Packet filter. When the Remote Wakeup Filters are to be programmed, the entire set of wkuppktfilter_reg registers must be written. The wkuppktfilter_reg register is programmed by sequentially writing the eight, sixteen or thirty-two register values in MAC_RWK_Packet_Filter register for wkuppktfilter_reg0, wkuppktfilter_reg1, ..., wkuppktfilter_reg31 respectively. The wkuppktfilter_reg register is read in a similar way. The MAC updates the wkuppktfilter_reg register current pointer value in RWKPTR field of MAC_PMT_Control_Status register. The Remote Wakeup Filters are arranged in blocks of 4 filters each and each such block have eight 32-bit wide registers, viz. wkuppktfilter_reg0-7, wkuppktfilter_reg8-15, wkuppktfilter_reg16-23 and wkuppktfilter_reg24-31. The fields of Remote Wakeup Filter are described as follows: Filter i Command: The 4-bit filter i command controls the filter i operation. 1. Bit 3 specifies the address type, defining the destination address type of the pattern. When the bit is set, the pattern applies to only multicast packets; when the bit is reset, the pattern applies only to unicast packet. 2. Bit 2 (Inverse Mode), when set, reverses the logic of the CRC16 hash function signal, to reject a packet with matching CRC-16 value. Bit 2, along with Bit 1, allows a MAC to reject a subset of remote wake-up packets by creating filter logic such as "Pattern 1 AND NOT Pattern 2". 3. Bit 1 (And_Previous) implements the Boolean logic. When set, the result of the current entry is logically ANDed with the result of the previous filter. This AND logic allows a filter pattern longer than 32 bytes by splitting the mask among two, three, or four filters. This depends on the number of filters that have the And_Previous bit set. The details are provided below: (1) The And_Previous bit setting is applicable within a set of 4 filters. (2) Setting of And_Previous bit of filter that is not enabled has no effect, that is setting And_Previous bit of lowest number filter in the set of 4 filters has no effect. For example, setting of And_Previous bit of Filter 0 has no effect. (3) If And_Previous bit is set for filter to form AND chained filter, the AND chain breaks at the point any filter is not enabled. For example: If Filter 2 And_Previous bit is set (bit 1 in Filter 2 command is set) but Filter 1 is not enabled (bit 0 in Filter 1 command is reset), then only Filter 2 result is considered. If Filter 2 And_Previous bit is set (bit 1 in Filter 2 command is set), Filter 3 And_Previous bit is set (bit 1 in Filter 3 command is set), but Filter 1 is not enabled (bit 0 in Filter 1 command is reset), then only Filter 2 result ANDed with Filter 3 result is considered. If Filter 2 And_Previous bit is set (bit 1 in Filter 2 command is set), Filter 3 And_Previous bit is set (bit 1 in Filter 3 command is set), but Filter 2 is not enabled (bit 0 in Filter 2 command is reset), then since setting of Filter 2 And_Previous bit has no effect only Filter 1 result ORed with Filter 3 result is considered.

GMAC_RWK_Filter0_Byte_Mask

Address: Operational Base + offset (0x10c0)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	Filter0_Byte_Mask Filter0 32-bit Mask Each bit in this mask corresponds to one byte in the detected packet. If the bit is 1', the corresponding byte is taken into the CRC16 calculation.

GMAC_RWK_Filter1_Byte_Mask

Address: Operational Base + offset (0x10c4)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	Filter1_Byte_Mask Filter1 32-bit Mask Each bit in this mask corresponds to one byte in the detected packet. If the bit is 1', the corresponding byte is taken into the CRC16 calculation.

GMAC_RWK_Filter2_Byte_Mask

Address: Operational Base + offset (0x10c8)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	Filter2_Byte_Mask Filter2 32-bit Mask Each bit in this mask corresponds to one byte in the detected packet. If the bit is 1', the corresponding byte is taken into the CRC16 calculation.

GMAC_RWK_Filter3_Byte_Mask

Address: Operational Base + offset (0x10cc)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	Filter3_Byte_Mask Filter3 32-bit Mask Each bit in this mask corresponds to one byte in the detected packet. If the bit is 1', the corresponding byte is taken into the CRC16 calculation.

GMAC_RWK_Filter01_CRC

Address: Operational Base + offset (0x10d0)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	Filter1_CRC Filter1 CRC-16 This filter CRC-16 contains the CRC_16 value of the pattern. The 16-bit CRC calculation uses the following polynomial: G(x) = x^16 + x^15 + x^2 + 1
15:0	RW	0x0000	Filter0_CRC Filter0 CRC-16 This filter CRC-16 contains the CRC_16 value of the pattern. The 16-bit CRC calculation uses the following polynomial: G(x) = x^16 + x^15 + x^2 + 1

GMAC_RWK_Filter23_CRC

Address: Operational Base + offset (0x10d4)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	Filter3_CRC Filter3 CRC-16 This filter CRC-16 contains the CRC_16 value of the pattern. The 16-bit CRC calculation uses the following polynomial: G(x) = x^16 + x^15 + x^2 + 1
15:0	RW	0x0000	Filter2_CRC Filter2 CRC-16 This filter CRC-16 contains the CRC_16 value of the pattern. The 16-bit CRC calculation uses the following polynomial: G(x) = x^16 + x^15+ x^2 + 1

GMAC_RWK_Filter_Offset

Address: Operational Base + offset (0x10d8)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	Filter3_Offset Filter3 Offset This filter offset defines the offset (within the packet) from which the filter examines the packets. 1. This 8-bit pattern-offset is the offset for the filter first byte to be examined. 2. The minimum allowed offset is 12, which refers to the 13th byte of the packet. 3. The offset value 0 refers to the first byte of the packet.
23:16	RW	0x00	Filter2_Offset Filter2 Offset This filter offset defines the offset (within the packet) from which the filter examines the packets. 1. This 8-bit pattern-offset is the offset for the filter first byte to be examined. 2. The minimum allowed offset is 12, which refers to the 13th byte of the packet. 3. The offset value 0 refers to the first byte of the packet.
15:8	RW	0x00	Filter1_Offset Filter1 Offset This filter offset defines the offset (within the packet) from which the filter examines the packets. 1. This 8-bit pattern-offset is the offset for the filter first byte to be examined. 2. The minimum allowed offset is 12, which refers to the 13th byte of the packet. 3. The offset value 0 refers to the first byte of the packet.
7:0	RW	0x00	Filter0_Offset Filter0 Offset This filter offset defines the offset (within the packet) from which the filter examines the packets. 1. This 8-bit pattern-offset is the offset for the filter first byte to be examined. 2. The minimum allowed offset is 12, which refers to the 13th byte of the packet. 3. The offset value 0 refers to the first byte of the packet.

GMAC_RWK_Filter_Command

Address: Operational Base + offset (0x10dc)

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

Bit	Attr	Reset Value	Description
27:24	RW	0x0	Filter3_Command Filter3 Command The 4-bit filter command controls the filter operation. 1. Bit 3 specifies the address type, defining the destination address type of the pattern. When the bit is set, the pattern applies to only multicast packets; when the bit is reset, the pattern applies only to unicast packet. 2. Bit 2 (Inverse Mode), when set, reverses the logic of the CRC16 hash function signal, to reject a packet with matching CRC_16 value. 3. Bit 2, along with Bit 1, allows a MAC to reject a subset of remote wake-up packets by creating filter logic such as "Pattern 1 AND NOT Pattern 2". 4. Bit 1 (And_Previous) implements the Boolean logic. When set, the result of the current entry is logically ANDed with the result of the previous filter. This AND logic allows a filter pattern longer than 32 bytes by splitting the mask among two, three, or four filters. This depends on the number of filters that have the And_Previous bit set. 5. Bit 0 is the enable for filter. If Bit 0 is not set, filter is disabled.
23:20	RO	0x0	reserved
19:16	RW	0x0	Filter2_Command Filter2 Command The 4-bit filter command controls the filter operation. 1. Bit 3 specifies the address type, defining the destination address type of the pattern. When the bit is set, the pattern applies to only multicast packets; when the bit is reset, the pattern applies only to unicast packet. 2. Bit 2 (Inverse Mode), when set, reverses the logic of the CRC16 hash function signal, to reject a packet with matching CRC_16 value. 3. Bit 2, along with Bit 1, allows a MAC to reject a subset of remote wake-up packets by creating filter logic such as "Pattern 1 AND NOT Pattern 2". 4. Bit 1 (And_Previous) implements the Boolean logic. When set, the result of the current entry is logically ANDed with the result of the previous filter. This AND logic allows a filter pattern longer than 32 bytes by splitting the mask among two, three, or four filters. This depends on the number of filters that have the And_Previous bit set. 5. Bit 0 is the enable for filter. If Bit 0 is not set, filter is disabled.
15:12	RO	0x0	reserved

Bit	Attr	Reset Value	Description
11:8	RW	0x0	Filter1_Command Filter1 Command The 4-bit filter command controls the filter operation. 1. Bit 3 specifies the address type, defining the destination address type of the pattern. When the bit is set, the pattern applies to only multicast packets; when the bit is reset, the pattern applies only to unicast packet. 2. Bit 2 (Inverse Mode), when set, reverses the logic of the CRC16 hash function signal, to reject a packet with matching CRC_16 value. 3. Bit 2, along with Bit 1, allows a MAC to reject a subset of remote wake-up packets by creating filter logic such as "Pattern 1 AND NOT Pattern 2". 4. Bit 1 (And_Previous) implements the Boolean logic. When set, the result of the current entry is logically ANDed with the result of the previous filter. This AND logic allows a filter pattern longer than 32 bytes by splitting the mask among two, three, or four filters. This depends on the number of filters that have the And_Previous bit set. 5. Bit 0 is the enable for filter. If Bit 0 is not set, filter is disabled.
7:4	RO	0x0	reserved
3:0	RW	0x0	Filter0_Command Filter0 Command The 4-bit filter command controls the filter operation. 1. Bit 3 specifies the address type, defining the destination address type of the pattern. When the bit is set, the pattern applies to only multicast packets; when the bit is reset, the pattern applies only to unicast packet. 2. Bit 2 (Inverse Mode), when set, reverses the logic of the CRC16 hash function signal, to reject a packet with matching CRC_16 value. 3. Bit 2, along with Bit 1, allows a MAC to reject a subset of remote wake-up packets by creating filter logic such as "Pattern 1 AND NOT Pattern 2". 4. Bit 1 (And_Previous) implements the Boolean logic. When set, the result of the current entry is logically ANDed with the result of the previous filter. This AND logic allows a filter pattern longer than 32 bytes by splitting the mask among two, three, or four filters. This depends on the number of filters that have the And_Previous bit set. 5. Bit 0 is the enable for filter. If Bit 0 is not set, filter is disabled.

GMAC_MAC_LPI_Control_Status

Address: Operational Base + offset (0x00d0)

Bit	Attr	Reset Value	Description
31:22	RO	0x000	reserved

Bit	Attr	Reset Value	Description
21	RW	0x0	LPITCSE LPI Tx Clock Stop Enable When this bit is set, the MAC asserts sbd_tx_clk_gating_ctrl_o signal high after it enters Tx LPI mode to indicate that the Tx clock to MAC can be stopped. When this bit is reset, the MAC does not assert sbd_tx_clk_gating_ctrl_o signal high after it enters Tx LPI mode. If RGMII Interface is selected, the Tx clock is required for transmitting the LPI pattern. The Tx Clock cannot be gated and so the LPITCSE bit cannot be programmed. Values: 1'b0: LPI Tx Clock Stop is disabled 1'b1: LPI Tx Clock Stop is enabled
20	RW	0x0	LPIATE LPI Timer Enable This bit controls the automatic entry of the MAC Transmitter into and exit out of the LPI state. When LPIATE, LPITXA and LPIEN bits are set, the MAC Transmitter enters LPI state only when the complete MAC TX data path is IDLE for a period indicated by the MAC_LPI_Entry_Timer register. After entering LPI state, if the data path becomes non-IDLE (due to a new packet being accepted for transmission), the Transmitter exits LPI state but does not clear LPIEN bit. This enables the re-entry into LPI state when it is IDLE again. When LPIATE is 0, the LPI Auto timer is disabled and MAC Transmitter enters LPI state based on the settings of LPITXA and LPIEN bit descriptions. Values: 1'b0: LPI Timer is disabled 1'b1: LPI Timer is enabled
19	RW	0x0	LPITXA LPI Tx Automate This bit controls the behavior of the MAC when it is entering or coming out of the LPI mode on the Transmit side. This bit is not functional in the EQOS-CORE configurations in which the Tx clock gating is done during the LPI mode. If the LPITXA and LPIEN bits are set to 1, the MAC enters the LPI mode only after all outstanding packets (in the core) and pending packets (in the application interface) have been transmitted. The MAC comes out of the LPI mode when the application sends any packet for transmission or the application issues a Tx FIFO Flush command. In addition, the MAC automatically clears the LPIEN bit when it exits the LPI state. If Tx FIFO Flush is set in the FTQ bit of MTL_TxQ0_Operation_Mode register, when the MAC is in the LPI mode, it exits the LPI mode. When this bit is 0, the LPIEN bit directly controls behavior of the MAC when it is entering or coming out of the LPI mode. Values: 1'b0: LPI Tx Automate is disabled 1'b1: LPI Tx Automate is enabled

Bit	Attr	Reset Value	Description
18	RW	0x0	PLSEN PHY Link Status Enable This bit enables the link status received on the RGMII, SGMII, or SMII Receive paths to be used for activating the LPI LS TIMER. When this bit is set, the MAC uses the link-status bits of the MAC_PHYIF_Control_Status register and the PLS bit for the LPI LS Timer trigger. When this bit is reset, the MAC ignores the link-status bits of the MAC_PHYIF_Control_Status register and takes only the PLS bit. Values: 1'b0: PHY Link Status is disabled 1'b1: PHY Link Status is enabled
17	RW	0x0	PLS PHY Link Status This bit indicates the link status of the PHY. The MAC Transmitter asserts the LPI pattern only when the link status is up (OKAY) at least for the time indicated by the LPI LS TIMER. When this bit is set, the link is considered to be okay (UP) and when this bit is reset, the link is considered to be down. Values: 1'b0: link is down 1'b1: link is okay (UP)
16	RW	0x0	LPIEN LPI Enable When this bit is set, it instructs the MAC Transmitter to enter the LPI state. When this bit is reset, it instructs the MAC to exit the LPI state and resume normal transmission. This bit is cleared when the LPITXA bit is set and the MAC exits the LPI state because of the arrival of a new packet for transmission. Values: 1'b0: LPI state is disabled 1'b1: LPI state is enabled
15:10	RO	0x00	reserved
9	RO	0x0	RLPIST Receive LPI State When this bit is set, it indicates that the MAC is receiving the LPI pattern on the GMII or MII interface. Values: 1'b0: Receive LPI state not detected 1'b1: Receive LPI state detected
8	RO	0x0	TLPIST Transmit LPI State When this bit is set, it indicates that the MAC is transmitting the LPI pattern on the GMII or MII interface. Values: 1'b0: Transmit LPI state not detected 1'b1: Transmit LPI state detected
7:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3	RW	0x0	RLPIEX Receive LPI Exit When this bit is set, it indicates that the MAC Receiver has stopped receiving the LPI pattern on the GMII or MII interface, exited the LPI state, and resumed the normal reception. This bit is cleared by a read into this register (or this bit is written to 1 when RCWE bit in MAC_CSR_SW_Ctrl register is set). Note: This bit may not be set if the MAC stops receiving the LPI pattern for a very short duration, such as, less than three clock cycles of CSR clock. Values: 1'b0: Receive LPI exit not detected 1'b1: Receive LPI exit detected
2	RO	0x0	RLPIEN Receive LPI Entry When this bit is set, it indicates that the MAC Receiver has received an LPI pattern and entered the LPI state. This bit is cleared by a read into this register (or this bit is written to 1 when RCWE bit in MAC_CSR_SW_Ctrl register is set). Note: This bit may not be set if the MAC stops receiving the LPI pattern for a very short duration, such as, less than three clock cycles of CSR clock. Values: 1'b0: Receive LPI entry not detected 1'b1: Receive LPI entry detected
1	RO	0x0	TLPIEX Transmit LPI Exit When this bit is set, it indicates that the MAC transmitter exited the LPI state after the application cleared the LPIEN bit and the LPI TW Timer has expired. This bit is cleared by a read into this register (or this bit is written to 1 when RCWE bit in MAC_CSR_SW_Ctrl register is set). Values: 1'b0: Transmit LPI exit not detected 1'b1: Transmit LPI exit detected
0	RO	0x0	TLPIEN Transmit LPI Entry When this bit is set, it indicates that the MAC Transmitter has entered the LPI state because of the setting of the LPIEN bit. This bit is cleared by a read into this register (or this bit is written to 1 when RCWE bit in MAC_CSR_SW_Ctrl register is set). Values: 1'b0: Transmit LPI entry not detected 1'b1: Transmit LPI entry detected

GMAC_MAC_LPI_Timers_Control

Address: Operational Base + offset (0x00d4)

Bit	Attr	Reset Value	Description
31:26	RO	0x00	reserved

Bit	Attr	Reset Value	Description
25:16	RW	0x3e8	LST LPI LS Timer This field specifies the minimum time (in milliseconds) for which the link status from the PHY should be up (OKAY) before the LPI pattern can be transmitted to the PHY. The MAC does not transmit the LPI pattern even when the LPIEN bit is set unless the LPI LS Timer reaches the programmed terminal count. The default value of the LPI LS Timer is 1000 (1 sec) as defined in the IEEE standard.
15:0	RW	0x0000	TWT LPI TW Timer This field specifies the minimum time (in microseconds) for which the MAC waits after it stops transmitting the LPI pattern to the PHY and before it resumes the normal transmission. The TLPIEX status bit is set after the expiry of this timer.

GMAC_MAC_LPI_Entry_Timer

Address: Operational Base + offset (0x00d8)

Bit	Attr	Reset Value	Description
31:20	RO	0x000	reserved
19:3	RW	0x00000	LPIET LPI Entry Timer This field specifies the time in microseconds the MAC waits to enter LPI mode, after it has transmitted all the frames. This field is valid and used only when LPITE and LPITXA are set to 1. Bits [2:0] are read-only so that the granularity of this timer is in steps of 8 micro-seconds.

GMAC_MAC_1US_Tic_Counter

Address: Operational Base + offset (0x00dc)

Bit	Attr	Reset Value	Description
31:12	RO	0x00000	reserved
11:0	RW	0x063	TIC_1US_CNTR 1US TIC Counter The application must program this counter so that the number of clock cycles of CSR clock is 1us. (Subtract 1 from the value before programming). For example if the CSR clock is 100MHz then this field needs to be programmed to value 100 - 1 = 99 (which is 0x63). This is required to generate the 1US events that are used to update some of the EEE related counters.

GMAC_MAC_PHYIF_Control_Status

Address: Operational Base + offset (0x00f8)

Bit	Attr	Reset Value	Description
31:20	RO	0x000	reserved
19	RO	0x0	LNKSTS Link Status This bit indicates whether the link is up (1'b1) or down (1'b0). Values: 1'b0: Link down 1'b1: Link up

Bit	Attr	Reset Value	Description
18:17	RO	0x0	LNKSPEED Link Speed This bit indicates the current speed of the link. Bit 2 is reserved when the MAC is configured for the SMII PHY interface. Values: 2'b00: 2.5 MHz 2'b01: 25 MHz 2'b10: 125 MHz 2'b11: Reserved
16	RO	0x0	LNKMOD Link Mode This bit indicates the current mode of operation of the link. Values: 1'b0: Half-duplex mode 1'b1: Full-duplex mode
15:2	RO	0x0000	reserved
1	RW	0x0	LUD Link Up or Down This bit indicates whether the link is up or down during transmission of configuration in the RGMII, SGMII, or SMII interface. Values: 1'b0: Link down 1'b1: Link up
0	RW	0x0	TC Transmit Configuration in RGMII, SGMII, or SMII When set, this bit enables the transmission of duplex mode, link speed, and link up or down information to the PHY in the RGMII, SMII, or SGMII port. When this bit is reset, no such information is driven to the PHY. The details of this feature are provided in the following sections: 1. "Reduced Gigabit Media Independent Interface" 2. "Serial Media Independent Interface" 3. "Serial Gigabit Media Independent Interface" Values: 1'b0: Disable Transmit Configuration in RGMII, SGMII, or SMII 1'b1: Enable Transmit Configuration in RGMII, SGMII, or SMII

GMAC_MAC_Version

Address: Operational Base + offset (0x0110)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15:8	RW	0x30	USERVER User-defined Version (configured with coreConsultant)
7:0	RW	0x51	SNPSVER Synopsys-defined Version

GMAC_MAC_Debug

Address: Operational Base + offset (0x0114)

Bit	Attr	Reset Value	Description
31:19	RO	0x0000	reserved

Bit	Attr	Reset Value	Description
18:17	RO	0x0	TFCSTS MAC Transmit Packet Controller Status This field indicates the state of the MAC Transmit Packet Controller module. Values: 2'b00: Idle state 2'b01: Waiting for one of the following: Status of the previous packet OR IPG or back off period to be over 2'b10: Generating and transmitting a Pause control packet (in full-duplex mode) 2'b11: Transferring input packet for transmission
16	RO	0x0	TPESTS MAC GMII or MII Transmit Protocol Engine Status When this bit is set, it indicates that the MAC GMII or MII transmit protocol engine is actively transmitting data, and it is not in the Idle state. Values: 1'b0: MAC GMII or MII Transmit Protocol Engine Status not detected 1'b1: MAC GMII or MII Transmit Protocol Engine Status detected
15:3	RO	0x0000	reserved
2:1	RO	0x0	RFCFCSTS MAC Receive Packet Controller FIFO Status When this bit is set, this field indicates the active state of the small FIFO Read and Write controllers of the MAC Receive Packet Controller module.
0	RO	0x0	RPESTS MAC GMII or MII Receive Protocol Engine Status When this bit is set, it indicates that the MAC GMII or MII receive protocol engine is actively receiving data, and it is not in the Idle state. Values: 1'b0: MAC GMII or MII Receive Protocol Engine Status not detected 1'b1: MAC GMII or MII Receive Protocol Engine Status detected

GMAC_MAC_HW_Feature0

Address: Operational Base + offset (0x011c)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	ACTPHYSEL Active PHY Selected When you have multiple PHY interfaces in your configuration, this field indicates the sampled value of phy_intf_sel_i during reset de-assertion. Values: 4'b0000: GMII or MII 4'b0001: RGMII 4'b0010: SGMII 4'b0011: TBI 4'b0100: RMII 4'b0101: RTBI 4'b0110: SMII 4'b0111: RevMII

Bit	Attr	Reset Value	Description
27	RO	0x0	SAVLANINS Source Address or VLAN Insertion Enable This bit is set to 1 when the Enable SA and VLAN Insertion on Tx option is selected. Values: 1'b0: Source Address or VLAN Insertion Enable option is not selected 1'b1: Source Address or VLAN Insertion Enable option is selected
26:25	RO	0x3	TSSTSSEL Timestamp System Time Source This bit indicates the source of the Timestamp system time: This bit is set to 1 when the Enable IEEE 1588 Timestamp Support option is selected. Values: 2'b00: Internal 2'b01: External 2'b10: Both 2'b11: Reserved
24	RO	0x0	MACADR64SEL MAC Addresses 64-127 Selected This bit is set to 1 when the Enable Additional 64 MAC Address Registers (64-127) option is selected. Values: 1'b0: MAC Addresses 64-127 Select option is not selected 1'b1: MAC Addresses 64-127 Select option is selected
23	RO	0x0	MACADR32SEL MAC Addresses 32-63 Selected This bit is set to 1 when the Enable Additional 32 MAC Address Registers (32-63) option is selected. Values: 1'b0: MAC Addresses 32-63 Select option is not selected 1'b1: MAC Addresses 32-63 Select option is selected
22:18	RO	0x00	ADDMACADRSEL MAC Addresses 1-31 Selected This bit is set to 1 when the non-zero value is selected for Enable Additional 1-31 MAC Address Registers option.
17	RO	0x0	reserved
16	RO	0x1	RXCOESEL Receive Checksum Offload Enabled This bit is set to 1 when the Enable Receive TCP/IP Checksum Check option is selected. Values: 1'b0: Receive Checksum Offload Enable option is not selected 1'b1: Receive Checksum Offload Enable option is selected
15	RO	0x0	reserved
14	RO	0x1	TXCOESEL Transmit Checksum Offload Enabled This bit is set to 1 when the Enable Transmit TCP/IP Checksum Insertion option is selected. Values: 1'b0: Transmit Checksum Offload Enable option is not selected 1'b1: Transmit Checksum Offload Enable option is selected

Bit	Attr	Reset Value	Description
13	RO	0x1	EEESEL Energy Efficient Ethernet Enabled This bit is set to 1 when the Enable Energy Efficient Ethernet (EEE) option is selected. Values: 1'b0: Energy Efficient Ethernet Enable option is not selected 1'b1: Energy Efficient Ethernet Enable option is selected
12	RO	0x1	TSSEL IEEE 1588-2008 Timestamp Enabled This bit is set to 1 when the Enable IEEE 1588 Timestamp Support option is selected. Values: 1'b0: IEEE 1588-2008 Timestamp Enable option is not selected. 1'b1: IEEE 1588-2008 Timestamp Enable option is selected.
11:10	RO	0x0	reserved
9	RO	0x0	ARPOFFSEL ARP Offload Enabled This bit is set to 1 when the Enable IPv4 ARP Offload option is selected. Values: 1'b0: ARP Offload Enable option is not selected 1'b1: ARP Offload Enable option is selected
8	RO	0x1	MMCSEL RMON Module Enable This bit is set to 1 when the Enable MAC Management Counters (MMC) option is selected. Values: 1'b0: RMON Module Enable option is not selected 1'b1: RMON Module Enable option is selected
7	RO	0x1	MGKSEL PMT Magic Packet Enable This bit is set to 1 when the Enable Magic Packet Detection option is selected. Values: 1'b0: PMT Magic Packet Enable option is not selected 1'b1: PMT Magic Packet Enable option is selected
6	RO	0x1	RWKSEL PMT Remote Wake-up Packet Enable This bit is set to 1 when the Enable Remote Wake-Up Packet Detection option is selected. Values: 1'b0: PMT Remote Wake-up Packet Enable option is not selected 1'b1: PMT Remote Wake-up Packet Enable option is selected
5	RO	0x1	SMASEL SMA (MDIO) Interface This bit is set to 1 when the Enable Station Management (MDIO Interface) option is selected. Values: 1'b0: SMA (MDIO) Interface not selected 1'b1: SMA (MDIO) Interface selected

Bit	Attr	Reset Value	Description
4	RO	0x0	VLHASH VLAN Hash Filter Selected This bit is set to 1 when the Enable VLAN Hash Table Based Filtering option is selected. Values: 1'b0: VLAN Hash Filter not selected 1'b1: VLAN Hash Filter selected
3	RO	0x0	PCSSEL PCS Registers (TBI, SGMII, or RTBI PHY interface) This bit is set to 1 when the TBI, SGMII, or RTBI PHY interface option is selected. Values: 1'b0: No PCS Registers (TBI, SGMII, or RTBI PHY interface) 1'b1: PCS Registers (TBI, SGMII, or RTBI PHY interface)
2	RO	0x1	HDSEL Half-duplex Support This bit is set to 1 when the half-duplex mode is selected. Values: 1'b0: No Half-duplex support 1'b1: Half-duplex support
1	RO	0x1	GMIISEL 1000 Mbps Support This bit is set to 1 when 1000 Mbps is selected as the Mode of Operation. Values: 1'b0: No 1000 Mbps support 1'b1: 1000 Mbps support
0	RO	0x1	MIISEL 10 or 100 Mbps Support This bit is set to 1 when 10/100 Mbps is selected as the Mode of Operation. Values: 1'b0: No 10 or 100 Mbps support 1'b1: 10 or 100 Mbps support

GMAC_MAC_HW_Feature1

Address: Operational Base + offset (0x0120)

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30:27	RO	0x0	L3L4FNUM Total number of L3 or L4 Filters This field indicates the total number of L3 or L4 filters: Values: 4'b0000: No L3 or L4 Filter 4'b0001: 1 L3 or L4 Filter 4'b0010: 2 L3 or L4 Filters 4'b0011: 3 L3 or L4 Filters 4'b0100: 4 L3 or L4 Filters 4'b0101: 5 L3 or L4 Filters 4'b0110: 6 L3 or L4 Filters 4'b0111: 7 L3 or L4 Filters 4'b1000: 8 L3 or L4 Filters
26	RO	0x0	reserved

Bit	Attr	Reset Value	Description
25:24	RO	0x1	HASHTBLSZ Hash Table Size This field indicates the size of the hash table: Values: 2'b00: No hash table 2'b01: 64 2'b10: 128 2'b11: 256
23	RO	0x0	POUOST One Step for PTP over UDP/IP Feature Enable This bit is set to 1 when the Enable One step timestamp for PTP over UDP/IP feature is selected. Values: 1'b0: One Step for PTP over UDP/IP Feature is not selected 1'b1: One Step for PTP over UDP/IP Feature is selected
22	RO	0x0	reserved
21	RO	0x0	RAVSEL Rx Side Only AV Feature Enable This bit is set to 1 when the Enable Audio Video Bridging option on Rx Side Only is selected. Values: 1'b0: Rx Side Only AV Feature is not selected 1'b1: Rx Side Only AV Feature is selected
20	RO	0x0	AVSEL AV Feature Enable This bit is set to 1 when the Enable Audio Video Bridging option is selected. Values: 1'b0: AV Feature is not selected 1'b1: AV Feature is selected
19	RO	0x1	DBGMEMA DMA Debug Registers Enable This bit is set to 1 when the Debug Mode Enable option is selected. Values: 1'b0: DMA Debug Registers option is not selected 1'b1: DMA Debug Registers option is selected
18	RO	0x1	TSOEN TCP Segmentation Offload Enable This bit is set to 1 when the Enable TCP Segmentation Offloading for TCP/IP Packets option is selected. Values: 1'b0: TCP Segmentation Offload Feature is not selected 1'b1: TCP Segmentation Offload Feature is selected
17	RO	0x0	SPHEN Split Header Feature Enable This bit is set to 1 when the Enable Split Header Structure option is selected. Values: 1'b0: Split Header Feature is not selected 1'b1: Split Header Feature is selected

Bit	Attr	Reset Value	Description
16	RO	0x0	DCBEN DCB Feature Enable This bit is set to 1 when the Enable Data Center Bridging option is selected. Values: 1'b0: DCB Feature is not selected 1'b1: DCB Feature is selected
15:14	RO	0x0	ADDR64 Address Width This field indicates the configured address width: Values: 2'b00: 32 2'b01: 40 2'b10: 48 2'b11: Reserved
13	RO	0x0	ADVTHWORD IEEE 1588 High Word Register Enable This bit is set to 1 when the Add IEEE 1588 Higher Word Register option is selected. Values: 1'b0: IEEE 1588 High Word Register option is not selected 1'b1: IEEE 1588 High Word Register option is selected
12	RO	0x0	PTOEN PTP Offload Enable This bit is set to 1 when the Enable PTP Timestamp Offload Feature is selected. Values: 1'b0: PTP Offload feature is not selected 1'b1: PTP Offload feature is selected
11	RO	0x0	OSTEN One-Step Timestamping Enable This bit is set to 1 when the Enable One-Step Timestamp Feature is selected. Values: 1'b0: One-Step Timestamping feature is not selected 1'b1: One-Step Timestamping feature is selected
10:6	RO	0x07	TXFIFOSIZE MTL Transmit FIFO Size This field contains the configured value of MTL Tx FIFO in bytes expressed as Log to base 2 minus 7, that is, Log2(TXFIFO_SIZE) -7: Values: 5'b00000: 128 bytes 5'b00001: 256 bytes 5'b00010: 512 bytes 5'b00011: 1024 bytes 5'b00100: 2048 bytes 5'b00101: 4096 bytes 5'b00110: 8192 bytes 5'b00111: 16384 bytes 5'b01000: 32 KB 5'b01001: 64 KB 5'b01010: 128 KB 5'b01011: Reserved

Bit	Attr	Reset Value	Description
5	RO	0x0	SPRAM Single Port RAM Enable This bit is set to 1 when the Use single port RAM Feature is selected. Values: 1'b0: Single Port RAM feature is not selected 1'b1: Single Port RAM feature is selected
4:0	RO	0x08	RXFIFOSIZE MTL Receive FIFO Size This field contains the configured value of MTL Rx FIFO in bytes expressed as Log to base 2 minus 7, that is, Log2(RXFIFO_SIZE) -7: Values: 5'b00000: 128 bytes 5'b00001: 256 bytes 5'b00010: 512 bytes 5'b00011: 1024 bytes 5'b00100: 2048 bytes 5'b00101: 4096 bytes 5'b00110: 8192 bytes 5'b00111: 16384 bytes 5'b01000: 32 KB 5'b01001: 64 KB 5'b01010: 128 KB 5'b01011: 256 KB 5'b01100: Reserved

GMAC_MAC_HW_Feature2

Address: Operational Base + offset (0x0124)

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30:28	RO	0x1	AUXSNAPNUM Number of Auxiliary Snapshot Inputs This field indicates the number of auxiliary snapshot inputs: Values: 3'b000: No auxiliary input 3'b001: 1 auxiliary input 3'b010: 2 auxiliary input 3'b011: 3 auxiliary input 3'b100: 4 auxiliary input 3'b101: Reserved
27	RO	0x0	reserved
26:24	RO	0x0	PPSOUTNUM Number of PPS Outputs This field indicates the number of PPS outputs: Values: 3'b000: No PPS output 3'b001: 1 PPS output 3'b010: 2 PPS output 3'b011: 3 PPS output 3'b100: 4 PPS output 3'b101: Reserved
23:22	RO	0x0	reserved

Bit	Attr	Reset Value	Description
21:18	RO	0x0	TXCHCNT Number of DMA Transmit Channels This field indicates the number of DMA Transmit channels: Values: 4'b0000: 1 MTL Tx Channel 4'b0001: 2 MTL Tx Channels 4'b0010: 3 MTL Tx Channels 4'b0011: 4 MTL Tx Channels 4'b0100: 5 MTL Tx Channels 4'b0101: 6 MTL Tx Channels 4'b0110: 7 MTL Tx Channels 4'b0111: 8 MTL Tx Channels
17:16	RO	0x0	reserved
15:12	RO	0x0	RXCHCNT Number of DMA Receive Channels This field indicates the number of DMA Receive channels: Values: 4'b0000: 1 MTL Rx Channel 4'b0001: 2 MTL Rx Channels 4'b0010: 3 MTL Rx Channels 4'b0011: 4 MTL Rx Channels 4'b0100: 5 MTL Rx Channels 4'b0101: 6 MTL Rx Channels 4'b0110: 7 MTL Rx Channels 4'b0111: 8 MTL Rx Channels
11:10	RO	0x0	reserved
9:6	RO	0x0	TXQCNT Number of MTL Transmit Queues This field indicates the number of MTL Transmit queues: Values: 4'b0000: 1 MTL Tx Queue 4'b0001: 2 MTL Tx Queues 4'b0010: 3 MTL Tx Queues 4'b0011: 4 MTL Tx Queues 4'b0100: 5 MTL Tx Queues 4'b0101: 6 MTL Tx Queues 4'b0110: 7 MTL Tx Queues 4'b0111: 8 MTL Tx Queues
5:4	RO	0x0	reserved
3:0	RO	0x0	RXQCNT Number of MTL Receive Queues This field indicates the number of MTL Receive queues: Values: 4'b0000: 1 MTL Rx Queue 4'b0001: 2 MTL Rx Queues 4'b0010: 3 MTL Rx Queues 4'b0011: 4 MTL Rx Queues 4'b0100: 5 MTL Rx Queues 4'b0101: 6 MTL Rx Queues 4'b0110: 7 MTL Rx Queues 4'b0111: 8 MTL Rx Queues

GMAC_MAC_HW_Feature3

Address: Operational Base + offset (0x0128)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:28	RO	0x0	ASP Automotive Safety Package Following are the encoding for the different Safety features. Values: 2'b00: No Safety features selected 2'b01: Only "ECC protection for external memory" feature is selected 2'b10: All the Automotive Safety features are selected without the "Parity Port Enable for external interface" feature 2'b11: All the Automotive Safety features are selected with the "Parity Port Enable for external interface" feature
27	RO	0x0	TBSSEL Time Based Scheduling Enable This bit is set to 1 when the Time Based Scheduling feature is selected. Values: 1'b0: Time Based Scheduling Enable feature is not selected 1'b1: Time Based Scheduling Enable feature is selected
26	RO	0x0	FPESEL Frame Preemption Enable This bit is set to 1 when the Enable Frame preemption feature is selected. Values: 1'b0: Frame Preemption Enable feature is not selected 1'b1: Frame Preemption Enable feature is selected
25:22	RO	0x0	reserved
21:20	RO	0x0	ESTWID Width of the Time Interval field in the Gate Control List This field indicates the width of the Configured Time Interval Field. Values: 2'b00: Width not configured 2'b01: 16 2'b10: 20 2'b11: 24
19:17	RW	0x0	ESTDEP Depth of the Gate Control List This field indicates the depth of Gate Control list expressed as Log2(DWC_EQOS_EST_DEP)-5. Values: 3'b000: No Depth configured 3'b001: 64 3'b010: 128 3'b011: 256 3'b100: 512 3'b101: 1024 3'b110: Reserved

Bit	Attr	Reset Value	Description
16	RO	0x0	ESTSEL Enhancements to Scheduling Traffic Enable This bit is set to 1 when the Enable Enhancements to Scheduling Traffic feature is selected. Values: 1'b0: Enable Enhancements to Scheduling Traffic feature is not selected 1'b1: Enable Enhancements to Scheduling Traffic feature is selected
15	RO	0x0	reserved
14:13	RO	0x0	FRPES Flexible Receive Parser Table Entries size This field indicates the Max Number of Parser Entries supported by Flexible Receive Parser. Values: 2'b00: 64 Entries 2'b01: 128 Entries 2'b10: 256 Entries 2'b11: Reserved
12:11	RO	0x0	FRPBS Flexible Receive Parser Buffer size This field indicates the supported Max Number of bytes of the packet data to be Parsed by Flexible Receive Parser. Values: 2'b00: 64 Bytes 2'b01: 128 Bytes 2'b10: 256 Bytes 2'b11: Reserved
10	RO	0x0	FRPSEL Flexible Receive Parser Selected This bit is set to 1 when the Enable Flexible Programmable Receive Parser option is selected. Values: 1'b0: Flexible Receive Parser feature is not selected 1'b1: Flexible Receive Parser feature is selected
9	RO	0x0	PDUPSEL Broadcast/Multicast Packet Duplication This bit is set to 1 when the Broadcast/Multicast Packet Duplication feature is selected. Values: 1'b0: Broadcast/Multicast Packet Duplication feature is not selected 1'b1: Broadcast/Multicast Packet Duplication feature is selected
8:6	RO	0x0	reserved
5	RO	0x0	DVLAN Double VLAN Tag Processing Selected This bit is set to 1 when the Enable Double VLAN Processing Feature is selected. Values: 1'b0: Double VLAN option is not selected 1'b1: Double VLAN option is selected

Bit	Attr	Reset Value	Description
4	RO	0x0	CBTISEL Queue/Channel based VLAN tag insertion on Tx Enable This bit is set to 1 when the Enable Queue/Channel based VLAN tag insertion on Tx Feature is selected. Values: 1'b0: Enable Queue/Channel based VLAN tag insertion on Tx feature is not selected 1'b1: Enable Queue/Channel based VLAN tag insertion on Tx feature is selected
3	RO	0x0	reserved
2:0	RO	0x0	NRVF Number of Extended VLAN Tag Filters Enabled This field indicates the Number of Extended VLAN Tag Filters selected: Values: 3'b000: No Extended Rx VLAN Filters 3'b001: 4 Extended Rx VLAN Filters 3'b010: 8 Extended Rx VLAN Filters 3'b011: 16 Extended Rx VLAN Filters 3'b100: 24 Extended Rx VLAN Filters 3'b101: 32 Extended Rx VLAN Filters 3'b110: Reserved

GMAC_MAC_MDIO_Address

Address: Operational Base + offset (0x0200)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27	RW	0x0	PSE Preamble Suppression Enable When this bit is set, the SMA suppresses the 32-bit preamble and transmits MDIO frames with only 1 preamble bit. When this bit is 0, the MDIO frame always has 32 bits of preamble as defined in the IEEE specifications. Values: 1'b0: Preamble Suppression disabled 1'b1: Preamble Suppression enabled
26	RW	0x0	BTB Back to Back transactions When this bit is set and the NTC has value greater than 0, then the MAC informs the completion of a read or write command at the end of frame transfer (before the trailing clocks are transmitted). The software can thus initiate the next command which is executed immediately irrespective of the number trailing clocks generated for the previous frame. When this bit is reset, then the read/write command completion (GB is cleared) only after the trailing clocks are generated. In this mode, it is ensured that the NTC is always generated after each frame. This bit must not be set when NTC=0. Values: 1'b0: Back to Back transactions disabled 1'b1: Back to Back transactions enabled

Bit	Attr	Reset Value	Description
25:21	RW	0x00	PA Physical Layer Address This field indicates which Clause 22 PHY devices (out of 32 devices) the MAC is accessing. For RevMII, this field gives the PHY Address of the RevMII module. This field indicates which Clause 45 capable PHYs (out of 32 PHYs) the MAC is accessing.
20:16	RW	0x00	RDA Register/Device Address These bits select the PHY register in selected Clause 22 PHY device. For RevMII, these bits select the CSR register in the RevMII Registers set. These bits select the Device (MMD) in selected Clause 45 capable PHY.
15	RO	0x0	reserved
14:12	RW	0x0	NTC Number of Trailing Clocks This field controls the number of trailing clock cycles generated on gmii_mdc_o (MDC) after the end of transmission of MDIO frame. The valid values can be from 0 to 7. Programming the value to 3'h3 indicates that there are additional three clock cycles on the MDC line after the end of MDIO frame transfer.
11:8	RW	0x0	CR CSR Clock Range The CSR Clock Range selection determines the frequency of the MDC clock according to the CSR clock frequency used in your design: 4'b0000: CSR clock = 60-100 MHz; MDC clock = CSR clock/42 4'b0001: CSR clock = 100-150 MHz; MDC clock = CSR clock/62 4'b0010: CSR clock = 20-35 MHz; MDC clock = CSR clock/16 4'b0011: CSR clock = 35-60 MHz; MDC clock = CSR clock/26 4'b0100: CSR clock = 150-250 MHz; MDC clock = CSR clock/102 4'b0101: CSR clock = 250-300 MHz; MDC clock = CSR clock/124 4'b0110: CSR clock = 300-500 MHz; MDC clock = CSR clock/204 4'b0111: CSR clock = 500-800 MHz; MDC clock = CSR clock/324 The suggested range of CSR clock frequency applicable for each value (when Bit 11 = 0) ensures that the MDC clock is approximately between 1.0 MHz to 2.5 MHz frequency range. When Bit 11 is set, you can achieve a higher frequency of the MDC clock than the frequency limit of 2.5 MHz (specified in the IEEE 802.3) and program a clock divider of lower value. For example, when CSR clock is of 100 MHz frequency and you program these bits as 1010, the resultant MDC clock is of 12.5 MHz which is above the range specified in IEEE 802.3. Program the following values only if the interfacing chips support faster MDC clocks: 4'b1000: CSR clock/4 4'b1001: CSR clock/6 4'b1010: CSR clock/8 4'b1011: CSR clock/10 4'b1100: CSR clock/12 4'b1101: CSR clock/14 4'b1110: CSR clock/16 4'b1111: CSR clock/18 These bits are not used for accessing RevMII. These bits are read-only if the RevMII interface is selected as single PHY interface.
7:5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4	RW	0x0	SKAP Skip Address Packet When this bit is set, the SMA does not send the address packets before read, write, or post-read increment address packets. This bit is valid only when C45E is set. Values: 1'b0: Skip Address Packet is disabled 1'b1: Skip Address Packet is enabled
3	RW	0x0	GOC_1 GMII Operation Command 1 This bit is higher bit of the operation command to the PHY or RevMII, GOC_1 and GOC_O is encoded as follows: 2'b00: Reserved 2'b01: Write 2'b10: Post Read Increment Address for Clause 45 PHY 2'b11: Read When Clause 22 PHY or RevMII is enabled, only Write and Read commands are valid. Values: 1'b0: GMII Operation Command 1 is disabled 1'b1: GMII Operation Command 1 is enabled
2	RW	0x0	GOC_0 GMII Operation Command 0 This is the lower bit of the operation command to the PHY or RevMII. When in SMA mode (MDIO master) this bit along with GOC_1 determines the operation to be performed to the PHY. When only RevMII is selected in configuration this bit is read-only and tied to 1. Values: 1'b0: GMII Operation Command 0 is disabled 1'b1: GMII Operation Command 0 is enabled
1	RW	0x0	C45E Clause 45 PHY Enable When this bit is set, Clause 45 capable PHY is connected to MDIO. When this bit is reset, Clause 22 capable PHY is connected to MDIO. Values: 1'b0: Clause 45 PHY is disabled 1'b1: Clause 45 PHY is enabled

Bit	Attr	Reset Value	Description
0	RW	0x0	GB GMII Busy The application sets this bit to instruct the SMA to initiate a Read or Write access to the MDIO slave. The MAC clears this bit after the MDIO frame transfer is completed. Hence the software must not write or change any of the fields in MAC_MDIO_Address and MAC_MDIO_Data registers as long as this bit is set. For write transfers, the application must first write 16-bit data in the GDl field (and also RA field when C45E is set) in MAC_MDIO_Data register before setting this bit. When C45E is set, it should also write into the RA field of MAC_MDIO_Data register before initiating a read transfer. When a read transfer is completed (GB=0), the data read from the PHY register is valid in the GD field of the MAC_MDIO_Data register. Note: Even if the addressed PHY is not present, there is no change in the functionality of this bit. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. Values: 1'b0: GMII Busy is disabled 1'b1: GMII Busy is enabled

GMAC_MAC_MDIO_Data

Address: Operational Base + offset (0x0204)

Bit	Attr	Reset Value	Description
31:17	RO	0x0000	reserved
16	RW	0x0	RA Register Address This field is valid only when C45E is set. It contains the Register Address in the PHY to which the MDIO frame is intended for.
15:0	RW	0x0000	GD GMII Data This field contains the 16-bit data value read from the PHY or RevMII after a Management Read operation or the 16-bit data value to be written to the PHY or RevMII before a Management Write operation.

GMAC_MAC_CSR_SW_Ctrl

Address: Operational Base + offset (0x0230)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RW	0x0	RCWE Register Clear on Write 1 Enable When this bit is set, the access mode of some register fields changes to Clear on Write 1, the application needs to set that respective bit to 1 to clear it. When this bit is reset, the access mode of these register fields remain as Clear on Read. Values: 1'b0: Register Clear on Write 1 is disabled 1'b1: Register Clear on Write 1 is enabled

GMAC_MAC_Address0_High

Address: Operational Base + offset (0x0300)

Bit	Attr	Reset Value	Description
31	RO	0x1	AE Address Enable This bit is always set to 1. Values: 1'b0: This bit must be always set to 1 1'b1: This bit is always set to 1
30:16	RO	0x0000	reserved
15:0	RW	0xffff	ADDRHI MAC Address0[47:32] This field contains the upper 16 bits [47:32] of the first 6-byte MAC address. The MAC uses this field for filtering the received packets and inserting the MAC address in the Transmit Flow Control (Pause) Packets.

GMAC_MAC_Address0_Low

Address: Operational Base + offset (0x0304)

Bit	Attr	Reset Value	Description
31:0	RW	0xffffffff	ADDRLO MAC Address0[31:0] This field contains the lower 32 bits of the first 6-byte MAC address. The MAC uses this field for filtering the received packets and inserting the MAC address in the Transmit Flow Control (Pause) Packets.

GMAC_MMC_Control

Address: Operational Base + offset (0x0700)

Bit	Attr	Reset Value	Description
31:9	RO	0x000000	reserved
8	RW	0x0	UCDBC Update MMC Counters for Dropped Broadcast Packets Note: The CNTRST bit has a higher priority than the CNTPRST bit. Therefore, when the software tries to set both bits in the same write cycle, all counters are cleared and the CNTPRST bit is not set. When set, the MAC updates all related MMC Counters for Broadcast packets that are dropped because of the setting of the DBF bit of MAC_Packet_Filter register. When reset, the MMC Counters are not updated for dropped Broadcast packets. Values: 1'b0: Update MMC Counters for Dropped Broadcast Packets is disabled 1'b1: Update MMC Counters for Dropped Broadcast Packets is enabled
7:6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5	RW	0x0	CNTPRSTLVL Full-Half Preset When this bit is low and the CNTPRST bit is set, all MMC counters get preset to almost-half value. All octet counters get preset to 0x7FFF_F800 (Half 2KBytes) and all packet-counters gets preset to 0x7FFF_FFF0 (Half 16). When this bit is high and the CNTPRST bit is set, all MMC counters get preset to almost-full value. All octet counters get preset to 0xFFFF_F800 (Full 2KBytes) and all packet-counters gets preset to 0xFFFF_FFF0 (Full 16). For 16-bit counters, the almost-half preset values are 0x7800 and 0x7FF0 for the respective octet and packet counters. Similarly, the almost-full preset values for the 16-bit counters are 0xF800 and 0xFFF0. Values: 1'b0: Full-Half Preset is disabled 1'b1: Full-Half Preset is enabled
4	RW	0x0	CNTPRST Counters Preset When this bit is set, all counters are initialized or preset to almost full or almost half according to the CNTPRSTLVL bit. This bit is cleared automatically after 1 clock cycle. This bit, along with the CNTPRSTLVL bit, is useful for debugging and testing the assertion of interrupts because of MMC counter becoming half-full or full. Access restriction applies. Self-cleared. Setting 0 clears. Setting 1 sets. Values: 1'b0: Counters Preset is disabled 1'b1: Counters Preset is enabled
3	RW	0x0	CNTFREEZ MMC Counter Freeze When this bit is set, it freezes all MMC counters to their current value. Until this bit is reset to 0, no MMC counter is updated because of any transmitted or received packet. If any MMC counter is read with the Reset on Read bit set, then that counter is also cleared in this mode. Values: 1'b0: MMC Counter Freeze is disabled 1'b1: MMC Counter Freeze is enabled
2	RW	0x0	RSTONRD Reset on Read When this bit is set, the MMC counters are reset to zero after Read (self-clearing after reset). The counters are cleared when the least significant byte lane (Bits[7:0]) is read. Values: 1'b0: Reset on Read is disabled 1'b1: Reset on Read is enabled
1	RW	0x0	CNTSTOPRO Counter Stop Rollover When this bit is set, the counter does not roll over to zero after reaching the maximum value. Values: 1'b0: Counter Stop Rollover is disabled 1'b1: Counter Stop Rollover is enabled

Bit	Attr	Reset Value	Description
0	RW	0x0	CNTRST Counters Reset When this bit is set, all counters are reset. This bit is cleared automatically after 1 clock cycle. Access restriction applies. Self-cleared. Setting 0 clears. Setting 1 sets. Values: 1'b0: Counters are not reset 1'b1: All counters are reset

GMAC_MMC_Rx_Interrupt

Address: Operational Base + offset (0x0704)

Bit	Attr	Reset Value	Description
31:22	RO	0x000	reserved
21	RO	0x0	RXFOVPIS MMC Receive FIFO Overflow Packet Counter Interrupt Status This bit is set when the rxfifooverflow counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Receive FIFO Overflow Packet Counter Interrupt Status not detected 1'b1: MMC Receive FIFO Overflow Packet Counter Interrupt Status detected
20:19	RO	0x0	reserved
18	RO	0x0	RXLENERPIS MMC Receive Length Error Packet Counter Interrupt Status This bit is set when the rxlengtherror counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Receive Length Error Packet Counter Interrupt Status not detected 1'b1: MMC Receive Length Error Packet Counter Interrupt Status detected
17:6	RO	0x000	reserved
5	RO	0x0	RXCRCERPIS MMC Receive CRC Error Packet Counter Interrupt Status This bit is set when the rxcrcerror counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Receive CRC Error Packet Counter Interrupt Status not detecte 1'b1: MMC Receive CRC Error Packet Counter Interrupt Status detected

Bit	Attr	Reset Value	Description
4	RO	0x0	RXMCGPIS MMC Receive Multicast Good Packet Counter Interrupt Status This bit is set when the rxmulticastpackets_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Receive Multicast Good Packet Counter Interrupt Status not detected 1'b1: MMC Receive Multicast Good Packet Counter Interrupt Status detected
3	RO	0x0	reserved
2	RO	0x0	RXGOCTIS MMC Receive Good Octet Counter Interrupt Status This bit is set when the rxoctetcount_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Receive Good Octet Counter Interrupt Status not detected 1'b1: MMC Receive Good Octet Counter Interrupt Status detected
1	RO	0x0	RXGBOCTIS MMC Receive Good Bad Octet Counter Interrupt Status This bit is set when the rxoctetcount_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Receive Good Bad Octet Counter Interrupt Status not detected 1'b1: MMC Receive Good Bad Octet Counter Interrupt Status detected
0	RO	0x0	RXGBPKTIS MMC Receive Good Bad Packet Counter Interrupt Status This bit is set when the rxpacketcount_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Receive Good Bad Packet Counter Interrupt Status not detected 1'b1: MMC Receive Good Bad Packet Counter Interrupt Status detected

GMAC_MMC_Tx_Interrupt

Address: Operational Base + offset (0x0708)

Bit	Attr	Reset Value	Description
31:22	RO	0x000	reserved

Bit	Attr	Reset Value	Description
21	RO	0x0	TXGPKTIS MMC Transmit Good Packet Counter Interrupt Status This bit is set when the txpacketcount_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Transmit Good Packet Counter Interrupt Status not detected 1'b1: MMC Transmit Good Packet Counter Interrupt Status detected
20	RO	0x0	TXGOCTIS MMC Transmit Good Octet Counter Interrupt Status This bit is set when the txoctetcount_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Transmit Good Octet Counter Interrupt Status not detected 1'b1: MMC Transmit Good Octet Counter Interrupt Status detected
19	RO	0x0	TXCARERPIS MMC Transmit Carrier Error Packet Counter Interrupt Status This bit is set when the txcarriererror counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Transmit Carrier Error Packet Counter Interrupt Status not detected 1'b1: MMC Transmit Carrier Error Packet Counter Interrupt Status detected
18:14	RO	0x00	reserved
13	RO	0x0	TXUFLOWERPIS MMC Transmit Underflow Error Packet Counter Interrupt Status This bit is set when the txunderflowerror counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Transmit Underflow Error Packet Counter Interrupt Status not detected 1'b1: MMC Transmit Underflow Error Packet Counter Interrupt Status detected
12:2	RO	0x000	reserved

Bit	Attr	Reset Value	Description
1	RO	0x0	TXGBPKTIS MMC Transmit Good Bad Packet Counter Interrupt Status This bit is set when the txpacketcount_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Transmit Good Bad Packet Counter Interrupt Status not detected 1'b1: MMC Transmit Good Bad Packet Counter Interrupt Status detected
0	RO	0x0	TXGBOCTIS MMC Transmit Good Bad Octet Counter Interrupt Status This bit is set when the txoctetcount_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Transmit Good Bad Octet Counter Interrupt Status not detected 1'b1: MMC Transmit Good Bad Octet Counter Interrupt Status detected

GMAC_MMC_Rx_Interrupt_Mask

Address: Operational Base + offset (0x070c)

Bit	Attr	Reset Value	Description
31:22	RO	0x000	reserved
21	RW	0x0	RXFOVPIM MMC Receive FIFO Overflow Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxfifooverflow counter reaches half of the maximum value or the maximum value. Values: 1'b0: MMC Receive FIFO Overflow Packet Counter Interrupt Mask is disabled 1'b1: MMC Receive FIFO Overflow Packet Counter Interrupt Mask is enabled
20:19	RO	0x0	reserved
18	RW	0x0	RXLENERPIM MMC Receive Length Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxlengtherror counter reaches half of the maximum value or the maximum value. Values: 1'b0: MMC Receive Length Error Packet Counter Interrupt Mask is disabled 1'b1: MMC Receive Length Error Packet Counter Interrupt Mask is enabled
17:6	RO	0x000	reserved

Bit	Attr	Reset Value	Description
5	RW	0x0	RXCRCERPIM MMC Receive CRC Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxcrcerror counter reaches half of the maximum value or the maximum value. Values: 1'b0: MMC Receive CRC Error Packet Counter Interrupt Mask is disabled 1'b1: MMC Receive CRC Error Packet Counter Interrupt Mask is enabled
4	RW	0x0	RXMCGPIM MMC Receive Multicast Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxmulticastpackets_g counter reaches half of the maximum value or the maximum value. Values: 1'b0: MMC Receive Multicast Good Packet Counter Interrupt Mask is disabled 1'b1: MMC Receive Multicast Good Packet Counter Interrupt Mask is enabled
3	RO	0x0	reserved
2	RW	0x0	RXGOCTIM MMC Receive Good Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxoctetcount_g counter reaches half of the maximum value or the maximum value. Values: 1'b0: MMC Receive Good Octet Counter Interrupt Mask is disabled 1'b1: MMC Receive Good Octet Counter Interrupt Mask is enabled
1	RW	0x0	RXGBOCTIM MMC Receive Good Bad Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxoctetcount_gb counter reaches half of the maximum value or the maximum value. Values: 1'b0: MMC Receive Good Bad Octet Counter Interrupt Mask is disabled 1'b1: MMC Receive Good Bad Octet Counter Interrupt Mask is enabled
0	RW	0x0	RXGBPKTIM MMC Receive Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxpacketcount_gb counter reaches half of the maximum value or the maximum value. Values: 1'b0: MMC Receive Good Bad Packet Counter Interrupt Mask is disabled 1'b1: MMC Receive Good Bad Packet Counter Interrupt Mask is enabled

GMAC_MMC_Tx_Interrupt_Mask

Address: Operational Base + offset (0x0710)

Bit	Attr	Reset Value	Description
31:22	RO	0x000	reserved

Bit	Attr	Reset Value	Description
21	RW	0x0	TXGPKTIM MMC Transmit Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txpacketcount_g counter reaches half of the maximum value or the maximum value. Values: 1'b0: MMC Transmit Good Packet Counter Interrupt Mask is disabled 1'b1: MMC Transmit Good Packet Counter Interrupt Mask is enabled
20	RW	0x0	TXGOCTIM MMC Transmit Good Octet Counter Interrupt Mask Setting this bit masks the interrupt when the txoctetcount_g counter reaches half of the maximum value or the maximum value. Values: 1'b0: MMC Transmit Good Octet Counter Interrupt Mask is disabled 1'b1: MMC Transmit Good Octet Counter Interrupt Mask is enabled
19	RW	0x0	TXCARERPIM MMC Transmit Carrier Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txcarriererror counter reaches half of the maximum value or the maximum value. Values: 1'b0: MMC Transmit Carrier Error Packet Counter Interrupt Mask is disabled 1'b1: MMC Transmit Carrier Error Packet Counter Interrupt Mask is enabled
18:14	RO	0x00	reserved
13	RW	0x0	TXUFLOWERPIM MMC Transmit Underflow Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txunderflowerror counter reaches half of the maximum value or the maximum value. Values: 1'b0: MMC Transmit Underflow Error Packet Counter Interrupt Mask is disabled 1'b1: MMC Transmit Underflow Error Packet Counter Interrupt Mask is enabled
12:2	RO	0x000	reserved
1	RW	0x0	TXGBPKTIM MMC Transmit Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txpacketcount_gb counter reaches half of the maximum value or the maximum value. Values: 1'b0: MMC Transmit Good Bad Packet Counter Interrupt Mask is disabled 1'b1: MMC Transmit Good Bad Packet Counter Interrupt Mask is enabled

Bit	Attr	Reset Value	Description
0	RW	0x0	TXGBOCTIM MMC Transmit Good Bad Octet Counter Interrupt Mask Setting this bit masks the interrupt when the txoctetcount_gb counter reaches half of the maximum value or the maximum value. Values: 1'b0: MMC Transmit Good Bad Octet Counter Interrupt Mask is disabled 1'b1: MMC Transmit Good Bad Octet Counter Interrupt Mask is enabled

GMAC_Tx_Octet_Count_Good_Bad

Address: Operational Base + offset (0x0714)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	TXOCTGB Tx Octet Count Good Bad This field indicates the number of bytes transmitted, exclusive of preamble and retried bytes, in good and bad packets.

GMAC_Tx_Packet_Count_Good_Bad

Address: Operational Base + offset (0x0718)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	TXPKTGB Tx Packet Count Good Bad This field indicates the number of good and bad packets transmitted, exclusive of retried packets.

GMAC_Tx_Underflow_Error_Packets

Address: Operational Base + offset (0x0748)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	TXUNDRFLW Tx Underflow Error Packets This field indicates the number of packets aborted because of packets underflow error.

GMAC_Tx_Carrier_Error_Packets

Address: Operational Base + offset (0x0760)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	TXCARR Tx Carrier Error Packets This field indicates the number of packets aborted because of carrier sense error (no carrier or loss of carrier).

GMAC_Tx_Octet_Count_Good

Address: Operational Base + offset (0x0764)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	TXOCTG Tx Octet Count Good This field indicates the number of bytes transmitted, exclusive of preamble, only in good packets.

GMAC_Tx_Packet_Count_Good

Address: Operational Base + offset (0x0768)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	TXPKTG Tx Packet Count Good This field indicates the number of good packets transmitted.

GMAC_Rx_Packets_Count_Good_Bad

Address: Operational Base + offset (0x0780)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXPKTGB Rx Packets Count Good Bad This field indicates the number of good and bad packets received.

GMAC_Rx_Octet_Count_Good_Bad

Address: Operational Base + offset (0x0784)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXOCTGB Rx Octet Count Good Bad This field indicates the number of bytes received, exclusive of preamble, in good and bad packets.

GMAC_Rx_Octet_Count_Good

Address: Operational Base + offset (0x0788)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXOCTG Rx Octet Count Good This field indicates the number of bytes received, exclusive of preamble, only in good packets.

GMAC_Rx_Multicast_Packets_Good

Address: Operational Base + offset (0x0790)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXMCASTG Rx Multicast Packets Good This field indicates the number of good multicast packets received.

GMAC_Rx_CRC_Error_Packets

Address: Operational Base + offset (0x0794)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXCRCERR Rx CRC Error Packets This field indicates the number of packets received with CRC error.

GMAC_Rx_Length_Error_Packets

Address: Operational Base + offset (0x07c8)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXLENERR Rx Length Error Packets This field indicates the number of packets received with length error (Length Type field not equal to packet size), for all packets with valid length field.

GMAC_Rx_FIFO_Overflow_Packets

Address: Operational Base + offset (0x07d4)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXFIFOOVFL Rx FIFO Overflow Packets This field indicates the number of missed received packets because of FIFO overflow.

GMAC_MMC_IPC_Rx_Interrupt_Mask

Address: Operational Base + offset (0x0800)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29	RW	0x0	RXICMPEROIM MMC Receive ICMP Error Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxicmp_err_octets counter reaches half of the maximum value or the maximum value. Values: 1'b0: MMC Receive ICMP Error Octet Counter Interrupt Mask is disabled 1'b1: MMC Receive ICMP Error Octet Counter Interrupt Mask is enabled
28	RO	0x0	reserved
27	RW	0x0	RXTCPEROIM MMC Receive TCP Error Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxtcp_err_octets counter reaches half of the maximum value or the maximum value. Values: 1'b0: MMC Receive TCP Error Octet Counter Interrupt Mask is disabled 1'b1: MMC Receive TCP Error Octet Counter Interrupt Mask is enabled
26	RO	0x0	reserved
25	RW	0x0	RXUDPEROIM MMC Receive UDP Error Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxudp_err_octets counter reaches half of the maximum value or the maximum value. Values: 1'b0: MMC Receive UDP Error Octet Counter Interrupt Mask is disabled 1'b1: MMC Receive UDP Error Octet Counter Interrupt Mask is enabled
24:23	RO	0x0	reserved
22	RW	0x0	RXIPV6HEROIM MMC Receive IPV6 Header Error Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv6_nopay_octets counter reaches half of the maximum value or the maximum value. Value: 1'b0: MMC Receive IPV6 Header Error Octet Counter Interrupt Mask is disabled 1'b1: MMC Receive IPV6 Header Error Octet Counter Interrupt Mask is enabled
21:18	RO	0x0	reserved

Bit	Attr	Reset Value	Description
17	RW	0x0	RXIPV4HEROIM MMC Receive IPV4 Header Error Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv4_hdrerr_octets counter reaches half of the maximum value or the maximum value. Values: 1'b0: MMC Receive IPV4 Header Error Octet Counter Interrupt Mask is disabled 1'b1: MMC Receive IPV4 Header Error Octet Counter Interrupt Mask is enabled
16:14	RO	0x0	reserved
13	RW	0x0	RXICMPERPIM MMC Receive ICMP Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxicmp_err_pkts counter reaches half of the maximum value or the maximum value. Values: 1'b0: MMC Receive ICMP Error Packet Counter Interrupt Mask is disabled 1'b1: MMC Receive ICMP Error Packet Counter Interrupt Mask is enabled
12	RO	0x0	reserved
11	RW	0x0	RXTCPERPIM MMC Receive TCP Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxtcp_err_pkts counter reaches half of the maximum value or the maximum value. Values: 1'b0: MMC Receive TCP Error Packet Counter Interrupt Mask is disabled 1'b1: MMC Receive TCP Error Packet Counter Interrupt Mask is enabled
10	RO	0x0	reserved
9	RW	0x0	RXUDPERPIM MMC Receive UDP Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxudp_err_pkts counter reaches half of the maximum value or the maximum value. Values: 1'b0: MMC Receive UDP Error Packet Counter Interrupt Mask is disabled 1'b1: MMC Receive UDP Error Packet Counter Interrupt Mask is enabled
8:7	RO	0x0	reserved
6	RW	0x0	RXIPV6HERPIM MMC Receive IPV6 Header Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv6_hdrerr_pkts counter reaches half of the maximum value or the maximum value. Values: 1'b0: MMC Receive IPV6 Header Error Packet Counter Interrupt Mask is disabled 1'b1: MMC Receive IPV6 Header Error Packet Counter Interrupt Mask is enabled

Bit	Attr	Reset Value	Description
5	RW	0x0	RXIPV6GPIM MMC Receive IPV6 Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv6_gd_pkts counter reaches half of the maximum value or the maximum value. Values: 1'b0: MMC Receive IPV6 Good Packet Counter Interrupt Mask is disabled 1'b1: MMC Receive IPV6 Good Packet Counter Interrupt Mask is enabled
4:2	RO	0x0	reserved
1	RW	0x0	RXIPV4HERPIM MMC Receive IPV4 Header Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv4_hdrerr_pkts counter reaches half of the maximum value or the maximum value. Values: 1'b0: MMC Receive IPV4 Header Error Packet Counter Interrupt Mask is disabled 1'b1: MMC Receive IPV4 Header Error Packet Counter Interrupt Mask is enabled
0	RW	0x0	RXIPV4GPIM MMC Receive IPV4 Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv4_gd_pkts counter reaches half of the maximum value or the maximum value. Values: 1'b0: MMC Receive IPV4 Good Packet Counter Interrupt Mask is disable 1'b1: MMC Receive IPV4 Good Packet Counter Interrupt Mask is enabled

GMAC_MMC_IPC_Rx_Interrupt

Address: Operational Base + offset (0x0808)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29	RO	0x0	RXICMPEROIS MMC Receive ICMP Error Octet Counter Interrupt Status This bit is set when the rxicmp_err_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Receive ICMP Error Octet Counter Interrupt Status not detected 1'b1: MMC Receive ICMP Error Octet Counter Interrupt Status detected
28	RO	0x0	reserved

Bit	Attr	Reset Value	Description
27	RO	0x0	RXTCPEROIS MMC Receive TCP Error Octet Counter Interrupt Status This bit is set when the rxtcp_err_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Receive TCP Error Octet Counter Interrupt Status not detected 1'b1: MMC Receive TCP Error Octet Counter Interrupt Status detected
26	RO	0x0	reserved
25	RO	0x0	RXUDPEROIS MMC Receive UDP Error Octet Counter Interrupt Status This bit is set when the rxudp_err_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Receive UDP Error Octet Counter Interrupt Status not detected 1'b1: MMC Receive UDP Error Octet Counter Interrupt Status detected
24:23	RO	0x0	reserved
22	RO	0x0	RXIPV6HEROIS MMC Receive IPV6 Header Error Octet Counter Interrupt Status This bit is set when the rxipv6_hdrerr_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Receive IPV6 Header Error Octet Counter Interrupt Status not detected 1'b1: MMC Receive IPV6 Header Error Octet Counter Interrupt Status detected
21:18	RO	0x0	reserved
17	RO	0x0	RXIPV4HEROIS MMC Receive IPV4 Header Error Octet Counter Interrupt Status This bit is set when the rxipv4_hdrerr_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Receive IPV4 Header Error Octet Counter Interrupt Status not detected 1'b1: MMC Receive IPV4 Header Error Octet Counter Interrupt Status detected
16:14	RO	0x0	reserved

Bit	Attr	Reset Value	Description
13	RO	0x0	RXICMPERPIS MMC Receive ICMP Error Packet Counter Interrupt Status This bit is set when the rxicmp_err_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Receive ICMP Error Packet Counter Interrupt Status not detected 1'b1: MMC Receive ICMP Error Packet Counter Interrupt Status detected
12	RO	0x0	reserved
11	RO	0x0	RXTCPERPIS MMC Receive TCP Error Packet Counter Interrupt Status This bit is set when the rxtcp_err_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Receive TCP Error Packet Counter Interrupt Status not detected 1'b1: MMC Receive TCP Error Packet Counter Interrupt Status detected
10	RO	0x0	reserved
9	RO	0x0	RXUDPERPIS MMC Receive UDP Error Packet Counter Interrupt Status This bit is set when the rxudp_err_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Receive UDP Error Packet Counter Interrupt Status not detected 1'b1: MMC Receive UDP Error Packet Counter Interrupt Status detected
8:7	RO	0x0	reserved
6	RO	0x0	RXIPV6HERPIS MMC Receive IPV6 Header Error Packet Counter Interrupt Status This bit is set when the rxipv6_hdrerr_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Receive IPV6 Header Error Packet Counter Interrupt Status not detected 1'b1: MMC Receive IPV6 Header Error Packet Counter Interrupt Status detected

Bit	Attr	Reset Value	Description
5	RO	0x0	RXIPV6GPIS MMC Receive IPV6 Good Packet Counter Interrupt Status This bit is set when the rxipv6_gd_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Receive IPV6 Good Packet Counter Interrupt Status not detected 1'b1: MMC Receive IPV6 Good Packet Counter Interrupt Status detected
4:2	RO	0x0	reserved
1	RO	0x0	RXIPV4HERPIS MMC Receive IPV4 Header Error Packet Counter Interrupt Status This bit is set when the rxipv4_hdrerr_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Receive IPV4 Header Error Packet Counter Interrupt Status not detected 1'b1: MMC Receive IPV4 Header Error Packet Counter Interrupt Status detected
0	RO	0x0	RXIPV4GPIS MMC Receive IPV4 Good Packet Counter Interrupt Status This bit is set when the rxipv4_gd_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Receive IPV4 Good Packet Counter Interrupt Status not detected 1'b1: MMC Receive IPV4 Good Packet Counter Interrupt Status detected

GMAC_RxIPv4_Good_Packets

Address: Operational Base + offset (0x0810)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXIPV4GDPKT RxIPv4 Good Packets This field indicates the number of good IPv4 datagrams received with the TCP, UDP, or ICMP payload.

GMAC_RxIPv4_Header_Error_Packets

Address: Operational Base + offset (0x0814)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXIPV4HDRERRPKT RxIPv4 Header Error Packets This field indicates the number of IPv4 datagrams received with header (checksum, length, or version mismatch) errors.

GMAC_RxIPv6_Good_Packets

Address: Operational Base + offset (0x0824)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXIPV6GDPKT RxIPv6 Good Packets This field indicates the number of good IPv6 datagrams received with the TCP, UDP, or ICMP payload.

GMAC_RxIPv6_Header_Error_Packets

Address: Operational Base + offset (0x0828)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXIPV6HDRERRPKT RxIPv6 Header Error Packets This field indicates the number of IPv6 datagrams received with header (length or version mismatch) errors.

GMAC_RxUDP_Error_Packets

Address: Operational Base + offset (0x0834)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXUDPERRPKT RxUDP Error Packets This field indicates the number of good IP datagrams received whose UDP payload has a checksum error.

GMAC_RxTCP_Error_Packets

Address: Operational Base + offset (0x083c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXTCPERRPKT RxTCP Error Packets This field indicates the number of good IP datagrams received whose TCP payload has a checksum error.

GMAC_RxICMP_Error_Packets

Address: Operational Base + offset (0x0844)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXICMPERRPKT RxICMP Error Packets This field indicates the number of good IP datagrams received whose ICMP payload has a checksum error.

GMAC_RxIPv4_Header_Error_Octets

Address: Operational Base + offset (0x0854)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXIPV4HDRERROCT RxIPv4 Header Error Octets This field indicates the number of bytes received in IPv4 datagrams with header errors (checksum, length, version mismatch). The value in the Length field of IPv4 header is used to update this counter. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

GMAC_RxIPv6_Header_Error_Octets

Address: Operational Base + offset (0x0868)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXIPV6HDRERROCT RxIPv6 Header Error Octets This field indicates the number of bytes received in IPv6 datagrams with header errors (length, version mismatch). The value in the Length field of IPv6 header is used to update this counter. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

GMAC_RxUDP_Error_Octets

Address: Operational Base + offset (0x0874)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXUDPERROCT RxUDP Error Octets This field indicates the number of bytes received in a UDP segment that had checksum errors. This counter does not count IP header bytes.

GMAC_RxTCP_Error_Octets

Address: Operational Base + offset (0x087c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXTCPERROCT RxTCP Error Octets This field indicates the number of bytes received in a TCP segment that had checksum errors. This counter does not count IP header bytes.

GMAC_RxICMP_Error_Octets

Address: Operational Base + offset (0x0884)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXICMPERROCT RxICMP Error Octets This field indicates the number of bytes received in a ICMP segment that had checksum errors. This counter does not count IP header bytes.

GMAC_MAC_Timestamp_Control

Address: Operational Base + offset (0x0b00)

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28	RW	0x0	AV8021ASMEN AV 802.1AS Mode Enable When this bit is set, the MAC processes only untagged PTP over Ethernet packets for providing PTP status and capturing timestamp snapshots, that is, IEEE 802.1AS mode of operation. When PTP offload feature is enabled, for the purpose of PTP offload, the transport specific field in the PTP header is generated and checked based on the value of this bit. Values: 1'b0: AV 802.1AS Mode is disabled 1'b1: AV 802.1AS Mode is enabled
27:25	RO	0x0	reserved

Bit	Attr	Reset Value	Description
24	RW	0x0	TXTSSTSM Transmit Timestamp Status Mode When this bit is set, the MAC overwrites the earlier transmit timestamp status even if it is not read by the software. The MAC indicates this by setting the TXTSSMIS bit of the MAC_Tx_Timestamp_Status_Nanoseconds register. When this bit is reset, the MAC ignores the timestamp status of current packet if the timestamp status of previous packet is not read by the software. The MAC indicates this by setting the TXTSSMIS bit of the MAC_Tx_Timestamp_Status_Nanoseconds register. Values: 1'b0: Transmit Timestamp Status Mode is disabled 1'b1: Transmit Timestamp Status Mode is enabled
23:21	RO	0x0	reserved
20	RW	0x0	ESTI External System Time Input When this bit is set, the MAC uses the external 64-bit reference System Time input for the following: 1. To take the timestamp provided as status 2. To insert the timestamp in transmit PTP packets when One-step Timestamp or Timestamp Offload feature is enabled. When this bit is reset, the MAC uses the internal reference System Time. Values: 1'b0: External System Time Input is disabled 1'b1: External System Time Input is enabled
19	RO	0x0	reserved
18	RW	0x0	TSENMACADDR Enable MAC Address for PTP Packet Filtering When this bit is set, the DA MAC address (that matches any MAC Address register) is used to filter the PTP packets when PTP is directly sent over Ethernet. When this bit is set, received PTP packets with DA containing a special multicast or unicast address that matches the one programmed in MAC address registers are considered for processing as indicated below, when PTP is directly sent over Ethernet. For normal time stamping operation, MAC address registers 0 to 31 is considered for unicast destination address matching. For PTP offload, only MAC address register 0 is considered for unicast destination address matching. Values: 1'b0: MAC Address for PTP Packet Filtering is disabled 1'b1: MAC Address for PTP Packet Filtering is enabled
17:16	RW	0x0	SNAPTYPSEL Select PTP packets for Taking Snapshots These bits, along with Bits 15 and 14, decide the set of PTP packet types for which snapshot needs to be taken. The encoding is given in Timestamp Snapshot Dependency on Register Bits Table.

Bit	Attr	Reset Value	Description
15	RW	0x0	TSMSTRENA Enable Snapshot for Messages Relevant to Master When this bit is set, the snapshot is taken only for the messages that are relevant to the master node. Otherwise, the snapshot is taken for the messages relevant to the slave node. Values: 1'b0: Snapshot for Messages Relevant to Master is disabled 1'b1: Snapshot for Messages Relevant to Master is enabled
14	RW	0x0	TSEVNTENA Enable Timestamp Snapshot for Event Messages When this bit is set, the timestamp snapshot is taken only for event messages (SYNC, Delay_Req, Pdelay_Req, or Pdelay_Resp). When this bit is reset, the snapshot is taken for all messages except Announce, Management, and Signaling. For more information about the timestamp snapshots, see Timestamp Snapshot Dependency on Register Bits Table. Values: 1'b0: Timestamp Snapshot for Event Messages is disabled 1'b1: Timestamp Snapshot for Event Messages is enabled
13	RW	0x1	TSIPV4ENA Enable Processing of PTP Packets Sent over IPv4-UDP When this bit is set, the MAC receiver processes the PTP packets encapsulated in IPv4-UDP packets. When this bit is reset, the MAC ignores the PTP transported over IPv4-UDP packets. This bit is set by default. Values: 1'b0: Processing of PTP Packets Sent over IPv4-UDP is disabled 1'b1: Processing of PTP Packets Sent over IPv4-UDP is enabled
12	RW	0x0	TSIPV6ENA Enable Processing of PTP Packets Sent over IPv6-UDP When this bit is set, the MAC receiver processes the PTP packets encapsulated in IPv6-UDP packets. When this bit is clear, the MAC ignores the PTP transported over IPv6-UDP packets. Values: 1'b0: Processing of PTP Packets Sent over IPv6-UDP is disabled 1'b1: Processing of PTP Packets Sent over IPv6-UDP is enabled
11	RW	0x0	TSIPENA Enable Processing of PTP over Ethernet Packets When this bit is set, the MAC receiver processes the PTP packets encapsulated directly in the Ethernet packets. When this bit is reset, the MAC ignores the PTP over Ethernet packets. Values: 1'b0: Processing of PTP over Ethernet Packets is disabled 1'b1: Processing of PTP over Ethernet Packets is enabled
10	RW	0x0	TSVER2ENA Enable PTP Packet Processing for Version 2 Format When this bit is set, the IEEE 1588 version 2 format is used to process the PTP packets. When this bit is reset, the IEEE 1588 version 1 format is used to process the PTP packets. The IEEE 1588 formats are described in 'PTP Processing and Control'. Values: 1'b0: PTP Packet Processing for Version 2 Format is disabled 1'b1: PTP Packet Processing for Version 2 Format is enabled

Bit	Attr	Reset Value	Description
9	RW	0x0	TSCTRLSSR Timestamp Digital or Binary Rollover Control When this bit is set, the Timestamp Low register rolls over after 0x3B9A_C9FF value (that is, 1 nanosecond accuracy) and increments the timestamp (High) seconds. When this bit is reset, the rollover value of sub-second register is 0x7FFF_FFFF. The sub-second increment must be programmed correctly depending on the PTP reference clock frequency and the value of this bit. Values: 1'b0: Timestamp Digital or Binary Rollover Control is disabled 1'b1: Timestamp Digital or Binary Rollover Control is enabled
8	RW	0x0	TSENALL Enable Timestamp for All Packets When this bit is set, the timestamp snapshot is enabled for all packets received by the MAC. Values: 1'b0: Timestamp for All Packets disabled 1'b1: Timestamp for All Packets enabled
7:6	RO	0x0	reserved
5	RW	0x0	TSADDREG Update Addend Register When this bit is set, the content of the Timestamp Addend register is updated in the PTP block for fine correction. This bit is cleared when the update is complete. This bit should be zero before it is set. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. Values: 1'b0: Addend Register is not updated 1'b1: Addend Register is updated
4	RW	0x0	TSTRIG Enable Timestamp Interrupt Trigger When this bit is set, the timestamp interrupt is generated when the System Time becomes greater than the value written in the Target Time register. This bit is reset after the Timestamp Trigger Interrupt is generated. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. Values: 1'b0: Timestamp Interrupt Trigger is not enabled 1'b1: Timestamp Interrupt Trigger is enabled

Bit	Attr	Reset Value	Description
3	RW	0x0	TSUPDT Update Timestamp When this bit is set, the system time is updated (added or subtracted) with the value specified in MAC_System_Time_Seconds_Update and MAC_System_Time_Nanoseconds_Update registers. This bit should be zero before updating it. This bit is reset when the update is complete in hardware. The Timestamp Higher Word register (if enabled during core configuration) is not updated.When Media Clock Generation and Recovery is configured (DWC_EQOS_FLEXI_PPS_OUT_EN) and enabled MAC_Presn_Time_Updt should also be updated before setting this field. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. Values: 1'b0: Timestamp is not updated 1'b1: Timestamp is updated
2	RW	0x0	TSINIT Initalize Timestamp When this bit is set, the system time is initialized (overwritten) with the value specified in the MAC_System_Time_Seconds_Update and MAC_System_Time_Nanoseconds_Update registers. This bit should be zero before it is updated. This bit is reset when the initialization is complete. The Timestamp Higher Word register (if enabled during core configuration) can only be initialized.When Media Clock Generation and Recovery is configured (DWC_EQOS_FLEXI_PPS_OUT_EN) and enabled MAC_Presn_Time_Updt should also be updated before setting this field. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. Values: 1'b0: Timestamp is not initialized 1'b1: Timestamp is initialized
1	RW	0x0	TSCFUPDT Fine or Coarse Timestamp Update When this bit is set, the Fine method is used to update system timestamp. When this bit is reset, Coarse method is used to update the system timestamp. Values: 1'b0: Coarse method is used to update system timestamp 1'b1: Fine method is used to update system timestamp
0	RW	0x0	TSENA Enable Timestamp When this bit is set, the timestamp is added for Transmit and Receive packets. When disabled, timestamp is not added for transmit and receive packets and the Timestamp Generator is also suspended. You need to initialize the Timestamp (system time) after enabling this mode.On the Receive side, the MAC processes the 1588 packets only if this bit is set. Values: 1'b0: Timestamp is disabled 1'b1: Timestamp is enabled

GMAC_MAC_Sub_Second_Increment

Address: Operational Base + offset (0x0b04)

Bit	Attr	Reset Value	Description
31:24	RO	0x00	reserved
23:16	RW	0x00	SSINC Sub-second Increment Value The value programmed in this field is accumulated every clock cycle (of clk_ptp_i) with the contents of the sub-second register. For example, when the PTP clock is 50 MHz (period is 20 ns), you should program 20 (0x14) when the System Time Nanoseconds register has an accuracy of 1 ns [Bit 9 (TSCTRLSSR) is set in MAC_Timestamp_Control]. When TSCTRLSSR is clear, the Nanoseconds register has a resolution of ~0.465 ns. In this case, you should program a value of 43 (0x2B) which is derived by 20 ns/0.465.

GMAC_MAC_System_Time_Secs

Address: Operational Base + offset (0x0b08)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	TSS Timestamp Second The value in this field indicates the current value in seconds of the System Time maintained by the MAC.

GMAC_MAC_System_Time_NS

Address: Operational Base + offset (0x0b0c)

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30:0	RW	0x00000000	TSSS Timestamp Sub Seconds The value in this field has the sub-second representation of time, with an accuracy of 0.46 ns. When Bit 9 is set in MAC_Timestamp_Control, each bit represents 1 ns. The maximum value is 0x3B9A_C9FF after which it rolls-over to zero.

GMAC_MAC_Sys_Time_Secs_Update

Address: Operational Base + offset (0x0b10)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	TSS Timestamp Seconds The value in this field is the seconds part of the update. When ADDSUB is reset, this field must be programmed with the seconds part of the update value. When ADDSUB is set, this field must be programmed with the complement of the seconds part of the update value. For example, if 2.000000001 seconds need to be subtracted from the system time, the TSS field in the MAC_Timestamp_Seconds_Update register must be 0xFFFF_FFFE (that is, 2^32 - 2).

GMAC_MAC_Sys_Time_NS_Update

Address: Operational Base + offset (0x0b14)

Bit	Attr	Reset Value	Description
31	RW	0x0	ADDSUB Add or Subtract Time When this bit is set, the time value is subtracted with the contents of the update register. When this bit is reset, the time value is added with the contents of the update register. Values: 1'b0: Add time 1'b1: Subtract time
30:0	RW	0x00000000	TSSS Timestamp Sub Seconds The value in this field is the sub-seconds part of the update. When ADDSUB is reset, this field must be programmed with the sub-seconds part of the update value, with an accuracy based on the TSCTRLSSR bit of the MAC_Timestamp_Control register.When ADDSUB is set, this field must be programmed with the complement of the sub-seconds part of the update value as described below. When TSCTRLSSR bit in MAC_Timestamp_Control is set, the programmed value must be 10^9 - <sub-second value>. When TSCTRLSSR bit in MAC_Timestamp_Control is reset, the programmed value must be 2^31 - <sub-second_value>. When the TSCTRLSSR bit is reset in the MAC_Timestamp_Control register, each bit represents an accuracy of 0.46 ns. When the TSCTRLSSR bit is set in the MAC_Timestamp_Control register, each bit represents 1 ns and the programmed value should not exceed 0x3B9A_C9FF. For example, if 2.000000001 seconds need to be subtracted from the system time, then the TSSS field in the MAC_Timestamp_Nanoseconds_Update register must be 0x7FFF_FFFF (that is, 2^31 - 1), when TSCTRLSSR bit in MAC_Timestamp_Control is reset and 0x3B9A_C9FF (that is, 10^9 - 1), when TSCTRLSSR bit in MAC_Timestamp_Control is set.

GMAC_MAC_Timestamp_Addend

Address: Operational Base + offset (0x0b18)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	TSAR Timestamp Addend Register This field indicates the 32-bit time value to be added to the Accumulator register to achieve time synchronization.

GMAC_MAC_Timestamp_Status

Address: Operational Base + offset (0x0b20)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:25	RO	0x00	ATSNS Number of Auxiliary Timestamp Snapshots This field indicates the number of Snapshots available in the FIFO. A value equal to the selected depth of FIFO (4, 8, or 16) indicates that the Auxiliary Snapshot FIFO is full. These bits are cleared (to 00000) when the Auxiliary snapshot FIFO clear bit is set. This bit is valid only if the Add IEEE 1588 Auxiliary Snapshot option is selected.

Bit	Attr	Reset Value	Description
24	RO	0x0	ATSSTM Auxiliary Timestamp Snapshot Trigger Missed This bit is set when the Auxiliary timestamp snapshot FIFO is full and external trigger was set. This indicates that the latest snapshot is not stored in the FIFO. This bit is valid only if the Add IEEE 1588 Auxiliary Snapshot option is selected. Values: 1'b0: Auxiliary Timestamp Snapshot Trigger Missed status not detected 1'b1: Auxiliary Timestamp Snapshot Trigger Missed status detected
23:16	RO	0x00	reserved
15	RO	0x0	TXTSSIS Tx Timestamp Status Interrupt Status In non-EQOS_CORE configurations when drop transmit status is enabled in MTL, this bit is set when the captured transmit timestamp is updated in the MAC_Tx_Timestamp_Status_Nanoseconds and MAC_Tx_Timestamp_Status_Seconds registers. When PTP offload feature is enabled, this bit is set when the captured transmit timestamp is updated in the MAC_Tx_Timestamp_Status_Nanoseconds and MAC_Tx_Timestamp_Status_Seconds registers, for PTO generated Delay Request and Pdelay request packets. This bit is cleared when the MAC_Tx_Timestamp_Status_Seconds register is read (or write to MAC_Tx_Timestamp_Status_Seconds register when RCWE bit of MAC_CSR_SW_Ctrl register is set). Values: 1'b0: Tx Timestamp Status Interrupt status not detected 1'b1: Tx Timestamp Status Interrupt status detected
14:4	RO	0x000	reserved
3	RO	0x0	TSTRGTERR0 Timestamp Target Time Error This bit is set when the latest target time programmed in the MAC_PPS0_Target_Time_Seconds and MAC_PPS0_Target_Time_Nanoseconds registers elapses. This bit is cleared when the application reads this bit. Access restriction applies. Clears on read (or this bit is written to 1 when RCWE bit in MAC_CSR_SW_Ctrl register is set). Self-set to 1 on internal event. Values: 1'b0: Timestamp Target Time Error status not detected 1'b1: Timestamp Target Time Error status detected
2	RO	0x0	AUXTSTRIG Auxiliary Timestamp Trigger Snapshot This bit is set high when the auxiliary snapshot is written to the FIFO. This bit is valid only if the Add IEEE 1588 Auxiliary Snapshot option is selected. Access restriction applies. Clears on read (or this bit is written to 1 when RCWE bit in MAC_CSR_SW_Ctrl register is set). Self-set to 1 on internal event. Values: 1'b0: Auxiliary Timestamp Trigger Snapshot status not detected 1'b1: Auxiliary Timestamp Trigger Snapshot status detected

Bit	Attr	Reset Value	Description
1	RO	0x0	TSTARGT0 Timestamp Target Time Reached When set, this bit indicates that the value of system time is greater than or equal to the value specified in the MAC_PPS0_Target_Time_Seconds and MAC_PPS0_Target_Time_Nanoseconds registers. Access restriction applies. Clears on read (or this bit is written to 1 when RCWE bit in MAC_CSR_SW_Ctrl register is set). Self-set to 1 on internal event. Values: 1'b0: Timestamp Target Time Reached status not detected 1'b1: Timestamp Target Time Reached status detected
0	RO	0x0	TSSOVF Timestamp Seconds Overflow When this bit is set, it indicates that the seconds value of the timestamp (when supporting version 2 format) has overflowed beyond 32'hFFFF_FFFF. Access restriction applies. Clears on read (or this bit is written to 1 when RCWE bit in MAC_CSR_SW_Ctrl register is set). Self-set to 1 on internal event. Values: 1'b0: Timestamp Seconds Overflow status not detected 1'b1: Timestamp Seconds Overflow status detected

GMAC_MAC_Tx_TS_Status_NS

Address: Operational Base + offset (0x0b30)

Bit	Attr	Reset Value	Description
31	RO	0x0	TXTSSMIS Transmit Timestamp Status Missed When this bit is set, it indicates one of the following: 1. The timestamp of the current packet is ignored if TXTSSTSM bit of the MAC_Timestamp_Control register is reset. 2. The timestamp of the previous packet is overwritten with timestamp of the current packet if TXTSSTSM bit of the MAC_Timestamp_Control register is set. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: Transmit Timestamp Status Missed status not detected 1'b1: Transmit Timestamp Status Missed status detected
30:0	RO	0x00000000	TXTSSLO Transmit Timestamp Status Low This field contains the 31 bits of the Nanoseconds field of the Transmit packet's captured timestamp.

GMAC_MAC_Tx_TS_Status_Secs

Address: Operational Base + offset (0x0b34)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	TXTSSHI Transmit Timestamp Status High This field contains the lower 32 bits of the Seconds field of Transmit packet's captured timestamp.

GMAC_MAC_Auxiliary_Control

Address: Operational Base + offset (0x0b40)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RW	0x0	ATSFC Auxiliary Snapshot FIFO Clear When set, this bit resets the pointers of the Auxiliary Snapshot FIFO. This bit is cleared when the pointers are reset and the FIFO is empty. When this bit is high, the auxiliary snapshots are stored in the FIFO. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. Values: 1'b0: Auxiliary Snapshot FIFO Clear is disabled 1'b1: Auxiliary Snapshot FIFO Clear is enabled

GMAC_MAC_Auxiliary_TS_NS

Address: Operational Base + offset (0x0b48)

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30:0	RO	0x00000000	AUXTSLO Auxiliary Timestamp Contains the lower 31 bits (nanoseconds field) of the auxiliary timestamp.

GMAC_MAC_Auxiliary_TS_Secs

Address: Operational Base + offset (0x0b4c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	AUXTSHI Auxiliary Timestamp Contains the lower 32 bits of the Seconds field of the auxiliary timestamp.

GMAC_MAC_TS_Ingress_Corr_NS

Address: Operational Base + offset (0x0b58)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	TSIC Timestamp Ingress Correction This field contains the ingress path correction value as defined by the Ingress Correction expression.

GMAC_MAC_TS_Egress_Corr_NS

Address: Operational Base + offset (0x0b5c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	TSEC Timestamp Egress Correction This field contains the nanoseconds part of the egress path correction value as defined by the Egress Correction expression.

GMAC_MAC_TS_Ingress_Latency

Address: Operational Base + offset (0x0b68)

Bit	Attr	Reset Value	Description
31:24	RO	0x00	reserved
23:16	RO	0x00	ITLNS Ingress Timestamp Latency, in sub-nanoseconds This register holds the average latency in sub-nanoseconds between the input ports (phy_rxd_i) of MAC and the actual point (GMII/MII) where the ingress timestamp is taken.

Bit	Attr	Reset Value	Description
15:8	RO	0x00	ITLSNS Ingress Timestamp Latency, in nanoseconds This register holds the average latency in nanoseconds between the input ports (phy_rxd_i) of MAC and the actual point (GMII/MII) where the ingress timestamp is taken.

GMAC_MAC_TS_Egress_Latency

Address: Operational Base + offset (0x0b6c)

Bit	Attr	Reset Value	Description
31:24	RO	0x00	reserved
23:16	RO	0x00	ETLNS Egress Timestamp Latency, in nanoseconds This register holds the average latency in nanoseconds between the actual point (GMII/MII) where the egress timestamp is taken and the output ports (phy_txd_o) of the MAC.
15:8	RO	0x00	ETLSNS Egress Timestamp Latency, in sub-nanoseconds This register holds the average latency in sub-nanoseconds between the actual point (GMII/MII) where the egress timestamp is taken and the output ports (phy_txd_o) of the MAC.

GMAC_MAC_PPS_Control

Address: Operational Base + offset (0x0b70)

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

Bit	Attr	Reset Value	Description
3:0	RW	0x0	PPSCTRL_PPSCMD(cont.) Flexible PPS Output (ptp_pps_o[0]) Control Programming these bits with a non-zero value instructs the MAC to initiate an event. When the command is transferred or synchronized to the PTP clock domain, these bits get cleared automatically. The software should ensure that these bits are programmed only when they are 'all-zero'. The following list describes the values of PPSCMD0: 4'b0000: No Command 4'b0001: START Single Pulse This command generates single pulse rising at the start point defined in MAC_PPS0_Target_Time_Seconds and MAC_PPS0_Target_Time_Nanoseconds register and of a duration defined in the PPS0 Width Register. 4'b0010: START Pulse Train This command generates the train of pulses rising at the start point defined in the Target Time Registers and of a duration defined in the PPS0 Width Register and repeated at interval defined in the PPS Interval Register. By default, the PPS pulse train is free-running unless stopped by the 'Stop Pulse train at time' or 'Stop Pulse Train immediately' commands. 4'b0011: Cancel START This command cancels the START Single Pulse and START Pulse Train commands if the system time has not crossed the programmed start time. 4'b0100: STOP Pulse train at time This command stops the train of pulses initiated by the START Pulse Train command (PPSCMD = 0010) after the time programmed in the Target Time registers elapses. 4'b0101: STOP Pulse Train immediately This command immediately stops the train of pulses initiated by the START Pulse Train command (PPSCMD = 0010). 4'b0110: Cancel STOP Pulse train This command cancels the STOP pulse train at time command if the programmed stop time has not elapsed. The PPS pulse train becomes free-running on the successful execution of this command. 4'b0111-4'b1111: Reserved

Bit	Attr	Reset Value	Description
3:0	RW	0x0	PPSCTRL_PPSCMD PPS Output Frequency Control This field controls the frequency of the PPS0 output (ptp_pps_o[0]) signal. The default value of PPSCTRL is 0000, and the PPS output is 1 pulse (of width clk_ptp_i) every second. For other values of PPSCTRL, the PPS output becomes a generated clock of following frequencies: 4'b0001: The binary rollover is 2 Hz, and the digital rollover is 1 Hz. 4'b0010: The binary rollover is 4 Hz, and the digital rollover is 2 Hz. 4'b0011: The binary rollover is 8 Hz, and the digital rollover is 4 Hz. 4'b0100: The binary rollover is 16 Hz, and the digital rollover is 8 Hz. ... 4'b1111: The binary rollover is 32.768 KHz and the digital rollover is 16.384 KHz. Note: In the binary rollover mode, the PPS output (ptp_pps_o) has a duty cycle of 50 percent with these frequencies. In the digital rollover mode, the PPS output frequency is an average number. The actual clock is of different frequency that gets synchronized every second. For example: 1. When PPSCTRL = 0001, the PPS (1 Hz) has a low period of 537 ms and a high period of 463 ms. 2. When PPSCTRL = 0010, the PPS (2 Hz) is a sequence of One clock of 50 percent duty cycle and 537 ms period Second clock of 463 ms period (268 ms low and 195 ms high). 3. When PPSCTRL = 0011, the PPS (4 Hz) is a sequence of Three clocks of 50 percent duty cycle and 268 ms period Fourth clock of 195 ms period (134 ms low and 61 ms high). This behavior is because of the non-linear toggling of bits in the digital rollover mode in the MAC_System_Time_Nanoseconds register.

GMAC_MTL_DBG_CTL

Address: Operational Base + offset (0x0c08)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15	RW	0x0	STSIE Transmit Status Available Interrupt Status Enable When this bit is set, an interrupt is generated when Transmit status is available in slave mode. Values: 1'b0: Transmit Packet Available Interrupt Status is disabled 1'b1: Transmit Packet Available Interrupt Status is enabled
14	RW	0x0	PKTIE Receive Packet Available Interrupt Status Enable When this bit is set, an interrupt is generated when EOP of received packet is written to the Rx FIFO. Values: 1'b0: Receive Packet Available Interrupt Status is disabled 1'b1: Receive Packet Available Interrupt Status is enabled

Bit	Attr	Reset Value	Description
13:12	RW	0x0	FIFOSEL FIFO Selected for Access This field indicates the FIFO selected for debug access: Values: 2'b00: Tx FIFO 2'b01: Tx Status FIFO (only read access when SLVMOD is set) 2'b10: TSO FIFO (cannot be accessed when SLVMOD is set) 2'b11: Rx FIFO
11	RW	0x0	FIFOWREN FIFO Write Enable When this bit is set, it enables the Write operation on selected FIFO when FIFO Debug Access is enabled. This bit must not be written to 1 when FIFO Debug Access is not enabled, that is FDBGEN bit is 0. Access restriction applies. Self-cleared. Setting 0 clears. Setting 1 sets. Values: 1'b0: FIFO Write is disabled 1'b1: FIFO Write is enabled
10	RW	0x0	FIFORDEN FIFO Read Enable When this bit is set, it enables the Read operation on selected FIFO when FIFO Debug Access is enabled. This bit must not be written to 1 when FIFO Debug Access is not enabled, that is FDBGEN bit is 0. Access restriction applies. Self-cleared. Setting 0 clears. Setting 1 sets. Values: 1'b0: FIFO Read is disabled 1'b1: FIFO Read is enabled
9	RW	0x0	RSTSEL Reset Pointers of Selected FIFO When this bit is set, the pointers of the currently-selected FIFO are reset when FIFO Debug Access is enabled. This bit must not be written to 1 when FIFO Debug Access is not enabled, that is FDBGEN bit is 0. Access restriction applies. Self-cleared. Setting 0 clears. Setting 1 sets. Values: 1'b0: Reset Pointers of Selected FIFO is disabled 1'b1: Reset Pointers of Selected FIFO is enabled
8	RW	0x0	RSTALL Reset All Pointers When this bit is set, the pointers of all FIFOs are reset when FIFO Debug Access is enabled. This bit must not be written to 1 when FIFO Debug Access is not enabled, that is FDBGEN bit is 0. Access restriction applies. Self-cleared. Setting 0 clears. Setting 1 sets. Values: 1'b0: Reset All Pointers is disabled 1'b1: Reset All Pointers is enabled
7	RO	0x0	reserved

Bit	Attr	Reset Value	Description
6:5	RW	0x0	PKTSTATE Encoded Packet State This field is used to write the control information to the Tx FIFO or Rx FIFO. Tx FIFO: 2'b00: Packet Data 2'b01: Control Word 2'b10: SOP Data 2'b11: EOP Data Rx FIFO: 2'b00: Packet Data 2'b01: Normal Status 2'b10: Last Status 2'b11: EOP Values: 2'b00: Packet Data 2'b01: Control Word/Normal Status 2'b10: SOP Data/Last Status 2'b11: EOP Data/EOP
4	RO	0x0	reserved
3:2	RW	0x0	BYTEEN Byte Enables This field indicates the number of data bytes valid in the data register during Write operation. This is valid only when PKTSTATE is 2'b10 (EOP) and Tx FIFO or Rx FIFO is selected. Values: 2'b00: Byte 0 valid 2'b01: Byte 0 and Byte 1 are valid 2'b10: Byte 0, Byte 1, and Byte 2 are valid 2'b11: All four bytes are valid
1	RW	0x0	DBGMOD Debug Mode Access to FIFO When this bit is set, it indicates that the current access to the FIFO is read, write, and debug access. In this mode, the following access types are allowed: 1. Read and Write access to Tx FIFO, TSO FIFO, and Rx FIFO 2. Read access is allowed to Tx Status FIFO. When this bit is reset, it indicates that the current access to the FIFO is slave access bypassing the DMA. In this mode, the following access are allowed: 1. Write access to the Tx FIFO 2. Read access to the Rx FIFO and Tx Status FIFO Values: 1'b0: Debug Mode Access to FIFO is disabled 1'b1: Debug Mode Access to FIFO is enabled
0	RW	0x0	FDBGEN FIFO Debug Access Enable When this bit is set, it indicates that the debug mode access to the FIFO is enabled. When this bit is reset, it indicates that the FIFO can be accessed only through a master interface. Values: 1'b0: FIFO Debug Access is disabled 1'b1: FIFO Debug Access is enabled

GMAC_MTL_DBG_STS

Address: Operational Base + offset (0x0c0c)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15	RO	0x0	LOCR Remaining Locations in the FIFO Slave Access Mode: This field indicates the space available in selected FIFO. Debug Access Mode: This field contains the Write or Read pointer value of the selected FIFO during Write or Read operation, respectively. Reset: In single Tx Queue configurations, (DWC_EQOS_TXFIFO_SIZE/(DWC_EQOS_DATAWIDTH/8)), Otherwise 0000H.
14:10	RO	0x00	reserved
9	RW	0x0	STSI Transmit Status Available Interrupt Status When set, this bit indicates that the Slave mode Tx packet is transmitted, and the status is available in Tx Status FIFO. This bit is reset when 1 is written to this bit. Values: 1'b0: Transmit Status Available Interrupt Status not detected 1'b1: Transmit Status Available Interrupt Status detected
8	RW	0x0	PKTI Receive Packet Available Interrupt Status When set, this bit indicates that MAC layer has written the EOP of received packet to the Rx FIFO. This bit is reset when 1 is written to this bit. Values: 1'b0: Receive Packet Available Interrupt Status not detected 1'b1: Receive Packet Available Interrupt Status detected
7:5	RO	0x0	reserved
4:3	RO	0x3	BYTEEN Byte Enables This field indicates the number of data bytes valid in the data register during Read operation. This is valid only when PKTSTATE is 2'b10 (EOP) and Tx FIFO or Rx FIFO is selected. Values: 2'b00: Byte 0 valid 2'b01: Byte 0 and Byte 1 are valid 2'b10: Byte 0, Byte 1, and Byte 2 are valid 2'b11: All four bytes are valid

Bit	Attr	Reset Value	Description
2:1	RO	0x0	PKTSTATE Encoded Packet State This field is used to get the control or status information of the selected FIFO. Tx FIFO: 2'b00: Packet Data 2'b01: Control Word 2'b10: SOP Data 2'b11: EOP Data Rx FIFO: 2'b00: Packet Data 2'b01: Normal Status 2'b10: Last Status 2'b11: EOP This field is applicable only for Tx FIFO and Rx FIFO during Read operation. Values: 2'b00: Packet Data 2'b01: Control Word/Normal Status 2'b10: SOP Data/Last Status 2'b11: EOP Data/EOP
0	RO	0x0	FIFOBUSY FIFO Busy When set, this bit indicates that a FIFO operation is in progress in the MAC and content of the following fields is not valid: 1. All other fields of this register 2. All fields of the MTL_FIFO_Debug_Data register Values: 1'b0: FIFO Busy not detected 1'b1: FIFO Busy detected

GMAC_MTL_FIFO_Debug_Data

Address: Operational Base + offset (0x0c10)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	FDBGDATA FIFO Debug Data During debug or slave access write operation, this field contains the data to be written to the Tx FIFO, Rx FIFO, or TSO FIFO. During debug or slave access read operation, this field contains the data read from the Tx FIFO, Rx FIFO, TSO FIFO, or Tx Status FIFO.

GMAC_MTL_Interrupt_Status

Address: Operational Base + offset (0x0c20)

Bit	Attr	Reset Value	Description
31:18	RO	0x0000	reserved
17	RO	0x0	DBGIS Debug Interrupt status This bit indicates an interrupt event during the slave access. To reset this bit, the application must read the FIFO Debug Access Status register to get the exact cause of the interrupt and clear its source. Values: 1'b0: Debug Interrupt status not detected 1'b1: Debug Interrupt status detected

Bit	Attr	Reset Value	Description
16:1	RO	0x0000	reserved
0	RO	0x0	Q0IS Queue 0 Interrupt status This bit indicates that there is an interrupt from Queue 0. To reset this bit, the application must read Queue 0 Interrupt Control and Status register to get the exact cause of the interrupt and clear its source. Values: 1'b0: Queue 0 Interrupt status not detected 1'b1: Queue 0 Interrupt status detected

GMAC_MTL_TxQ0_Operation_Mode

Address: Operational Base + offset (0x0d00)

Bit	Attr	Reset Value	Description
31:22	RO	0x000	reserved
21:16	RW	0x3f	TQS Transmit Queue Size This field indicates the size of the allocated Transmit queues in blocks of 256 bytes. The TQS field is read-write only if the number of Tx Queues more than one, the reset value is 0x0 and indicates size of 256 bytes. When the number of Tx Queues is one, the field is read-only and the configured TX FIFO size in blocks of 256 bytes is reflected in the reset value. The width of this field depends on the Tx memory size selected in your configuration. For example, if the memory size is 2048, the width of this field is 3 bits: LOG2(2048/256) = LOG2(8) = 3 bits.
15:7	RO	0x000	reserved
6:4	RW	0x0	TTC Transmit Threshold Control These bits control the threshold level of the MTL Tx Queue. The transmission starts when the packet size within the MTL Tx Queue is larger than the threshold. In addition, full packets with length less than the threshold are also transmitted. These bits are used only when the TSF bit is reset. Values: 3'b000: 32 bytes 3'b001: 64 bytes 3'b010: 96 bytes 3'b011: 128 bytes 3'b100: 192 bytes 3'b101: 256 bytes 3'b110: 384 bytes 3'b111: 512 bytes
3:2	RW	0x2	TXQEN Transmit Queue Enable This field is used to enable/disable the transmit queue 0. 2'b00: Not enabled 2'b01: Reserved 2'b10: Enabled 2'b11: Reserved This field is Read Only in Single Queue configurations and Read Write in Multiple Queue configurations.

Bit	Attr	Reset Value	Description
1	RW	0x0	TSF Transmit Store and Forward When this bit is set, the transmission starts when a full packet resides in the MTL Tx queue. When this bit is set, the TTC values specified in Bits[6:4] of this register are ignored. This bit should be changed only when the transmission is stopped. Values: 1'b0: Transmit Store and Forward is disabled 1'b1: Transmit Store and Forward is enabled
0	RW	0x0	FTQ Flush Transmit Queue When this bit is set, the Tx queue controller logic is reset to its default values. Therefore, all the data in the Tx queue is lost or flushed. This bit is internally reset when the flushing operation is complete. Until this bit is reset, you should not write to the MTL_TxQ1_Operation_Mode register. The data which is already accepted by the MAC transmitter is not flushed. It is scheduled for transmission and results in underflow and runt packet transmission. Note: The flush operation is complete only when the Tx queue is empty and the application has accepted the pending Tx Status of all transmitted packets. To complete this flush operation, the PHY Tx clock (clk_tx_i) should be active. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. Values: 1'b0: Flush Transmit Queue is disabled 1'b1: Flush Transmit Queue is enabled

GMAC_MTL_TxQ0_Underflow

Address: Operational Base + offset (0x0d04)

Bit	Attr	Reset Value	Description
31:12	RO	0x00000	reserved
11	RO	0x0	UFCNTOVF Overflow Bit for Underflow Packet Counter This bit is set every time the Tx queue Underflow Packet Counter field overflows, that is, it has crossed the maximum count. In such a scenario, the overflow packet counter is reset to all-zeros and this bit indicates that the rollover happened. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: Overflow not detected for Underflow Packet Counter 1'b1: Overflow detected for Underflow Packet Counter
10:0	RO	0x000	UFFRMCNT Underflow Packet Counter This field indicates the number of packets aborted by the controller because of Tx Queue Underflow. This counter is incremented each time the MAC aborts outgoing packet because of underflow. The counter is cleared when this register is read with mci_be_i[0] at 1'b1. Access restriction applies. Clears on read. Self-set to 1 on internal event.

GMAC_MTL_TxQ0_Debug

Address: Operational Base + offset (0x0d08)

Bit	Attr	Reset Value	Description
31:23	RO	0x000	reserved
22:20	RO	0x0	STXSTSF Number of Status Words in Tx Status FIFO of Queue This field indicates the current number of status in the Tx Status FIFO of this queue.When the DTXSTS bit of MTL_Operation_Mode register is set to 1, this field does not reflect the number of status words in Tx Status FIFO.
19	RO	0x0	reserved
18:16	RO	0x0	PTXQ Number of Packets in the Transmit Queue This field indicates the current number of packets in the Tx Queue. When the DTXSTS bit of MTL_Operation_Mode register is set to 1, this field does not reflect the number of packets in the Transmit queue.
15:6	RO	0x000	reserved
5	RO	0x0	TXSTSFSTS MTL Tx Status FIFO Full Status When high, this bit indicates that the MTL Tx Status FIFO is full. Therefore, the MTL cannot accept any more packets for transmission. Values: 1'b0: MTL Tx Status FIFO Full status is not detected 1'b1: MTL Tx Status FIFO Full status is detected
4	RO	0x0	TXQSTS MTL Tx Queue Not Empty Status When this bit is high, it indicates that the MTL Tx Queue is not empty and some data is left for transmission. Values: 1'b0: MTL Tx Queue Not Empty status is not detected 1'b1: MTL Tx Queue Not Empty status is detected
3	RO	0x0	TWCSTS MTL Tx Queue Write Controller Status When high, this bit indicates that the MTL Tx Queue Write Controller is active, and it is transferring the data to the Tx Queue. Values: 1'b0: MTL Tx Queue Write Controller status is not detected 1'b1: MTL Tx Queue Write Controller status is detected
2:1	RO	0x0	TRCSTS MTL Tx Queue Read Controller Status This field indicates the state of the Tx Queue Read Controller: Values: 2'b00: Idle state 2'b01: Read state (transferring data to the MAC transmitter) 2'b10: Waiting for pending Tx Status from the MAC transmitter 2'b11: Flushing the Tx queue because of the Packet Abort request from the MAC

Bit	Attr	Reset Value	Description
0	RO	0x0	TXQPAUSED Transmit Queue in Pause When this bit is high and the Rx flow control is enabled, it indicates that the Tx Queue is in the Pause condition (in the full-duplex only mode) because of the following: 1. Reception of the PFC packet for the priorities assigned to the Tx Queue when PFC is enabled 2. Reception of 802.3x Pause packet when PFC is disabled Values: 1'b0: Transmit Queue in Pause status is not detected 1'b1: Transmit Queue in Pause status is detected

GMAC_MTL_Q0_Interrupt_Ctrl_Status

Address: Operational Base + offset (0x0d2c)

Bit	Attr	Reset Value	Description
31:25	RO	0x00	reserved
24	RW	0x0	RXOIE Receive Queue Overflow Interrupt Enable When this bit is set, the Receive Queue Overflow interrupt is enabled. When this bit is reset, the Receive Queue Overflow interrupt is disabled. Values: 1'b0: Receive Queue Overflow Interrupt is disabled 1'b1: Receive Queue Overflow Interrupt is enabled
23:17	RO	0x00	reserved
16	RW	0x0	RXOVFIS Receive Queue Overflow Interrupt Status This bit indicates that the Receive Queue had an overflow while receiving the packet. If a partial packet is transferred to the application, the overflow status is set in RDES3[21]. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. Values: 1'b0: Receive Queue Overflow Interrupt Status not detected 1'b1: Receive Queue Overflow Interrupt Status detected
15:9	RO	0x00	reserved
8	RW	0x0	TXUIE Transmit Queue Underflow Interrupt Enable When this bit is set, the Transmit Queue Underflow interrupt is enabled. When this bit is reset, the Transmit Queue Underflow interrupt is disabled. Values: 1'b0: Transmit Queue Underflow Interrupt Status is disabled 1'b1: Transmit Queue Underflow Interrupt Status is enabled
7:1	RO	0x00	reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	TXUNFIS Transmit Queue Underflow Interrupt Status This bit indicates that the Transmit Queue had an underflow while transmitting the packet. Transmission is suspended and an Underflow Error TDES3[2] is set. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. Values: 1'b0: Transmit Queue Underflow Interrupt Status not detected 1'b1: Transmit Queue Underflow Interrupt Status detected

GMAC_MTL_RxQ0_Operation_Mode

Address: Operational Base + offset (0x0d30)

Bit	Attr	Reset Value	Description
31:27	RO	0x00	reserved
26:20	RW	0x7f	RQS Receive Queue Size This field indicates the size of the allocated Receive queues in blocks of 256 bytes. The RQS field is read-write only if the number of Rx Queues more than one, the reset value is 0x0 and indicates size of 256 bytes. When the number of Rx Queues is one, the field is read-only and the configured RX FIFO size in blocks of 256 bytes is reflected in the reset value. The width of this field depends on the Rx memory size selected in your configuration. For example, if the memory size is 2048, the width of this field is 3 bits: LOG2(2048/256) = LOG2(8) = 3 bits.
19:14	RW	0x00	RFD Threshold for Deactivating Flow Control (in half-duplex and full-duplex modes) These bits control the threshold (fill-level of Rx queue) at which the flow control is de-asserted after activation: 0: Full minus 1 KB, that is, FULL 1 KB 1: Full minus 1.5 KB, that is, FULL 1.5 KB 2: Full minus 2 KB, that is, FULL 2 KB 3: Full minus 2.5 KB, that is, FULL 2.5 KB ... 62: Full minus 32 KB, that is, FULL 32 KB 63: Full minus 32.5 KB, that is, FULL 32.5 KB The de-assertion is effective only after flow control is asserted. Note: The value must be programmed in such a way to make sure that the threshold is a positive number. When the EHFC is set high, these values are applicable only when the Rx queue size determined by the RQS field of this register, is equal to or greater than 4 KB. For a given queue size, the values ranges between 0 and the encoding for FULL minus (QSIZE - 0.5 KB) and all other values are illegal. Here the term FULL and QSIZE refers to the queue size determined by the RQS field of this register. The width of this field depends on RX FIFO size selected during the configuration. Remaining bits are reserved and read only.

Bit	Attr	Reset Value	Description
13:8	RW	0x00	RFA Threshold for Activating Flow Control (in half-duplex and full-duplex) These bits control the threshold (fill-level of Rx queue) at which the flow control is activated:For more information on encoding for this field, see RFD.
7	RW	0x0	EHFC Enable Hardware Flow Control When this bit is set, the flow control signal operation, based on the fill-level of Rx queue, is enabled. When reset, the flow control operation is disabled. Values: 1'b0: Hardware Flow Control is disabled 1'b1: Hardware Flow Control is enabled
6	RW	0x0	DIS_TCP_EF Disable Dropping of TCP/IP Checksum Error Packets When this bit is set, the MAC does not drop the packets which only have the errors detected by the Receive Checksum Offload engine. Such packets have errors only in the encapsulated payload. There are no errors (including FCS error) in the Ethernet packet received by the MAC. When this bit is reset, all error packets are dropped if the FEP bit is reset. Values: 1'b0: Dropping of TCP/IP Checksum Error Packets is enabled 1'b1: Dropping of TCP/IP Checksum Error Packets is disabled
5	RW	0x0	RSF Receive Queue Store and Forward When this bit is set, the DWC_ether_qos reads a packet from the Rx queue only after the complete packet has been written to it, ignoring the RTC field of this register. When this bit is reset, the Rx queue operates in the Threshold (cut-through) mode, subject to the threshold specified by the RTC field of this register. Values: 1'b0: Receive Queue Store and Forward is disabled 1'b1: Receive Queue Store and Forward is enabled
4	RW	0x0	FEP Forward Error Packets When this bit is reset, the Rx queue drops packets with error status (CRC error, GMII_ER, watchdog timeout, or overflow). However, if the start byte (write) pointer of a packet is already transferred to the read controller side (in Threshold mode), the packet is not dropped. When this bit is set, all packets except the runt error packets are forwarded to the application or DMA. If the RSF bit is set and the Rx queue overflows when a partial packet is written, the packet is dropped irrespective of the setting of this bit. However, if the RSF bit is reset and the Rx queue overflows when a partial packet is written, a partial packet may be forwarded to the application or DMA. Values: 1'b0: Forward Error Packets is disabled 1'b1: Forward Error Packets is enabled

Bit	Attr	Reset Value	Description
3	RW	0x0	FUP Forward Undersized Good Packets When this bit is set, the Rx queue forwards the undersized good packets (packets with no error and length less than 64 bytes), including pad-bytes and CRC. When this bit is reset, the Rx queue drops all packets of less than 64 bytes, unless a packet is already transferred because of the lower value of Rx Threshold, for example, RTC = 01. Values: 1'b0: Forward Undersized Good Packets is disabled 1'b1: Forward Undersized Good Packets is enabled
2	RO	0x0	reserved
1:0	RW	0x0	RTC Receive Queue Threshold Control These bits control the threshold level of the MTL Rx queue (in bytes): The received packet is transferred to the application or DMA when the packet size within the MTL Rx queue is larger than the threshold. In addition, full packets with length less than the threshold are automatically transferred. This field is valid only when the RSF bit is zero. This field is ignored when the RSF bit is set to 1.

GMAC_MTL_RxQ0_Miss_Pkt_Ovf_Cnt

Address: Operational Base + offset (0x0d34)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27	RO	0x0	MISCNTOVF Missed Packet Counter Overflow Bit When set, this bit indicates that the Rx Queue Missed Packet Counter crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: Missed Packet Counter overflow not detected 1'b1: Missed Packet Counter overflow detected
26:16	RO	0x000	MISPKTCNT Missed Packet Counter This field indicates the number of packets missed by the DWC_ether_qos because the application asserted ari_pkt_flush_i[] for this queue. This counter is incremented each time the application issues ari_pkt_flush_i[] for this queue. This counter is reset when this register is read with mci_be_i[0] at 1b1. In EQOS-DMA, EQOS-AXI, and EQOS-AHB configurations, This counter is incremented by 1 when the DMA discards the packet because of buffer unavailability. Access restriction applies. Clears on read. Self-set to 1 on internal event.
15:12	RO	0x0	reserved

Bit	Attr	Reset Value	Description
11	RO	0x0	OVFCNTOVF Overflow Counter Overflow Bit When set, this bit indicates that the Rx Queue Overflow Packet Counter field crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: Overflow Counter overflow not detected 1'b1: Overflow Counter overflow detected
10:0	RO	0x000	OVFPKTCNT Overflow Packet Counter This field indicates the number of packets discarded by the DWC_ether_qos because of Receive queue overflow. This counter is incremented each time the DWC_ether_qos discards an incoming packet because of overflow. This counter is reset when this register is read with mci_be_i[0] at 1'b1. Access restriction applies. Clears on read. Self-set to 1 on internal event.

GMAC_MTL_RxQ0_Debug

Address: Operational Base + offset (0x0d38)

Bit	Attr	Reset Value	Description
31:20	RO	0x000	reserved
19:16	RO	0x0	PRXQ Number of Packets in Receive Queue This field indicates the current number of packets in the Rx Queue. The theoretical maximum value for this field is 256KB/16B = 16K Packets, that is, Max_Queue_Size/Min_Packet_Size.
15:6	RO	0x000	reserved
5:4	RO	0x1	RXQSTS MTL Rx Queue Fill-Level Status This field gives the status of the fill-level of the Rx Queue: Values: 2'b00: Rx Queue empty 2'b01: Rx Queue fill-level below flow-control deactivate threshold 2'b10: Rx Queue fill-level above flow-control activate threshold 2'b11: Rx Queue full
3	RO	0x0	reserved
2:1	RO	0x0	RRCSTS MTL Rx Queue Read Controller State This field gives the state of the Rx queue Read controller: Values: 2'b00: Idle state 2'b01: Reading packet data 2'b10: Reading packet status (or timestamp) 2'b11: Flushing the packet data and status
0	RO	0x0	RWCSTS MTL Rx Queue Write Controller Active Status When high, this bit indicates that the MTL Rx queue Write controller is active, and it is transferring a received packet to the Rx Queue. Values: 1'b0: MTL Rx Queue Write Controller Active Status not detected 1'b1: MTL Rx Queue Write Controller Active Status detected

GMAC_DMA_Mode

Address: Operational Base + offset (0x1000)

Bit	Attr	Reset Value	Description
31:18	RO	0x0000	reserved
17:16	RW	0x0	INTM Interrupt Mode This field defines the interrupt mode of DWC_ether_qos. The behavior of the following outputs changes depending on the following settings: 1. sbd_perch_tx_intr_o[] (Transmit Per Channel Interrupt) 2. sbd_perch_rx_intr_o[] (Receive Per Channel Interrupt) 3. sbd_intr_o (Common Interrupt) It also changes the behavior of the RI/TI bits in the DMA_CH0_Status. 2'b00: sbd_perch_* are pulse signals for each TX/RX packet transfer completion events (irrespective of whether corresponding interrupts are enabled) for which IOC bits are enabled in descriptor. sbd_intr_o is also asserted when corresponding interrupts are enabled and cleared only when software clears the corresponding RI/TI status bits. 2'b01: sbd_perch_* are level signals asserted on TX/RX packet transfer completion event when corresponding interrupts are enabled and de-asserted when the software clears the corresponding RI/TI status bits. The sbd_intr_o is not asserted for these TX/RX packet transfer completion events. 2'b10: sbd_perch_* are level signals asserted on TX/RX packet transfer completion event when corresponding interrupts are enabled and de-asserted when the software clears the corresponding RI/TI status bits. However, the signal is asserted again if the same event occurred again before it was cleared. The sbd_intr_o is not asserted for these TX/RX packet transfer completion events. 2'b11: Reserved Values: 2'b00: See above description 2'b01: See above description 2'b10: See above description 2'b11: Reserved
15:9	RO	0x00	reserved
8	RW	0x0	DSPW Descriptor Posted Write When this bit is set to 0, the descriptor writes are always non-posted. When this bit is set to 1, the descriptor writes are non-posted only when IOC (Interrupt on completion) is set in last descriptor, otherwise the descriptor writes are always posted. Values: 1'b0: Descriptor Posted Write is disabled 1'b1: Descriptor Posted Write is enabled
7:1	RO	0x00	reserved

Bit	Attr	Reset Value	Description
0	RW	0x1	SWR Software Reset When this bit is set, the MAC and the DMA controller reset the logic and all internal registers of the DMA, MTL, and MAC. This bit is automatically cleared after the reset operation is complete in all DWC_ether_qos clock domains. Before reprogramming any DWC_ether_qos register, a value of zero should be read in this bit. This bit must be read at least 4 CSR clock cycles after it is written to 1. Note: The reset operation is complete only when all resets in all active clock domains are de-asserted. Therefore, it is essential that all PHY inputs clocks (applicable for the selected PHY interface) are present for software reset completion. The time to complete the software reset operation depends on the frequency of the slowest active clock. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. Values: 1'b0: Software Reset is disabled 1'b1: Software Reset is enabled

GMAC_DMA_SysBus_Mode

Address: Operational Base + offset (0x1004)

Bit	Attr	Reset Value	Description
31	RW	0x0	EN_LPI Enable Low Power Interface (LPI) When set to 1, this bit enables the LPI mode supported by the EQOS-AXI configuration and accepts the LPI request from the AXI System Clock controller. When set to 0, this bit disables the LPI mode and always denies the LPI request from the AXI System Clock controller. Values: 1'b0: Low Power Interface (LPI) is disabled 1'b1: Low Power Interface (LPI) is enabled
30	RW	0x0	LPI_XIT_PKT Unlock on Magic Packet or Remote Wake-Up Packet When set to 1, this bit enables the AXI master to come out of the LPI mode only when the magic packet or remote wake-up packet is received. When set to 0, this bit enables the AXI master to come out of the LPI mode when any packet is received. Values: 1'b0: Unlock on Magic Packet or Remote Wake-Up Packet is disabled 1'b1: Unlock on Magic Packet or Remote Wake-Up Packet is enabled
29:26	RO	0x0	reserved

Bit	Attr	Reset Value	Description
25:24	RW	0x1	WR_OSR_LMT AXI Maximum Write Outstanding Request Limit This value limits the maximum outstanding request on the AXI write interface. Maximum outstanding requests = WR_OSR_LMT + 1 Note: 1. Bit 26 is reserved if DWC_ETHER_QOS_AXI_MAX-_WR_REQ = 4 2. Bit 27 is reserved if DWC_ETHER_QOS_AXI_MAX-_WR_REQ!= 16
23:19	RO	0x00	reserved
18:16	RW	0x1	RD_OSR_LMT AXI Maximum Read Outstanding Request Limit This value limits the maximum outstanding request on the AXI read interface. Maximum outstanding requests = RD_OSR_LMT + 1 Note: 1. Bit 18 is reserved if parameter DWC_ETHER_QOS_AXI_-MAX_RD_REQ = 4 2. Bit 19 is reserved if parameter DWC_ETHER_QOS_AXI_-MAX_RD_REQ!= 16
15:13	RO	0x0	reserved
12	RW	0x0	AAL Address-Aligned Beats When this bit is set to 1, the EQOS-AXI or EQOS-AHB master performs address-aligned burst transfers on Read and Write channels. Values: 1'b0: Address-Aligned Beats is disabled 1'b1: Address-Aligned Beats is enabled
11	RO	0x0	reserved
10	RW	0x0	AALE Automatic AXI LPI enable When set to 1, enables the AXI master to enter into LPI state when there is no activity in the DWC_ether_qos for number of system clock cycles programmed in the LPIEI field of AXI_LPI_Entry_Interval register. Values: 1'b0: Automatic AXI LPI is disabled 1'b1: Automatic AXI LPI is enabled
9:4	RO	0x00	reserved
3	RW	0x0	BLEN16 AXI Burst Length 16 When this bit is set to 1 or the FB bit is set to 0, the EQOS-AXI master can select a burst length of 16 on the AXI interface. When the FB bit is set to 0, setting this bit has no effect. Values: 1'b0: No effect 1'b1: AXI Burst Length 16

Bit	Attr	Reset Value	Description
2	RW	0x0	BLEN8 AXI Burst Length 8 When this bit is set to 1 or the FB bit is set to 0, the EQOS-AXI master can select a burst length of 8 on the AXI interface. When the FB bit is set to 0, setting this bit has no effect. Values: 1'b0: No effect 1'b1: AXI Burst Length 8
1	RW	0x0	BLEN4 AXI Burst Length 4 When this bit is set to 1 or the FB bit is set to 0, the EQOS-AXI master can select a burst length of 4 on the AXI interface. When the FB bit is set to 0, setting this bit has no effect. Values: 1'b0: No effect 1'b1: AXI Burst Length 4
0	RW	0x0	FB Fixed Burst Length When this bit is set to 1, the EQOS-AXI master initiates burst transfers of specified lengths as given below. 1. Burst transfers of fixed burst lengths as indicated by the BLEN256, BLEN128, BLEN64, BLEN32, BLEN16, BLEN8, or BLEN4 field 2. Burst transfers of length 1 When this bit is set to 0, the EQOS-AXI master initiates burst transfers that are equal to or less than the maximum allowed burst length programmed in Bits[7:1]. Values: 1'b0: Fixed Burst Length is disabled 1'b1: Fixed Burst Length is enabled

GMAC_DMA_Interrupt_Status

Address: Operational Base + offset (0x1008)

Bit	Attr	Reset Value	Description
31:18	RO	0x0000	reserved
17	RO	0x0	MACIS MAC Interrupt Status This bit indicates an interrupt event in the MAC. To reset this bit to 1'b0, the software must read the corresponding register in the MAC to get the exact cause of the interrupt and clear its source. Values: 1'b0: MAC Interrupt Status not detected 1'b1: MAC Interrupt Status detected
16	RO	0x0	MTLIS MTL Interrupt Status This bit indicates an interrupt event in the MTL. To reset this bit to 1'b0, the software must read the corresponding register in the MTL to get the exact cause of the interrupt and clear its source. Values: 1'b0: MTL Interrupt Status not detected 1'b1: MTL Interrupt Status detected
15:1	RO	0x0000	reserved

Bit	Attr	Reset Value	Description
0	RO	0x0	DC0IS DMA Channel 0 Interrupt Status This bit indicates an interrupt event in DMA Channel 0. To reset this bit to 1'b0, the software must read the corresponding register in DMA Channel 0 to get the exact cause of the interrupt and clear its source. Values: 1'b0: DMA Channel 0 Interrupt Status not detected 1'b1: DMA Channel 0 Interrupt Status detected

GMAC_DMA_Debug_Status0

Address: Operational Base + offset (0x100c)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15:12	RO	0x0	TPS0 DMA Channel 0 Transmit Process State This field indicates the Tx DMA FSM state for Channel 0. The MSB of this field always returns 0. This field does not generate an interrupt. Values: 4'b0000: Stopped (Reset or Stop Transmit Command issued) 4'b0001: Running (Fetching Tx Transfer Descriptor) 4'b0010: Running (Waiting for status) 4'b0011: Running (Reading Data from system memory buffer and queuing it to the Tx buffer (Tx FIFO)) 4'b0100: Timestamp write state 4'b0101: Reserved for future use 4'b0110: Suspended (Tx Descriptor Unavailable or Tx Buffer Underflow) 4'b0111: Running (Closing Tx Descriptor)
11:8	RO	0x0	RPS0 DMA Channel 0 Receive Process State This field indicates the Rx DMA FSM state for Channel 0. The MSB of this field always returns 0. This field does not generate an interrupt. Values: 4'b0000: Stopped (Reset or Stop Receive Command issued) 4'b0001: Running (Fetching Rx Transfer Descriptor) 4'b0010: Reserved for future use 4'b0011: Running (Waiting for Rx packet) 4'b0100: Suspended (Rx Descriptor Unavailable) 4'b0101: Running (Closing the Rx Descriptor) 4'b0110: Timestamp write state 4'b0111: Running (Transferring the received packet data from the Rx buffer to the system memory)
7:2	RO	0x00	reserved
1	RO	0x0	AXRHSTS AXI Master Read Channel Status When high, this bit indicates that the read channel of the AXI master is active, and it is transferring the data. Values: 1'b0: AXI Master Read Channel Status not detected 1'b1: AXI Master Read Channel Status detected

Bit	Attr	Reset Value	Description
0	RO	0x0	AXWHSTS AXI Master Write Channel or AHB Master Status EQOS-AXI Configuration: When high, this bit indicates that the write channel of the AXI master is active, and it is transferring data. EQOS-AHB Configuration: When high, this bit indicates that the AHB master FSMs are in the non-idle state. Values: 1'b0: AXI Master Write Channel or AHB Master Status not detected 1'b1: AXI Master Write Channel or AHB Master Status detected

GMAC_AXI_LPI_Entry_Interval

Address: Operational Base + offset (0x1040)

Bit	Attr	Reset Value	Description
31:4	RO	0x0000000	reserved
3:0	RW	0x0	LPIEI LPI Entry Interval Contains the number of system clock cycles, multiplied by 64, to wait for an activity in the DWC_ether_qos to enter into the AXI low power state. 0 indicates 64 clock cycles.

30.5 Interface Description

Table 30-3 EQOS Interface M0

Module Pin	Direction	Pad Name	IOMUX Setting
RMII/RGMII interface			
mac_clk_m0	I/O	CIF_D12_M0/RGMII_CLK_M0/PDM_CLk0_M1/SPI1_CLK_M0/GPIO3_C0_d	GRF_GPIO3C_IOMUX_SEL_L[3:0]=3'b0010
mac_txen_m0	O	CIF_D9_M0/RGMII_TXEN_M0/I2S0_SDO3_SDI1_M1/SPI1_CS0n_M0/GPIO3_B5_d	GRF_GPIO3B_IOMUX_SEL_H[7:4]=3'b0010
mac_txd3_m0	O	CIF_D6_M0/RGMII_TXD3_M0/I2S0_LRCK_RX_M1/UART4_RTSN_M0/GPIO3_B2_d	GRF_GPIO3B_IOMUX_SEL_L[11:8]=3'b0010
mac_txd2_m0	O	CIF_D5_M0/RGMII_TXD2_M0/I2S0_SCLK_RX_M1/UART5_CTSN_M0/I2C5_SDA_M1/GPIO3_B1_d	GRF_GPIO3B_IOMUX_SEL_L[7:4]=3'b0010
mac_txd1_m0	O	CIF_D8_M0/RGMII_TXD1_M0/I2S0_SDO2_SDI2_M1/SPI1_CS1n_M0/GPIO3_B4_d	GRF_GPIO3B_IOMUX_SEL_H[3:0]=3'b0010
mac_txd0_m0	O	CIF_D7_M0/RGMII_TXD0_M0/I2S0_SDO1_SDI3_M1/UART4_CTSN_M0/GPIO3_B3_d	GRF_GPIO3B_IOMUX_SEL_L[15:12]=3'b0010
mac_txclk_m0	O	CIF_CLKOUT_M0/RGMII_TXCLK_M0/UART3_TX_M0/GPIO3_C6_d	GRF_GPIO3C_IOMUX_SEL_H[11:8]=3'b0010
mac_rxdv_m0	I	CIF_D13_M0/RGMII_RXDV_M0/PDM_SDI0_M1/GPIO3_C1_d	GRF_GPIO3C_IOMUX_SEL_L[7:4]=3'b0010
mac_rxer_m0	I	CIF_D14_M0/RGMII_RXER_M0/PDM_SDI1_M1/GPIO3_C2_d	GRF_GPIO3C_IOMUX_SEL_L[11:8]=3'b0010
mac_rxd3_m0	I	CIF_D4_M0/RGMII_RXD3_M0/I2S0_MCLK_M1/UART5_RTSN_M0/I2C5_SCL_M1/GPIO3_B0_d	GRF_GPIO3B_IOMUX_SEL_L[3:0]=3'b0010
mac_rxd2_m0	I	CIF_D3_M0/RGMII_RXD2_M0/I2S0_SDI0_M1/UART5_RX_M0/CAN_TXD_M1/PWM11_IR_M0/GPIO3_A7_d	GRF_GPIO3A_IOMUX_SEL_H[15:12]=3'b0010
mac_rxd1_m0	I	CIF_D11_M0/RGMII_RXD1_M0/PDM_SDI3_M1/SPI1_MISO_M0/GPIO3_B7_d	GRF_GPIO3B_IOMUX_SEL_H[15:12]=3'b0010
mac_rxd0_m0	I	CIF_D10_M0/RGMII_RXD0_M0/PDM_SDI2_M1/SPI1_MOSI_M0/GPIO3_B6_d	GRF_GPIO3B_IOMUX_SEL_H[11:8]=3'b0010

mac_rxclk_m0	I	CIF_HSYNC_M0/RGMII_RXCLK_M0/UART3_RX_M0/GPIO3_C7_d	GRF_GPIO3C_IOMUX_SEL_H[15:12]=3'b0010
mac_crs_m0	I	CIF_D1_M0/RGMII_CRS_M0/I2S0_LRCK_TX_M1/UART4_RX_M0/I2C3_SDA_M0/PWM9_M0/GPIO3_A5_d	GRF_GPIO3A_IOMUX_SEL_H[7:4]=3'b0010
mac_col_m0	I	CIF_D2_M0/RGMII_COL_M0/I2S0_SDO0_M1/UART5_TX_M0/CAN_RXD_M1/PWM10_M0/GPIO3_A6_d	GRF_GPIO3A_IOMUX_SEL_H[11:8]=3'b0010
Management interface			
mac_mdio_m0	I/O	CIF_D15_M0/RGMII_MDIO_M0/PDM_CLk1_M1/GPIO3_C3_d	GRF_GPIO3C_IOMUX_SEL_L[15:12]=3'b0010
mac_mdc_m0	O	CIF_VSYNC_M0/RGMII_MDC_M0/UART3_RTSN_M0/GPIO3_C4_d	GRF_GPIO3C_IOMUX_SEL_H[3:0]=3'b0010

Table 30-4 EQOS Interface M1

Module Pin	Direction	Pad Name	IOMUX Setting
RMII/RGMII interface			
mac_clk_m1	I/O	LCDC_D11/RGMII_CLK_M1/CIF_D7_M1/GPIO2_B7_d	GRF_GPIO2B_IOMUX_SEL_H[15:12]=3'b0010
mac_txen_m1	O	LCDC_D18/RGMII_TXEN_M1/CIF_D14_M1/GPIO2_C6_d	GRF_GPIO2C_IOMUX_SEL_H[11:8]=3'b0010
mac_txd3_m1	O	LCDC_D0/RGMII_TXD3_M1/CIF_D0_M1/UART4_RTSN_M1/GPIO2_A4_d	GRF_GPIO2A_IOMUX_SEL_H[3:0]=3'b0010
mac_txd2_m1	O	LCDC_D21/RGMII_TXD2_M1/CIF_CLKOUT_M1/I2S1_SCLK_M2/GPIO2_D1_d	GRF_GPIO2D_IOMUX_SEL_L[7:4]=3'b0010
mac_txd1_m1	O	LCDC_D16/RGMII_TXD1_M1/CIF_D12_M1/GPIO2_C4_d	GRF_GPIO2C_IOMUX_SEL_H[3:0]=3'b0010
mac_txd0_m1	O	LCDC_D15/RGMII_TXD0_M1/CIF_D11_M1/GPIO2_C3_d	GRF_GPIO2C_IOMUX_SEL_L[15:12]=3'b0010
mac_txclk_m1	O	LCDC_D22/RGMII_TXCLK_M1/CIF_CLKIN_M1/I2S1_LRCK_M2/GPIO2_D2_d	GRF_GPIO2D_IOMUX_SEL_L[11:8]=3'b0010
mac_rxdv_m1	I	LCDC_D8/RGMII_RXDV_M1/CIF_D4_M1/GPIO2_B4_d	GRF_GPIO2B_IOMUX_SEL_H[3:0]=3'b0010
mac_rxer_m1	I	LCDC_D12/RGMII_RXER_M1/CIF_D8_M1/GPIO2_C0_d	GRF_GPIO2C_IOMUX_SEL_L[3:0]=3'b0010
mac_rxd3_m1	I	LCDC_D20/RGMII_RXD3_M1/CIF_VSYNC_M1/I2S1_SDO_M2/GPIO2_D0_d	GRF_GPIO2D_IOMUX_SEL_L[3:0]=3'b0010
mac_rxd2_m1	I	LCDC_D19/RGMII_RXD2_M1/CIF_D15_M1/I2S1_MCLK_M2/GPIO2_C7_d	GRF_GPIO2C_IOMUX_SEL_H[15:12]=3'b0010
mac_rxd1_m1	I	LCDC_D10/RGMII_RXD1_M1/CIF_D6_M1/GPIO2_B6_d	GRF_GPIO2A_IOMUX_SEL_H[11:8]=3'b0010
mac_rxd0_m1	I	LCDC_D9/RGMII_RXD0_M1/CIF_D5_M1/GPIO2_B5_d	GRF_GPIO2A_IOMUX_SEL_H[7:4]=3'b0010
mac_rxclk_m1	I	LCDC_D23/RGMII_RXCLK_M1/CIF_HSYNC_M1/I2S1_SDI_M2/GPIO2_D3_d	GRF_GPIO2D_IOMUX_SEL_L[15:12]=3'b0010
mac_crs_m1	I	LCDC_D1/RGMII_CRS_M1/CIF_D1_M1/UART4_CTSN_M1/I2C5_SCL_M0/GPIO2_A5_d	GRF_GPIO2A_IOMUX_SEL_H[7:4]=3'b0010
mac_col_m1	I	LCDC_D2/RGMII_COL_M1/CIF_D2_M1/PWM5_M1/UART4_TX_M1/GPIO2_A6_d	GRF_GPIO2A_IOMUX_SEL_H[11:8]=3'b0010
Management interface			
mac_mdio_m1	I/O	LCDC_D13/RGMII_MDIO_M1/CIF_D9_M1/GPIO2_C1_d	GRF_GPIO2C_IOMUX_SEL_L[7:4]=3'b0010
mac_mdc_m1	O	LCDC_D14/RGMII_MDC_M1/CIF_D10_M1/GPIO2_C2_d	GRF_GPIO2C_IOMUX_SEL_L[11:8]=3'b0010

30.6 Application Note

30.6.1 Descriptors

The DMA engine uses descriptors to efficiently move data from source to destination with minimal application CPU intervention. The DMA is designed for packet-oriented data transfers such as packets in Ethernet. The DMA controller can be programmed to interrupt the application CPU for situations such as Packet Transmit and Receive Transfer completion, and other normal or error conditions.

The DMA and the application communicate through the following two data structures:

- Control and Status registers (CSR)
- Descriptor lists and data buffers

The base address of each list is written to the respective Tx Descriptor List Address register and Rx Descriptor List Address register. The descriptor list is forward linked and the next descriptor is always considered at a fixed offset to the current one. The offset is controlled by the DSL field of DMA_Ch[n]_Control register. The number of descriptors in the list is programmed in the respective Tx (or Rx) Descriptor Ring Length register. Once the DMA processes the last descriptor in the list, it automatically jumps back to the descriptor in the List Address register to create a descriptor ring.

The descriptor lists reside in the physical memory address space of the application. Each descriptor can point to a maximum of two buffers in the system memory. This enables two buffers to be used, physically addressed, rather than contiguous buffers in memory.

A data buffer resides in the application physical memory space and consists of an entire packet or part of a packet but cannot exceed a single packet. Buffers contain only data. Buffer status is maintained in the descriptor. Data chaining refers to packets that span multiple data buffers. However, a single descriptor cannot span multiple packets. The DMA skips to the data buffer of next packet when EOP is detected.

The GMAC supports the following two types of descriptors:

- Normal Descriptor: Normal descriptors are used for packet data and to provide control information applicable to the packets to be transmitted
- Context Descriptor: Context descriptors are used to provide control information applicable to the packet to be transmitted

The GMAC supports the ring structure for the DMA descriptor.

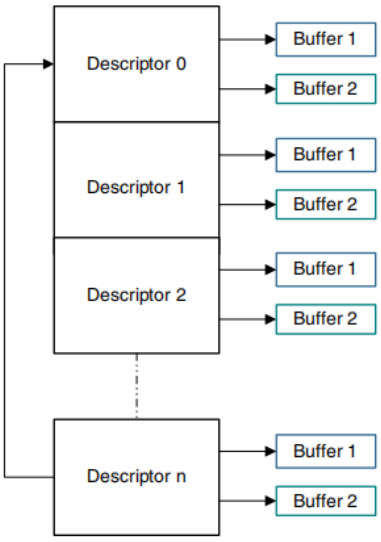


Fig. 30-10 Descriptor Ring Structure

In Ring structure, descriptors are separated by the Word, DWord, or LWord number programmed in the DSL field of the DMA_CH#_Control register. The application needs to program the total ring length, that is, the total number of descriptors in ring span in the following registers of a DMA channel:

- Transmit Descriptor Ring Length Register (DMA_CH#_TxDesc_Ring_Length)
- Receive Descriptor Ring Length Register (DMA_CH#_RxDesc_Ring_Length)

The Descriptor Tail Pointer Register contains the pointer to the descriptor address (N). The base address and the current descriptor pointer decide the address of the current descriptor that the DMA can process.
The descriptors up to one location less than the one indicated by the descriptor tail pointer (N – 1) are owned by the DMA. The DMA continues to process the descriptors until the following condition occurs:

Current Descriptor Pointer == Descriptor Tail Pointer

The DMA goes into the Suspend mode when this condition occurs. The application must perform a write to the Descriptor Tail pointer register and update the tail pointer so that the following condition is true:

Current Descriptor Pointer < Descriptor Tail Pointer

The DMA automatically wraps around the base address when the end of ring is reached.
For descriptors owned by the application, the OWN bit of DES3 is reset to 0. For descriptors owned by the DMA, the OWN bit is set to 1. If the application has only one descriptor in the beginning, the application sets the last descriptor address (tail pointer) to Descriptor Base Address + 1. The DMA processes the first descriptor and then waits for the application to advance the tail pointer.

30.6.2 Transmit Descriptors

The DMA in GMAC requires at least one descriptor for a transmit packet. In addition to two buffers, two byte-count buffers, and two address pointers, the transmit descriptor has control fields which can be used to control the MAC operation on per-transmit packet basis.
The Transmit Normal descriptor has two formats: Read format and Write-Back format.

30.6.3 Transmit Normal Descriptor (Read Format)

Table 30-5 TDES0 Normal Descriptor (Read Format)

Bit	Name	Description
31:0	BUF1AP	Buffer 1 Address Pointer or TSO Header Address Pointer These bits indicate the physical address of Buffer 1. These bits indicate the TSO Header Address pointer when the following bits are set: ■ TSE bit of TDES3 ■ FD bit of TDES3

Table 30-6 TDES1 Normal Descriptor (Read Format)

Bit	Name	Description
31:0	BUF2AP	Buffer 2 or Buffer 1 Address Pointer This bit indicates the physical address of Buffer 2 when a descriptor ring structure is used. There is no limitation for the buffer address alignment. In 40- or 48-bit addressing mode, these bits indicate the most-significant 8- or 16- bits of the Buffer 1 Address Pointer.

Table 30-7 TDES2 Normal Descriptor (Read Format)

Bit	Name	Description
31	IOC	Interrupt on Completion This bit controls the setting of TI and ETI status bits in the DMA_CH#_Status register. When ETIC = 1 and TDES2[LD] = 0, this bit sets the ETI bit. When TDES3[LD] = 1, this bit sets the TI status bit.
30	TTSE/T MWD	Transmit Timestamp Enable or External TSO Memory Write Enable This bit enables the IEEE1588 time stamping for Transmit packet referenced by the descriptor, if TSE bit is not set. If TSE bit is set and external TSO memory is enabled, setting this bit disables external TSO memory writing for this packet.

Bit	Name	Description
29:16	B2L	Buffer 2 Length The driver sets this field. When set, this field indicates Buffer 2 length
15:14	VTIR	VLAN Tag Insertion or Replacement These bits request the MAC to perform VLAN tagging or untagging before transmitting the packets. The application must set the CRC Pad Control bits appropriately when VLAN Tag Insertion, Replacement, or Deletion is enabled for the packet. The following list describes the values of these bits: ■ 2'b00: Do not add a VLAN tag. ■ 2'b01: Remove the VLAN tag from the packets before transmission. This option should be used only with the VLAN packets. ■ 2'b10: Insert a VLAN tag with the tag value programmed in the MAC_VLAN_Incl register or context descriptor. ■ 2'b11: Replace the VLAN tag in packets with the tag value programmed in the MAC_VLAN_Incl register or context descriptor. This option should be used only with the VLAN packets. These bits are valid when the Enable SA and VLAN Insertion on Tx option is selected while configuring the core
13:0	HL or B1L	Header Length or Buffer 1 Length For Header length only bits [9:0] are taken. The size 13:0 is applicable only when interpreting buffer 1 length. If the TCP Segmentation Offload feature is enabled through the TSE bit of TDES3, this field is equal to the header length. When the TSE bit is set in TDES3, the header length includes the length in bytes from Ethernet Source address till the end of the TCP header. The maximum header length supported for TSO feature is 1023 bytes. The maximum header length supported for TSO feature is 1023 bytes. If the TCP Segmentation Offload feature is not enabled, this field is equal to Buffer 1 length.

Table 30-8 TDES3 Normal Descriptor (Read Format)

Bit	Name	Description
31	OWN	Own Bit When this bit is set, it indicates that the DMA owns the descriptor. When this bit is reset, it indicates that the application owns the descriptor. The DMA clears this bit after it completes the transfer of data given in the associated buffer(s).
30	CTXT	Context Type This bit should be set to 1'b0 for normal descriptor.
29	FD	First Descriptor When this bit is set, it indicates that the buffer contains the first segment of a packet.
28	LD	Last Descriptor When this bit is set, it indicates that the buffer contains the last segment of the packet. When this bit is set, the B1L or B2L field should have a non-zero value.
27:26	CPC	CRC Pad Control

Bit	Name	Description
		This field controls the CRC and Pad Insertion for Tx packet. This field is valid only when the first descriptor bit (TDES3[29]) is set. The following list describes the values of Bits[27:26]: ■ 2'b00: CRC and Pad Insertion. The MAC appends the cyclic redundancy check (CRC) at the end of the transmitted packet of length greater than or equal to 60 bytes. The MAC automatically appends padding and CRC to a packet with length less than 60 bytes. ■ 2'b01: CRC Insertion (Disable Pad Insertion). The MAC appends the CRC at the end of the transmitted packet but it does not append padding. The application should ensure that the padding bytes are present in the packet being transferred from the Transmit Buffer, that is, the packet being transferred from the Transmit Buffer is of length greater than or equal to 60 bytes. ■ 2'b10: Disable CRC Insertion. The MAC does not append the CRC at the end of the transmitted packet. The application should ensure that the padding and CRC bytes are present in the packet being transferred from the Transmit Buffer. ■ 2'b11: CRC Replacement. The MAC replaces the last four bytes of the transmitted packet with recalculated CRC bytes. The application should ensure that the padding and CRC bytes are present in the packet being transferred from the Transmit Buffer. This field is valid only for the first descriptor. Note: When the TSE bit is set, the MAC ignores this field because the CRC and pad insertion is always done for segmentation.
25:23	SAIC	SA Insertion Control These bits request the MAC to add or replace the Source Address field in the Ethernet packet with the value given in the MAC Address 0 register. The application must set the CRC Pad Control bits appropriately when SA Insertion Control is enabled for the packet. Bit 25 specifies the MAC Address Register (1 or 0) value that is used for Source Address insertion or replacement. The following list describes the values of Bits[24:23]: ■ 2'b00: Do not include the source address ■ 2'b01: Include or insert the source address. For reliable transmission, the application must provide frames without source addresses. ■ 2'b10: Replace the source address. For reliable transmission, the application must provide frames with source addresses. ■ 2'b11: Reserved These bits are valid in the EQOS-DMA, EQOS-AXI, and EQOS-AHB configurations when the Enable SA and VLAN Insertion on Tx option is selected while configuring the core and when the First Segment control bit (TDES3 [29]) is set. This field is valid only for the first descriptor.
22:19	SLOTNUM or THL	SLOTNUM: Slot Number Control Bits in AV Mode These bits indicate the slot interval in which the data should be fetched from the corresponding buffers

Bit	Name	Description
		addressed by TDES0 or TDES1. When the Transmit descriptor is fetched, the DMA compares the slot number value in this field with the slot interval maintained in the RSN field DMA_CH#_Slot_Function_Control_Status. It fetches the data from the buffers only if a value matches. These bits are valid only for the AV channels. THL: TCP/UDP Header Length If the TSE bit is set, this field contains the length of the TCP/UDP header. The minimum value of this field must be 5 for TCP header. The value must be equal to 2 for UDP header. This field is valid only for the first descriptor.
18	TSE	TCP Segmentation Enable When this bit is set, the DMA performs the TCP/UDP segmentation or UDP fragmentation for a packet depending on the TSE_MODE[1:0] bit of the DMA_CH(#i)_Tx_Control Register. This bit is valid only if the FD bit is set.
17:16	CIC/TPL	Checksum Insertion Control or TCP Payload Length These bits control the checksum calculation and insertion. The following list describes the bit encoding: ■ 2'b00: Checksum Insertion Disabled. ■ 2'b01: Only IP header checksum calculation and insertion are enabled. ■ 2'b10: IP header checksum and payload checksum calculation and insertion are enabled, but pseudo-header checksum is not calculated in hardware. ■ 2'b11: IP Header checksum and payload checksum calculation and insertion are enabled, and pseudo-header checksum is calculated in hardware. This field is valid when the Enable Transmit TCP/IP Checksum Offload option is selected and the TSE bit is reset. When the TSE bit is set, this field contains the upper bits [17:16] of the TCP Payload (or IP Payload for UDP fragmentation). This allows the TCP/UDP packet length field to be spanned across TDES3[17:0] to provide 256 KB packet length support. This field is valid only for the first descriptor.
15	TPL	Reserved or TCP Payload Length When the TSE bit is reset, this bit is reserved. When the TSE bit is set, this is Bit 15 of the TCP payload length [17:0]. This field is valid only when the Enable TCP Segmentation Offloading for TCP/IP Packets option is selected while configuring the core.
14:0	FL/TPL	Frame Length or TCP Payload Length This field is equal to the length of the packet to be transmitted in bytes. When the TSE bit is not set, this field is equal to the total length of the packet to be transmitted: Ethernet Header Length + TCP /IP Header Length – Preamble Length – SFD Length + Ethernet Payload Length When the TSE bit is set, this field is equal to the lower 15 bits of the TCP payload length in case of segmentation and IP payload in case of UDP fragmentation. In case of segmentation, this length does not include Ethernet

Bit	Name	Description
		header or TCP/UDP/IP header length. In case of fragmentation, this length does not include Ethernet header and IP header. When DWRR/WFQ algorithm is NOT enabled, value written into this field is not used when TSE = 0.

30.6.4 Transmit Normal Descriptor (Write-back Format)

The write-back format of the Transmit Descriptor includes timestamp low, timestamp high, OWN, and Status bits. The write-back format is applicable only for the last descriptor of the corresponding packet. The LD bit (TDES3[28]) is set in the descriptor where the DMA writes back the status and timestamp information for the corresponding Transmit packet.
this format is only applicable to the last descriptor of a packet.

Table 30-9 TDES0 Normal Descriptor (Write-Back Format)

Bit	Name	Description
31:0	TTSL	Transmit Packet Timestamp Low The DMA updates this field with least significant 32 bits of the timestamp captured for the corresponding Transmit packet. The DMA writes the timestamp only if TTSE bit of TDES2 is set in the first descriptor of the packet. This field has the timestamp only if the Last Segment bit (LS) in the descriptor is set and the Timestamp status (TTSS) bit is set

Table 30-10 TDES1 Normal Descriptor (Write-Back Format)

Bit	Name	Description
31:0	TTSH	Transmit Packet Timestamp High The DMA updates this field with the most significant 32 bits of the timestamp captured for corresponding transmit packet. The DMA writes the timestamp only if the TTSE bit of TDES2 is set in the first descriptor of the packet. This field has the timestamp only if the Last Segment bit (LS) in the descriptor is set and Timestamp status (TTSS) bit is set.

Table 30-11 TDES2 Normal Descriptor (Write-Back Format)

Bit	Name	Description
31:0	Rsvd	Reserved

Table 30-12 TDES2 Normal Descriptor (Write-Back Format)

Bit	Name	Description
31	OWN	Own Bit When this bit is set, it indicates that the DMA owns the descriptor. When this bit is reset, it indicates that the application owns the descriptor. The DMA clears this bit after it completes the transfer of data given in the associated buffer(s).
30	CTXT	Context Type This bit should be set to 1'b0 for normal descriptor.
29	FD	First Descriptor When this bit is set, it indicates that the buffer contains the first segment of a packet.
28	LD	Last Descriptor When this bit is set, it indicates that the buffer contains the last segment of the packet. When this bit is set, the B1L or B2L field should have a non-zero value.
27:24	Rsvd	Reserved
23	DE	Descriptor Error

Bit	Name	Description
		When this bit is set, it indicates that the descriptor content is incorrect. The DMA sets this bit during write-back while closing the descriptor. Descriptor Errors can be: ■ Incorrect sequence from the context descriptor. For example, a location after the first descriptor for a packet ■ All 1s ■ CTXT, LD, and FD bits set to 1 Note 1: When Descriptor Error occurs due to All 1s or CTXT, LD, and FD bits set to 1, the Transmit DMA closes the transmit descriptor with DE and LD bits set to 1. When IOC bit in TDES2 of corresponding first descriptor is set to 1, Transmit DMA will set the TI bit in the DMA_CH#_Status register. Note 2: Based on CTXT, LD, and FD bits of the transmit descriptor, the subsequent descriptor might be considered as the First Descriptor (even if FD bit is not set) and partial packet is sent.
22:18	Rsvd	Reserved
17	TTSS	Tx Timestamp Status This status bit indicates that a timestamp has been captured for the corresponding transmit packet. When this bit is set, TDES0 and TDES1 have timestamp values that were captured for the Transmit packet. This field is valid only when the Last Segment control bit (TDES3 [28]) in a descriptor is set. This bit is valid only when IEEE1588 timestamping feature is enabled; otherwise, it is reserved.
16	EUE	ECC Uncorrectable Error Status Indicates the ECC uncorrectable error in the TSO memory. Note: Uncorrectable error in Transmit FIFO memory is reported with (Bit 13) FF = 1. This is because, all such packets are flushed by GMAC.
15	ES	Error Summary This bit indicates the logical OR of the following bits: ■ TDES3[0]: IP Header Error ■ TDES3[14]: Jabber Timeout TDES3[13]: Packet Flush ■ TDES3[12]: Payload Checksum Error ■ TDES3[11]: Loss of Carrier ■ TDES3[10]: No Carrier ■ TDES3[9]: Late Collision ■ TDES3[8]: Excessive Collision ■ TDES3[3]: Excessive Deferral ■ TDES3[2]: Underflow Error This bit is also set when EUE (bit 16) is set
14	JT	Jabber Timeout This bit indicates that the MAC transmitter has experienced a jabber time-out. This bit is set only when the JD bit of the MAC_Configuration register is not set.
13	PF	Packet Flushed This bit indicates that the DMA or MTL flushed the packet because of a software flush command given by the CPU.
12	PCE	Payload Checksum Error This bit indicates that the Checksum Offload engine had a failure and did not insert any checksum into the

Bit	Name	Description
		encapsulated TCP, UDP, or ICMP payload. This failure can be either because of insufficient bytes, as indicated by the Payload Length field of the IP Header or the MTL starting to forward the packet to the MAC transmitter in Store-and-Forward mode without the checksum having been calculated yet. This second error condition only occurs when the Transmit FIFO depth is less than the length of the Ethernet packet being transmitted to avoid deadlock, the MTL starts forwarding the packet when the FIFO is full, even in the store-and-forward mode. This error can also occur when Bus Error is detected during packet transfer. When the Full Checksum Offload engine is not enabled, this bit is reserved.
11	LoC	Loss of Carrier This bit indicates that Loss of Carrier occurred during packet transmission (that is, the gmii_crs_i signal was inactive for one or more transmit clock periods during packet transmission). This is valid only for the packets transmitted without collision and when the MAC operates in the half-duplex mode.
10	NC	No Carrier This bit indicates that the carrier sense signal form the PHY was not asserted during transmission.
9	LC	Late Collision This bit indicates that packet transmission was aborted because a collision occurred after the collision window (64 byte times including Preamble in MII mode and 512 byte times including Preamble and Carrier Extension in GMII mode). This bit is not valid if Underflow Error is set.
8	EC	Excessive Collision This bit indicates that the transmission was aborted after 16 successive collisions while attempting to transmit the current packet. If the DR bit is set in the MAC_Configuration register, this bit is set after first collision and the transmission of the packet is aborted.
7:4	CC	Collision Count This 4-bit counter value indicates the number of collisions occurred before the packet was transmitted. The count is not valid when the EC bit is set.
3	ED	Excessive Deferral This bit indicates that the transmission ended because of excessive deferral of over 24,288 bit times (155,680 bits times in 1000 Mbps mode or Jumbo Packet enabled mode) if DC bit is set in the MAC_Configuration register. When TBS is enabled in full duplex mode and this bit is set, it indicates that the frame has been dropped after the expiry time has reached.
2	UF	Underflow Error This bit indicates that the MAC aborted the packet because the data arrived late from the system memory. The underflow error can occur because of either of the following conditions: ■ The DMA encountered an empty Transmit Buffer while transmitting the packet ■ The application filled the MTL Tx FIFO slower than the MAC transmit rate

Bit	Name	Description
		The transmission process enter the suspended state and sets the underflow bit corresponding to a queue in the MTL_Interrupt_Status register.
1	DB	Deferred Bit This bit indicates that the MAC deferred before transmitting because of presence of carrier. This bit is valid only in the half-duplex mode.
0	IHE	IP Header Error When IP Header Error is set, this bit indicates that the Checksum Offload engine detected an IP header error. This bit is valid only when Tx Checksum Offload is enabled. Otherwise, it is reserved. If COE detects an IP header error, it still inserts an IPv4 header checksum if the Ethernet Type field indicates an IPv4 payload. In full duplex mode, when EST/Qbv is enabled and this bit is set, it indicates the frame drop status due to Frame Size error or Schedule Error.

30.6.5 Transmit Context Descriptor

The context descriptor is used to provide the timestamps for one-step timestamp correction, VLAN Tag ID for VLAN insertion feature. The Transmit Context descriptor can be provided any time before a packet descriptor. The context is valid for the current packet and subsequent packets. The context descriptor is used to provide the timestamps for one-step timestamp correction and VLAN Tag ID for VLAN insertion feature. Write back is done on a context descriptor only to reset the OWN bit

Table 30-13 TDES0 Context Descriptor

Bit	Name	Description
31:0	TTSL	Transmit Packet Timestamp Low For one-step correction, the driver can provide the lower 32 bits of timestamp in this descriptor word. The DMA uses this value as the low word for doing one-step timestamp correction. This field is valid only if the OSTC and TCMSSV bits of TDES3 context descriptor are set.

Table 30-14 TDES1 Context Descriptor

Bit	Name	Description
31:0	TTSH	Transmit Packet Timestamp High For one-step correction, the driver can provide the upper 32 bits of timestamp in this descriptor. The DMA uses this value as the high word for doing one-step timestamp correction. This field is valid only if the OSTC and TCMSSV bits of TDES3 context descriptor are set.

Table 30-15 TDES2 Context Descriptor

Bit	Name	Description
31:16	IVT	Inner VLAN Tag When the IVLTV bit of TDES3 context descriptor is set and the TCMSSV and OSTC bits of TDES3 context descriptor are reset, TDES2[31:16] contains the inner VLAN Tag to be inserted in the subsequent Transmit packets.
15:14	Rsvd	Reserved
13:0	MSS	Maximum Segment Size When the Enable TCP Segmentation Offloading for TCP/IP Packets option is selected, the driver can provide maximum

Bit	Name	Description
		segment size in this field. This segment size is used while segmenting the TCP/IP payload. This field is valid only if the TCMSSV bit of TDES3 context descriptor is set and the OSTC bit of the TDES3 context descriptor is reset.

Table 30-16 TDES3 Context Descriptor

Bit	Name	Description
31	OWN	Own Bit When this bit is set, it indicates that the GMAC DMA owns the descriptor. When this bit is reset, it indicates that the application owns the descriptor. The DMA clears this bit immediately after the read.
30	CTXT	Context Type This bit should be set to 1'b1 for Context descriptor.
29:28	Rsvd	Reserved
27	OSTC	One-Step Timestamp Correction Enable When this bit is set, the DMA performs a one-step timestamp correction with reference to the timestamp values provided in TDES0 and TDES1.
26	TCMSSV	One-Step Timestamp Correction Input or MSS Valid When this bit and the OSTC bit are set, it indicates that the Timestamp Correction input provided in TDES0 and TDES1 is valid. When the OSTC bit is reset and this bit and the TSE bit of TDES3 are set in subsequent normal descriptor, it indicates that the MSS input in TDES2 is valid.
25:24	Rsvd	Reserved
23	DE	Descriptor Error When this bit is set, it indicates that the descriptor content is incorrect.The DMA sets this bit during write-back while closing the context descriptor. Descriptor Errors can be: ■ Incorrect sequence from the context descriptor. For example, a location before the first descriptor for a packet ■ All 1s ■ CD, LD, and FD bits set to 1 Note 1: When Descriptor Error occurs due to All 1s or CTXT, LD, and FD bits set to 1, the Transmit DMA closes the transmit descriptor with DE and LD bits set to 1. When IOC bit in TDES2 of corresponding first descriptor is set to 1, Transmit DMA will set the TI bit in the DMA_CH#_Status Register. Note 2: Based on CTXT, LD, and FD bits of the transmit descriptor, the subsequent descriptor might be considered as the First Descriptor (even if FD bit is not set) and partial packet is sent.
22:20	Rsvd	Reserved
19:18	IVTIR	Inner VLAN Tag Insert or Replace When this bit is set, these bits request the MAC to perform Inner VLAN tagging or un-tagging before transmitting the packets. If the packet is modified for VLAN tags, the MAC automatically recalculates and replaces the CRC bytes. The following list describes the values of these bits: ■ 2'b00: Do not add the inner VLAN tag. ■ 2'b01: Remove the inner VLAN tag from the packets before transmission. This option should be used only with the VLAN frames.

Bit	Name	Description
		■ 2'b10: Insert an inner VLAN tag with the tag value programmed in the MAC_Inner_VLAN_Incl register or context descriptor. ■ 2'b11: Replace the inner VLAN tag in packets with the tag value programmed in the MAC_Inner_VLAN_Incl register or context descriptor. This option should be used only with the VLAN frames. These bits are valid when the Enable SA and VLAN Insertion on Tx and Enable Double VLAN Processing options are selected.
17	IVLTV	Inner VLAN Tag Valid When this bit is set, it indicates that the IVT field of TDES2 is valid.
16	VLTV	VLAN Tag Valid When this bit is set, it indicates that the VT field of TDES3 is valid.
15:0	VT	VLAN Tag This field contains the VLAN Tag to be inserted or replaced in the packet. This field is used as VLAN Tag only when the VLTI bit of the MAC_VLAN_Incl register is reset.

30.6.6 Receive Descriptors

The DMA in GMAC attempts to read a descriptor only if the Tail Pointer is different from the Base Pointer or current pointer. It is recommended to have a descriptor ring with a length that can accommodate at least two complete packets received by the MAC. Otherwise, the performance of the DMA is impacted greatly because of the unavailability of the descriptors. In such situations, the RxFIFO in MTL becomes full and starts dropping packets.
The following Receive Descriptors are present:

- Normal descriptors
- Context descriptors

All RX descriptors are prepared by the software and given to the DMA as "Normal" Descriptors with the content as shown in Receive Normal Descriptor (Read Format). The DMA reads this descriptor and after transferring a received packet (or part of) to the buffers indicated by the descriptor, the Rx DMA will close the descriptor with the corresponding packet status. The format of this status is given in the "Receive Normal Descriptor (Write-Back Format)". For some packets, the normal descriptor bits are not enough to write the complete status. For such packets, the RX DMA writes the extended status to the next descriptor (without processing or using the Buffers/ Pointers embedded in that descriptor). The format and content of the descriptor write back is described in "Receive Context Descriptor".

30.6.7 Receive Normal Descriptors(Read Format)

Table 30-17 TDES0 Normal Descriptor(Read Format)

Bit	Name	Description
31:0	BUF1AP	Header or Buffer 1 Address Pointer When the SPH bit of Control register of a channel is reset, these bits indicate the physical address of Buffer 1. When the SPH bit is set, these bits indicate the physical address of Header Buffer where the Rx DMA writes the L2/L3/L4 header bytes of the received packet. The application can program a byte-aligned address for this buffer which means that the LS bits of this field can be non-zero. However, while transferring the start of packet, the DMA performs a Write operation with RDES0[1:0] (or RDES0[2:0]/[3:0] in case of 64-/128-bit configuration) as zero. However, the packet data is shifted as per actual offset as given by buffer address pointer.

Bit	Name	Description
		If the address pointer points to a buffer where the middle or last part of the packet is stored, the DMA ignores the offset address and writes to the full location as indicated by the data-width.

Table 30-18 TDES1 Normal Descriptor(Read Format)

Bit	Name	Description
31:0	Reserved or BUF1AP	In 64-bit addressing mode, this field contains the most-significant 32 bits of the Buffer 1 Address Pointer. Otherwise, this field is reserved.

Table 30-19 TDES2 Normal Descriptor(Read Format)

Bit	Name	Description
31:0	Reserved or BUF1AP	In 64-bit addressing mode, this field contains the most-significant 32 bits of the Buffer 1 Address Pointer. Otherwise, this field is reserved.

Table 30-20 TDES3 Normal Descriptor(Read Format)

Bit	Name	Description
31	OWN	Own Bit When this bit is set, it indicates that the GMAC DMA owns the descriptor. When this bit is reset, it indicates that the application owns the descriptor. The DMA clears this bit when either of the following conditions is true: ■ The DMA completes the packet reception ■ The buffers associated with the descriptor are full
30	IOC	Interrupt Enabled on Completion When this bit is set, an interrupt is issued to the application when the DMA closes this descriptor.
29:26	Rsvd	Reserved
25	BUF2V	Buffer 2 Address Valid When this bit is set, it indicates to the DMA that the buffer 2 address specified in RDES2 is valid. The application must set this bit so that the DMA can use the address, to which the Buffer 2 address in RDES2 is pointing, to write received packet data.
24	BUF1V	Buffer 1 Address Valid When set, this indicates to the DMA that the buffer 1 address specified in RDES1 is valid. The application must set this value if the address pointed to by Buffer 1 address in RDES1 can be used by the DMA to write received packet data.
23:0	Rsvd	Reserved

30.6.8 Receive Normal Descriptors(Write-Back Format)

Table 30-21 TDES0 Normal Descriptor(Write-Back Format)

Bit	Name	Description
31:16	IVT	Inner VLAN Tag This field contains the Inner VLAN tag of the received packet if the RS0V bit of RDES3 is set. This is valid only when Double VLAN tag processing and VLAN tag stripping are enabled.
15:0	OVT	Outer VLAN Tag

Bit	Name	Description
		This field contains the Outer VLAN tag of the received packet if the RS0V bit of RDES3 is set.

Table 30-22 TDES1 Normal Descriptor(Write-Back Format)

Bit	Name	Description
31:16	OPC	OAM Sub-Type Code, or MAC Control Packet opcode OAM Sub-Type Code If Bits[18:16] of RDES3 are set to 3'b111, this field contains the OAM sub-type and code fields. MAC Control Packet opcode If Bits[18:16] of RDES3 are set to 3'b110, this field contains the MAC Control packet opcode field.
15	TD	Timestamp Dropped This bit indicates that the timestamp was captured for this packet but it got dropped in the MTL Rx FIFO because of overflow. This bit is available only when you select the Timestamp feature. Otherwise, this bit is reserved.
14	TSA	Timestamp Available When Timestamp is present, this bit indicates that the timestamp value is available in a context descriptor word 2 (RDES2) and word 1(RDES1). This is valid only when the Last Descriptor bit (RDES3 [28]) is set. The context descriptor is written in the next descriptor just after the last normal descriptor for a packet.
13	PV	PTP Version This bit indicates that the received PTP message has the IEEE 1588 version 2 format. When this bit is reset, it indicates the IEEE 1588 version 1 format. This bit is available only when you select the Timestamp feature. Otherwise, this bit is reserved.
12	PFT	PTP Packet Type This bit indicates that the PTP message is sent directly over Ethernet. This bit is available only when you select the Timestamp feature. Otherwise, this bit is reserved.
11:8	PMT	PTP Message Type These bits are encoded to give the type of the message received: ■ 0000: No PTP message received ■ 0001: SYNC (all clock types) ■ 0010: Follow_Up (all clock types) ■ 0011: Delay_Req (all clock types) ■ 0100: Delay_Resp (all clock types) ■ 0101: Pdelay_Req (in peer-to-peer transparent clock)

Bit	Name	Description
		■ 0110: Pdelay_Resp (in peer-to-peer transparent clock) ■ 0111: Pdelay_Resp_Follow_Up (in peer-to-peer transparent clock) ■ 1000: Announce ■ 1001: Management ■ 1010: Signaling ■ 1011–1110: Reserved ■ 1111: PTP packet with Reserved message type These bits are available only when you select the Timestamp feature.
7	IPCE	IP Payload Error When this bit is set, it indicates either of the following: ■ The 16-bit IP payload checksum (that is, the TCP, UDP, or ICMP checksum) calculated by the MAC does not match the corresponding checksum field in the received segment. ■ The TCP, UDP, or ICMP segment length does not match the payload length value in the IP Header field. ■ The TCP, UDP, or ICMP segment length is less than minimum allowed segment length for TCP, UDP, or ICMP. Bit 15 (ES) of RDES3 is not set when this bit is set.
6	IPCB	IP Checksum Bypassed This bit indicates that the checksum offload engine is bypassed. This bit is available when you select the Enable Receive TCP/IP Checksum Check feature.
5	IPV6	IPv6 header Present This bit indicates that an IPV6 header is detected. When the Enable Split Header Feature option is selected and the SPH bit of Control Register of a channel is set, the IPV6 header is available in the header buffer area to which RDES0 is pointing.
4	IPV4	IPV4 Header Present This bit indicates that an IPV4 header is detected. When the SPH bit of RDES3 is set, the IPV4 header is available in the header buffer area to which RDES0 is pointing.
3	IPHE	IP Header Error When this bit is set, it indicates either of the following: ■ The 16-bit IPv4 header checksum calculated by the MAC does not match the received checksum bytes. ■ The IP datagram version is not consistent with the Ethernet Type value. ■ Ethernet packet does not have the expected number of IP header bytes. This bit is valid when either Bit 5 or Bit 4 is set. This bit is available when you select the

Bit	Name	Description
		Enable Receive TCP/IP Checksum Check feature.
2:0	PT	Payload Type These bits indicate the type of payload encapsulated in the IP datagram processed by the Receive Checksum Offload Engine (COE): ■ 3'b000: Unknown type or IP/AV payload not processed ■ 3'b001: UDP ■ 3'b010: TCP ■ 3'b011: ICMP ■ 3'b110: AV Tagged Data Packet ■ 3'b111: AV Tagged Control Packet ■ 3'b101: AV Untagged Control Packet ■ 3'b100: IGMP if IPV4 Header Present bit is set else DCB (LLDP) Control Packet If the COE does not process the payload of an IP datagram because there is an IP header error or fragmented IP, it sets these bits to 3'b000.

Table 30-23 TDES2 Normal Descriptor(Write-Back Format)

Bit	Name	Description
31:29	L3L4FM	Layer 3 and Layer 4 Filter Number Matched These bits indicate the number of the Layer 3 and Layer 4 Filter that matched the received packet: ■ 000: Filter 0 ■ 001: Filter 1 ■ 010: Filter 2 ■ 011: Filter 3 ■ 100: Filter 4 ■ 101: Filter 5 ■ 110: Filter 6 ■ 111: Filter 7 This field is valid only when Bit 28 or Bit 27 is set high. When more than one filter matches, these bits give the number of lowest filter. Note: This status is not available when Flexible RX Parser is enabled.
28	L4FM	Layer 4 Filter Match When this bit is set, it indicates that the received packet matches one of the enabled Layer 4 Port Number fields. This status is given only when one of the following conditions is true: ■ Layer 3 fields are not enabled and all enabled Layer 4 fields match ■ All enabled Layer 3 and Layer 4 filter fields match When more than one filter matches, this bit gives the layer 4 filter status of filter indicated by Bits[31:29]. Note: This status is not available when Flexible RX Parser is enabled.
27	L3FM	Layer 3 Filter Match When this bit is set, it indicates that the received packet matches one of the enabled Layer 3 IP Address fields. This status is given only when one of the following conditions is true:

Bit	Name	Description
		■ All enabled Layer 3 fields match and all enabled Layer 4 fields are bypassed ■ All enabled filter fields match When more than one filter matches, this bit gives the layer 3 filter status of filter indicated by Bits[31:29]. Note: This status is not available when Flexible RX Parser is enabled.
26:19	MADRM	MAC Address Match or Hash Value When the HF bit is reset, this field contains the MAC address register number that matched the Destination address of the received packet. This field is valid only if the DAF bit is reset. When the HF bit is set, this field contains the hash value computed by the MAC. A packet passes the hash filter when the bit corresponding to the hash value is set in the hash filter register. Note: This status is not available when Flexible RX Parser is enabled.
18	HF	Hash Filter Status When this bit is set, it indicates that the packet passed the MAC address hash filter. Bits[26:19] indicate the hash value. Note: This status is not available when Flexible RX Parser is enabled.
17	DAF/RXPI	Destination Address Filter Fail When Flexible RX Parser is disabled, and this bit is set, it indicates that the packet failed the DA Filter in the MAC. When Flexible RX Parser is enabled, this bit is set to indicate that the packet parsing is incomplete (RXPI) due to ECC error. Note: When this bit is set, ES bit of RDES3 is also set.
16	SAF/RXPD	SA Address Filter Fail When Flexible RX Parser is disabled, and this bit is set, it indicates that the packet failed the SA Filter in the MAC. When Flexible RX Parser is enabled, this bit is set to indicate that the packet is dropped (RXPD) by the parser. Note: When this bit is set, ES bit of RDES3 is also set.
15	OTS	VLAN Filter Status When set, this bit indicates that the VLAN Tag of the received packed passed the VLAN filter. This bit is valid only when DWC_EQOS_ERVFE is not enabled. If DWC_EQOS_ERVFE is enabled, the bit is redefined as Outer VLAN Tag Filter Status (OTS). This bit is valid for both Single and Double VLAN Tagged frames.
14	ITS	Inner VLAN Tag Filter Status (ITS) This bit is valid only when DWC_EQOS_ERVFE is enabled. This bit is valid only for Double VLAN Tagged frames, when Double VLAN Processing is enabled. For more information, see the Filter Status topic.
13:11	Rsvd	Reserved
10	ARPNR	ARP Reply Not Generated When this bit is set, it indicates that the MAC did not generate the ARP Reply for received ARP Request packet. This bit is set when the MAC is busy transmitting ARP reply

Bit	Name	Description
		to earlier ARP request (only one ARP request is processed at a time). This bit is reserved when the Enable IPv4 ARP Offload option is not selected.
9:0	HL	L3/L4 Header Length This field contains the length of the header of the packet split by the MAC at L3 or L4 header. boundary as identified by the MAC receiver. This field is valid only when the first descriptor bit is set (FD = 1). The header data is written to the Buffer 1 address of corresponding descriptor. If header length is zero, this field is not valid. It implies that the MAC did not identify and split the header. This field is valid when the Enable Split Header Feature option is selected.

Table 30-24 TDES3 Normal Descriptor(Write-Back Format)

Bit	Name	Description
31	OWN	Own Bit When this bit is set, it indicates that the GMAC DMA owns the descriptor. When this bit is reset, it indicates that the application owns the descriptor. The DMA clears this bit when either of the following conditions is true: ■ The DMA completes the packet reception ■ The buffers associated with the descriptor are full
30	CTXT	Receive Context Descriptor When this bit is set, it indicates that the current descriptor is a context type descriptor. The DMA writes 1'b0 to this bit for normal receive descriptor. When CTXT and FD bits are used together, {CTXT, FD} ■ 2’b00: Intermediate Descriptor ■ 2’b01: First Descriptor ■ 2’b10: Reserved ■ 2’b11: Descriptor Error (due to all 1s) Note: When Descriptor Error occurs, the Receive DMA closes the receive descriptor indicating Descriptor Error. This receive descriptor is skipped and the buffer addresses are not used to write the packet data Receive DMA will set the CDE bit in DMA_CH#_Status register but not the RI bit even when IOC is set, as this is not marked as last receive descriptor for the packet. The subsequent valid receive descriptor is used to write the packet data.
29	FD	First Descriptor When this bit is set, it indicates that this descriptor contains the first buffer of the packet. If the size of the first buffer is 0, the second buffer contains the beginning of the packet. If the size of the second buffer is also 0, the next descriptor contains the beginning of the packet. See the CTXT bit description for details of using the CTXT bit and FD bit together.
28	LD	Last Descriptor When this bit is set, it indicates that the buffers to which this descriptor is pointing are the last buffers of the packet.
27	RS2V	Receive Status RDES2 Valid When this bit is set, it indicates that the status in RDES2 is valid and it is written by the DMA. This bit is valid only when the LD bit of RDES3 is set.

Bit	Name	Description
26	RS1V	Receive Status RDES1 Valid When this bit is set, it indicates that the status in RDES1 is valid and it is written by the DMA. This bit is valid only when the LD bit of RDES3 is set.
25	RS0V	Receive Status RDES0 Valid When this bit is set, it indicates that the status in RDES0 is valid and it is written by the DMA. This bit is valid only when the LD bit of RDES3 is set.
24	CE	CRC Error When this bit is set, it indicates that a Cyclic Redundancy Check (CRC) Error occurred on the received packet. This field is valid only when the LD bit of RDES3 is set.
23	GP	Giant Packet When this bit is set, it indicates that the packet length exceeds the specified maximum Ethernet size of 1518, 1522, or 2000 bytes (9018 or 9022 bytes if jumbo packet enable is set). Note: Giant packet indicates only the packet length. It does not cause any packet truncation.
22	RWT	Receive Watchdog Timeout When this bit is set, it indicates that the Receive Watchdog Timer has expired while receiving the current packet. The current packet is truncated after watchdog timeout.
21	OE	Overflow Error When this bit is set, it indicates that the received packet is damaged because of buffer overflow in Rx FIFO. Note: This bit is set only when the DMA transfers a partial packet to the application. This happens only when the Rx FIFO is operating in the threshold mode. In the store-and-forward mode, all partial packets are dropped completely in Rx FIFO.
20	RE	Receive Error When this bit is set, it indicates that the gmii_rxer_i signal is asserted while the gmii_rxdv_i signal is asserted during packet reception. This error also includes carrier extension error in the GMII and half-duplex mode. Error can be of less or no extension, or error (rxd!= 0f) during extension.
19	DE	Dribble Bit Error When this bit is set, it indicates that the received packet has a non-integer multiple of bytes (odd nibbles). This bit is valid only in the MII Mode.
18:16	LT	Length/Type Field This field indicates if the packet received is a length packet or a type packet. The encoding of the 3 bits is as follows: ■ 3'b000: The packet is a length packet ■ 3'b001: The packet is a type packet. ■ 3'b011: The packet is a ARP Request packet type ■ 3'b100: The packet is a type packet with VLAN Tag ■ 3'b101: The packet is a type packet with Double VLAN Tag ■ 3'b110: The packet is a MAC Control packet type ■ 3'b111: The packet is a OAM packet type ■ 3'b010: Reserved
15	ES	Error Summary

Bit	Name	Description
		When this bit is set, it indicates the logical OR of the following bits: ■ RDES3[24]: CRC Error ■ RDES3[19]: Dribble Error ■ RDES3[20]: Receive Error ■ RDES3[22]: Watchdog Timeout ■ RDES3[21]: Overflow Error ■ RDES3[23]: Giant Packet ■ RDES2[17]: Destination Address Filter Fail, when Flexible RX Parser is enabled ■ RDES2[16]: SA Address Filter Fail, when Flexible RX Parser is enabled This field is valid only when the LD bit of RDES3 is set.
14	PL	Packet Length These bits indicate the byte length of the received packet that was transferred to system memory (including CRC). This field is valid when the LD bit of RDES3 is set and Overflow Error bits are reset. The packet length also includes the two bytes appended to the Ethernet packet when IP checksum calculation is enabled and the received packet is not a MAC control packet. This field is valid when the LD bit of RDES3 is set. When the Last Descriptor and Error Summary bits are not set, this field indicates the accumulated number of bytes that have been transferred for the current packet.

30.6.9 Receive Context Descriptor

This descriptor is read-only for the application. Only the DMA can write to this descriptor. The context descriptor provides information about the extended status related to the last received packet. The Bit 30 of RDES3 indicates the context type descriptor.

Table 30-25 TDES0 Context Descriptor

Bit	Name	Description
31:0	RSTL	Receive Packet Timestamp Low The DMA updates this field with least significant 32 bits of the timestamp captured for corresponding Receive packet. When this field and the RTSH field of RDES1 show all-ones value, the timestamp must be considered as corrupt.

Table 30-26 TDES1 Context Descriptor

Bit	Name	Description
31:0	RSTH	Receive Packet Timestamp High The DMA updates this field with most significant 32 bits of the timestamp captured for corresponding receive packet. When this field

Bit	Name	Description
		and the RTSL field of RDES0 show all-ones value, the timestamp must be considered as corrupt.

Table 30-27 TDES2 Context Descriptor

Bit	Name	Description
31:0	Rsvd	Reserved

Table 30-28 TDES3 Context Descriptor

Bit	Name	Description
31	OWN	Own Bit When this bit is set, it indicates that the DMA owns the descriptor. When this bit is reset, it indicates that the application owns the descriptor. The DMA clears this bit when either of the following conditions is true: ■ The DMA completes the packet reception ■ The buffers associated with the descriptor are full
30	CTXT	Receive Context Descriptor When this bit is set, it indicates that the current descriptor is a context descriptor. The DMA writes 1'b1 to this bit for context descriptor. DMA writes 2'b11 to indicate a descriptor error due to all 1s. When CTXT and DE bits are used together, {CTXT, DE} ■ 2'b00: Reserved ■ 2'b01: Reserved ■ 2'b10: Context Descriptor ■ 2'b11: Descriptor Error Note: When Descriptor Error occurs, the Receive DMA closes the receive descriptor indicating Descriptor Error. This receive descriptor is skipped and the buffer addresses are not used to write the packet data Receive DMA will set the CDE bit in DMA_CH#_Status register but not the RI bit even when IOC is set, as this is not marked as last receive descriptor for the packet. The subsequent valid receive descriptor is used to write the packet data.
29	DE	Descriptor Error See the CTXT bit description for details of using theDE bit along with CTXT bit.
28:0	Rsvd	Reserved

30.6.10 Programming Guide

30.6.10.1 Initializing DMA

Complete the following steps to initialize the DMA:
1. Provide a software reset. This resets all of the MAC internal registers and logic (bit-0 of DMA_Mode).
2. Wait for the completion of the reset process (poll bit 0 of the DMA_Mode, which is only cleared after the reset operation is completed).
3. Program the following fields to initialize the DMA_SysBus_Mode register:
a. AAL
b. Fixed burst or undefined burst
c. Burst mode values in case of AHB bus interface, OSR_LMT in case of AXI bus interface.
d. If fixed length value is enabled, select the maximum burst length possible on the AXI Bus (bits [7:1])

4. Create a descriptor list for transmit and receive. In addition, ensure that the descriptors are owned by DMA (set bit 31 of descriptor TDES3/RDES3). For more information about descriptors, see section "Descriptors".
5. Program the Transmit and Receive Ring length registers (DMA_CH(#i)_TxDesc_Ring_Length (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1) and DMA_CH(#i)_RxDesc_Ring_Length (for i = 0; i <= DWC_EQOS_NUM_DMA_RX_CH-1)). The ring length programmed must be at least 4.
6. Initialize receive and transmit descriptor list address with the base address of the transmit and receive descriptor (DMA_CH(#i)_TxDesc_List_Address (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1), DMA_CH(#i)_RxDesc_List_Address (for i = 0; i <= DWC_EQOS_NUM_DMA_RX_CH-1)). Also, program transmit and receive tail pointer registers indicating to the DMA about the available descriptors (DMA_CH(#i)_TxDesc_Tail_Pointer (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1) and DMA_CH(#i)_RxDesc_Tail_Pointer (for i = 0; i <= DWC_EQOS_NUM_DMA_RX_CH-1)).
7. Program the settings of the following registers for the parameters like maximum burst-length (PBL) initiated by DMA, descriptor skip lengths, OSP in case of TxDMA, RBSZ in case of RxDMA, and so on:

- DMA_CH(#i)_Control (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1)
- DMA_CH(#i)_TX_Control (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1)
- DMA_CH(#i)_RX_Control (for i = 0; i <= DWC_EQOS_NUM_DMA_RX_CH-1)

8. Enable the interrupts by programming the DMA_CH(#i)_Interrupt_Enable (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1) register.
9. Start the Receive and Transmit DMAs by setting SR (bit 0) of the DMA_CH(#i)_RX_Control (for i = 0; i <= DWC_EQOS_NUM_DMA_RX_CH-1) and ST (bit 0) of the DMA_CH(#i)_TX_Control (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1) register.
10. Repeat steps 4 to 9 for all the Tx DMA and Rx DMA channels selected in the hardware.

30.6.10.2 Initializing MTL Registers

The Transaction Layer (MTL) registers must be initialized to establish the transmit and receive operating modes and commands.
Complete the following steps to initialize the MTL registers:
1. Program the Tx Scheduling (SCHALG) and Receive Arbitration Algorithm (RAA) fields in MTL_Operation_Mode to initialize the MTL operation in case of multiple Tx and Rx queues.
2. Program the Receive Queue to DMA mapping in MTL_RxQ_DMA_Map0 and MTL_RxQ_DMA_Map1 registers.
3. Program the following fields to initialize the mode of operation in the MTL_TxQ0_Operation_Mode register.
a. Transmit Store And Forward (TSF) or Transmit Threshold Control (TTC) in case of threshold mode
b. Transmit Queue Enable (TXQEN) to value 2'b10 to enable Transmit Queue0
c. Transmit Queue Size (TQS)
4. Program the following fields to initialize the mode of operation in the MTL_RxQ0_Operation_Mode register:
a. Receive Store and Forward (RSF) or RTC in case of Threshold mode
b. Flow Control Activation and De-activation thresholds for MTL Receive FIFO (RFA and RFD)
c. Error Packet and undersized good Packet forwarding enable (FEP and FUP)
d. Receive Queue Size (RQS)
5. Repeat previous two steps for all MTL Tx and Rx queues selected in the configuration.

30.6.10.3 Initializing MAC

The MAC configuration registers establish the operating mode of the MAC. These registers must be initialized before initializing the DMA.
The following MAC Initialization operations can be performed after DMA initialization. If the MAC initialization is completed before the DMA is configured, enable the MAC receiver (last step in the following sequence) only after the DMA is active. Otherwise, received frames fill the Rx FIFO and overflow.
1. Provide the MAC address registers: MAC_Address0_High and MAC_Address0_Low. If more than one MAC address is enabled in your configuration, program the MAC addresses appropriately.

2. Program the following fields to set the appropriate filters for the incoming frames in the MAC_Packet_Filter register:
a. Receive All
b. Promiscuous mode
c. Hash or Perfect Filter
d. Unicast, multicast, broadcast, and control frames filter settings
3. Program the following fields for proper flow control in the MAC_Q0_Tx_Flow_Ctrl register:
a. Pause time and other Pause frame control bits
b. Transmit Flow control bits
c. Flow Control Busy
4. Program the MAC_Interrupt_Enable register, as required, and if applicable, for your configuration.
5. Program the appropriate fields in the MAC_Configuration register. For ex: Inter-packet gap while transmission and jabber disable.
6. Set bit 0 and 1 in MAC_Configuration registers to start the MAC transmitter and receiver.

30.6.10.4 Performing Normal Receive and Transmit Operation

During normal operation of the GMAC, normal and transmit interrupts are read, descriptors polled, the DMA is suspended (if it does not own descriptors), and values of current host transmitter or receiver descriptor pointers are read for debugging.
For normal operation, complete the following steps:
1. For normal transmit and receive interrupts, read the interrupt status. Then, poll the descriptors, reading the status of the descriptor owned by the Host (either transmit or receive).
2. Set appropriate values for the descriptors, ensuring that transmit and receive descriptors are owned by the DMA to resume the transmission and reception of data.
3. If the descriptors are not owned by the DMA (or no descriptor is available), the DMA goes into SUSPEND state. The transmission or reception can be resumed by freeing the descriptors and writing the descriptor tail pointer to Tx/Rx tail pointer register (DMA_CH[n]_TxDesc_Tail_Pointer and DMA_CH[n]_RxDesc_Tail_Pointer).
4. The values of the current host transmitter or receiver descriptor address pointer can be read for the debug process (DMA_CH[n]_Current_App_TxDesc and DMA_CH[n]_Current_App_RxDesc).
5. The values of the current host transmit buffer address pointer and receive buffer address pointer can be read for the debug process (Register DMA_CH[n]_Current_App_TxBuffer and DMA_CH[n]_Current_App_RxBuffer)

30.6.10.5 Stopping and Starting Transmission

Complete the following steps to pause the transmission for some time.
1. Disable the Transmit DMA (if applicable) by clearing Bit 0 (ST) of DMA_CH(#i)_TX_Control (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1) Register.
2. Wait for any previous frame transmissions to complete. You can check this by reading the appropriate bits of MTL_TxQ0_Debug Register (TRCSTS is not 01 and TXQSTS=0).
3. Disable the MAC transmitter and MAC receiver by clearing Bit (RE) and Bit 1(TE) of the MAC_Configuration Register.
4. Disable the Receive DMA (if applicable), after making sure that the data in the Rx FIFO is transferred to the system memory (by reading the appropriate bits of MTL_TxQ0_Debug Register, PRXQ=0 and RXQSTS=00).
5. Make sure that both Tx Queue and Rx Queue are empty (TXQSTS is 0 in MTL_TxQ0_Debug Register and RXQSTS is 0 in MTL_RxQ0_Debug Register).
6. To restart the operation, first start the DMAs, and then enable the MAC Transmitter and Receiver.

30.6.10.6 Initialization Guidelines for System Time Generation

You can enable the timestamp feature by setting Bit 0 of the MAC_Timestamp_Control Register. However, it is essential that the timestamp counter be initialized after this bit is set. Complete the following steps during GMAC initialization:
1. Mask the Timestamp Trigger interrupt by clearing the bit 16 of MAC_Interrupt_Enable Register.
2. Set Bit 0 of MAC_Timestamp_Control Register to enable timestamping.

3. Program MAC_Sub_Second_Increment Register based on the PTP clock frequency.
4. If you are using the Fine Correction approach, program MAC_Timestamp_Addend and set Bit 5 of MAC_Timestamp_Control Register.
5. Poll the MAC_Timestamp_Control Register until Bit 5 is cleared.
6. Program Bit 1 of MAC_Timestamp_Control Register to select the Fine Update method (if required).
7. Program MAC_System_Time_Seconds_Update Register and MAC_System_Time_Nanoseconds_Update Register with the appropriate time value.
8. Set Bit 2 in MAC_Timestamp_Control Register.
The timestamp counter starts operation as soon as it is initialized with the value written in the Timestamp Update registers. If one-step timestamping is enabled
a. To enable one-step timestamping, program Bit 27 of the TDES3 Context Descriptor.
b. Program registers MAC_Timestamp_Ingress_Asym_Corr and MAC_Timestamp_Egress_Asym_Corr to update the correction field in PDelay_Req PTP messages.
9. Enable the MAC receiver and transmitter for proper timestamping.

30.6.10.7 Coarse Correction Method

To synchronize or update the system time in one process (coarse correction method), complete the following steps:
1. Set the offset (positive or negative) in the Timestamp Update registers (MAC_System_Time_Seconds_Update and MAC_System_Time_Nanoseconds_Update).
2. Set Bit 3 (TSUPDT) of MAC_Timestamp_Control Register. The value in the Timestamp Update registers is added to or subtracted from the system time when the TSUPDT bit is cleared.

30.6.10.8 Programming Guidelines for TSO

The TCP Segmentation Offload (TSO) engine is used to offload the TCP segmentation functions to the hardware. To program the TSO, set the TSE bit to enable TCP packet segmentation, and program descriptor fields to enable TSO for the current packet. Complete the following steps to program TSO:
1. Program the TSE bit of corresponding DMA_CH[n]_Tx_Control register to enable TCP packet segmentation in that DMA.
2. In addition to the normal transfer descriptor setting, the following descriptor fields must be programmed to enable TSO for the current packet:
a. Enable TSE in Bit 18 of TDES3
b. Program the length of the un-segmented TCP/IP packet payload in bits [17:0] of TDES3 and the TCP header in bits [22:19] of TDES3.
c. Program the maximum size of the segment in MSS of DMA_CH[n]_Control register or MSS in the context descriptor. If MSS field is programmed in both DMA_CH[n]_Control register and in the context descriptor, the latest software programmed sequence is considered.
3. The header of the unsegmented TCP/IP packet should be in Buffer 1 of the first descriptor and this buffer must not hold any payload bytes. The payload is allocated to Buffer 2 and the buffers of the subsequent descriptors.

30.6.11 Clock Architecture

In RMII mode, reference clock and TX/RX clock can be from CRU or external OSC as following figure. The mux selecting rmii_speed is CRU_GMAC_CON0[1].

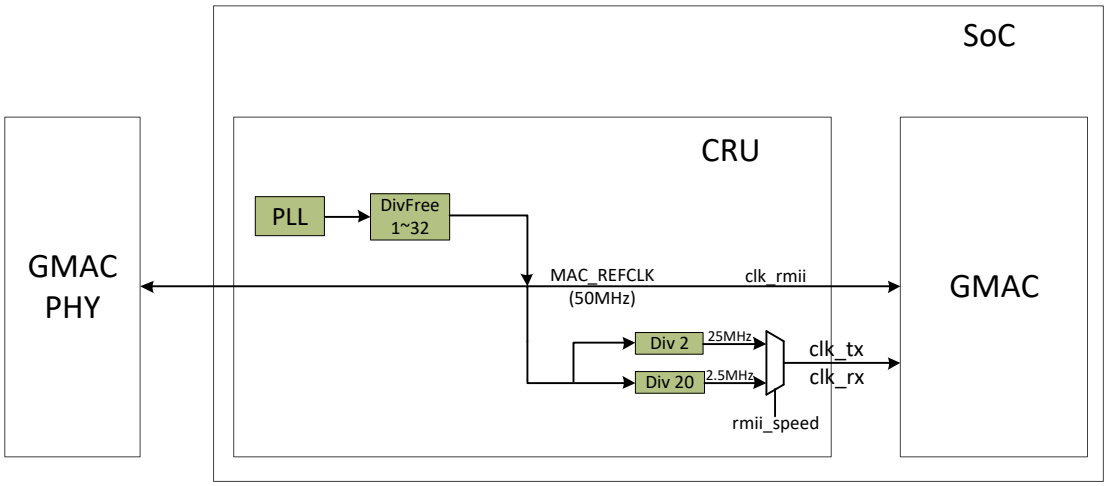


Fig. 30-11 RMII Clock Architecture When Clock Source From CRU

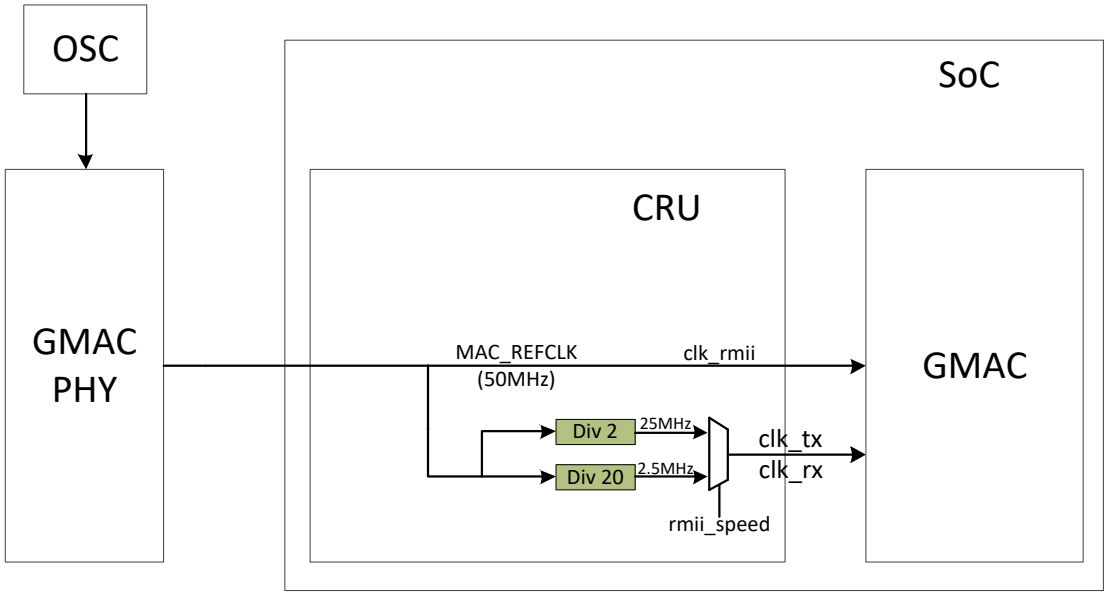


Fig. 30-12 RMII Clock Architecture When Clock Source From External OSC

In RGMII mode, clock architecture only supports that TX clock source is from CRU as following figure.In order to dynamically adjust the timing between TX/RX clocks with data, delayline is integrated in TX and RX clock path. Register GRF_MAC_CON0[3:0] can enable the delayline and GRF_MAC_CON1[15:0]/GRF_MAC_CON2[15:0] is used to determine the delay length. There are 200 delay elements in each delayline.

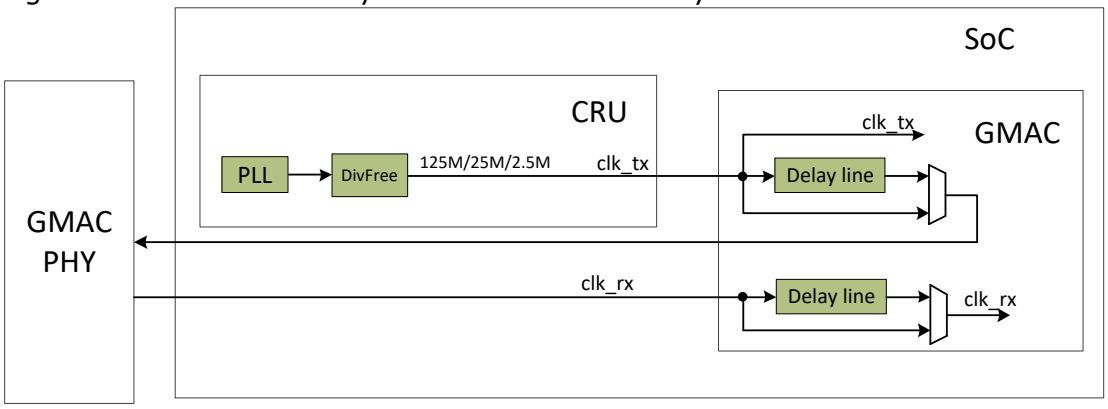


Fig. 30-13 RGMII Clock Architecture When Clock Source From CRU

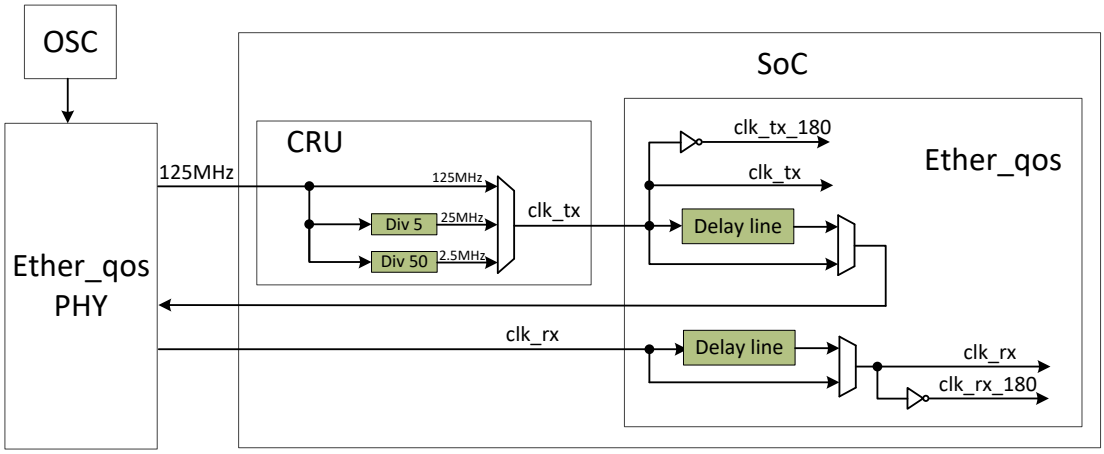


Fig. 30-14 RGMII Clock Architecture When Clock Source From External OSC

Chapter 31 GPIO

31.1 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

31.2 Block Diagram

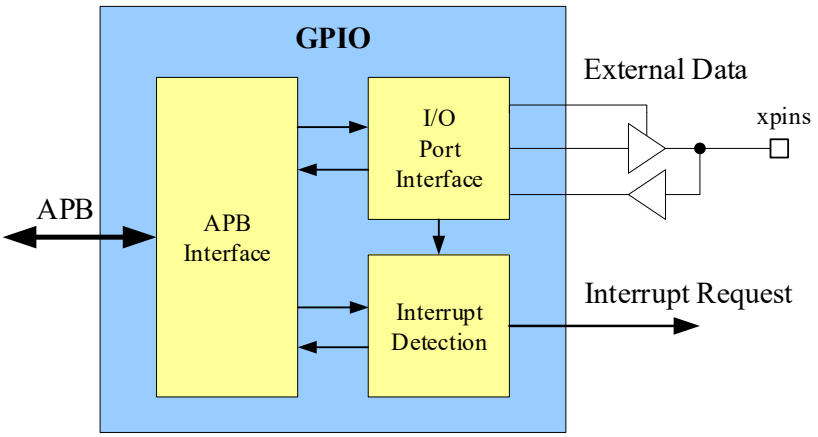


Fig. 31-1 GPIO Block Diagram

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.

31.3 Function Description

31.3.1 Data Control

Under software control, the data and direction control for the signal are sourced from the port data registers (GPIO_SWPORT_DR_L/GPIO_SWPORT_DR_H) and direction control registers (GPIO_SWPORT_DDR_L/GPIO_SWPORT_DRR_H).
The direction of the external I/O pad is controlled by the value of the port data direction registers. The data written to these memory-mapped registers gets mapped onto an output signal (gpio_port_ddr) of the GPIO peripheral. This output signal controls the direction of an external I/O pad. The default data direction is Input.
The data written to the port data registers drives the output buffer (gpio_port_dr) of the I/O pad.
External data are input on the external data signal (gpio_ext_port). Reading the external signal register (GPIO_EXT_PORT) shows the value of this signal, regardless of the direction. This register is read-only, meaning that it cannot be written from the APB software interface.

31.3.2 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.

31.3.3 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).

31.4 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 five GPIOs (GPIO0 in PD_PMU, GPIO1/GPIO2/GPIO3/GPIO4 in PD_BUS), and each of them has same register group. Therefore, five GPIOs' register groups have five different base addresses.

31.4.1 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)

Name	Offset	Size	Reset Value	Description
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:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access **DW**-Double WORD (64 bits) access*

31.4.2 Detail Registers Description

GPIO_SWPORT_DR_L

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	RW	0x0000	swport_dr_low Output data for the lower 16 bits of I/O Port, each bit is individual. 1'b0: Low 1'b1: High Values written to this register are output on the I/O signals for the lower 16 bits of I/O Port if the corresponding data direction bits for I/O Port are set to Output mode. The value read back is equal to the last value written to this register.

GPIO_SWPORT_DR_H

Address: Operational Base + offset (0x0004)

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

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	swport_dr_high Output data for the upper 16 bits of I/O Port, each bit is individual. 1'b0: Low 1'b1: High Values written to this register are output on the I/O signals for the upper 16 bits of I/O Port if the corresponding data direction bits for I/O Port are set to Output mode. The value read back is equal to the last value written to this register.

GPIO_SWPORT_DDR_L

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	RW	0x0000	swport_ddr_low Data direction for the lower 16 bits of I/O Port, each bit is individual. 1'b0: Input 1'b1: Output Values written to this register independently control the direction of the corresponding data bit in the lower 16 bits of I/O Port.

GPIO_SWPORT_DDR_H

Address: Operational Base + offset (0x000C)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	RW	0x0000	swport_ddr_high Data direction for the upper 16 bits of I/O Port, each bit is individual. 1'b0: Input 1'b1: Output Values written to this register independently control the direction of the corresponding data bit in the upper 16 bits of I/O Port.

GPIO_INT_EN_L

Address: Operational Base + offset (0x0010)

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

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	int_en_low Allows each bit of the lower 16 bits of I/O Port to be configured for interrupts. 1'b0: Interrupt is disabled 1'b1: Interrupt is enabled Whenever a 1 is written to a bit of this register, it configures the corresponding bit on I/O Port to become an interrupt source; otherwise, I/O Port operates as a normal GPIO signal. Interrupts are disabled on the corresponding bits of I/O Port if the corresponding data direction register is set to Output.

GPIO_INT_EN_H

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	RW	0x0000	int_en_high Allows each bit of the upper 16 bits of I/O Port to be configured for interrupts. 1'b0: Interrupt is disabled 1'b1: Interrupt is enabled Whenever a 1 is written to a bit of this register, it configures the corresponding bit on I/O Port to become an interrupt source; otherwise, I/O Port operates as a normal GPIO signal. Interrupts are disabled on the corresponding bits of I/O Port if the corresponding data direction register is set to Output.

GPIO_INT_MASK_L

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	RW	0x0000	int_mask_low Controls whether an interrupt on the lower 16 bits of I/O Port can create an interrupt for the interrupt controller by not masking it. 1'b0: Interrupt is unmasked 1'b1: Interrupt is masked Whenever a 1 is written to a bit in this register, it masks the interrupt generation capability for this signal; otherwise interrupts are allowed through.

GPIO_INT_MASK_H

Address: Operational Base + offset (0x001C)

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

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	int_mask_high Controls whether an interrupt on the upper 16 bits of I/O Port can create an interrupt for the interrupt controller by not masking it. 1'b0: Interrupt is unmasked 1'b1: Interrupt is masked Whenever a 1 is written to a bit in this register, it masks the interrupt generation capability for this signal; otherwise interrupts are allowed through.

GPIO_INT_TYPE_L

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	RW	0x0000	int_type_low Controls the type of interrupt that can occur on the lower 16 bits of I/O Port. 1'b0: Level-sensitive 1'b1: Edge-sensitive Whenever a 1 is written to a bit of this register, it configures the interrupt type to be edge-sensitive; otherwise, it is level-sensitive.

GPIO_INT_TYPE_H

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	RW	0x0000	int_type_high Controls the type of interrupt that can occur on the upper 16 bits of I/O Port. 1'b0: Level-sensitive 1'b1: Edge-sensitive Whenever a 1 is written to a bit of this register, it configures the interrupt type to be edge-sensitive; otherwise, it is level-sensitive.

GPIO_INT_POLARITY_L

Address: Operational Base + offset (0x0028)

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

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	int_polarity_low Controls the polarity of edge or level sensitivity that can occur on the lower 16 bits of I/O Port. 1'b0: Active-low 1'b1: Active-high Whenever a 1 is written to a bit of this register, it configures the interrupt type to rising-edge or active-high sensitive; otherwise, it is falling-edge or active-low sensitive.

GPIO_INT_POLARITY_H

Address: Operational Base + offset (0x002C)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	RW	0x0000	int_polarity_high Controls the polarity of edge or level sensitivity that can occur on the upper 16 bits of I/O Port. 1'b0: Active-low 1'b1: Active-high Whenever a 1 is written to a bit of this register, it configures the interrupt type to rising-edge or active-high sensitive; otherwise, it is falling-edge or active-low sensitive.

GPIO_INT_BOTHEDGE_L

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	RW	0x0000	int_bothedge_low Controls the edge type of interrupt that can occur on the lower 16 bits of I/O Port. 1'b0: Disable both-edge detection 1'b1: Enable both-edge detection Whenever a particular bit is programmed to 1, it enables the generation of interrupts on both the rising edge and the falling edge of an external input signal corresponding to that bit on I/O Port. The values programmed in the registers int_type_low and int_polarity_low for this particular bit are not considered when the corresponding bit of this register is set to 1. Whenever a particular bit is programmed to 0, the interrupt type depends on the value of the corresponding bits in the int_type_low and int_polarity_low registers.

GPIO_INT_BOTHEDGE_H

Address: Operational Base + offset (0x0034)

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

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	int_bothedge_high Controls the edge type of interrupt that can occur on the upper 16 bits of I/O Port. 1'b0: Disable both-edge detection 1'b1: Enable both-edge detection Whenever a particular bit is programmed to 1, it enables the generation of interrupts on both the rising edge and the falling edge of an external input signal corresponding to that bit on I/O Port. The values programmed in the registers int_type_high and int_polarity_high for this particular bit are not considered when the corresponding bit of this register is set to 1. Whenever a particular bit is programmed to 0, the interrupt type depends on the value of the corresponding bits in the int_type_high and int_polarity_high registers.

GPIO_DEBOUNCE_L

Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	RW	0x0000	debounce_low Controls whether an external signal of the lower 16 bits of I/O Port that is the source of an interrupt needs to be debounced to remove any spurious glitches. 1'b0: Disable debounce 1'b1: Enable debounce Writing a 1 to a bit in this register enables the debouncing circuitry. A signal must be valid for two periods of an external clock before it is internally processed.

GPIO_DEBOUNCE_H

Address: Operational Base + offset (0x003C)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	RW	0x0000	debounce_high Controls whether an external signal of the lower 16 bits of I/O Port that is the source of an interrupt needs to be debounced to remove any spurious glitches. 1'b0: Disable debounce 1'b1: Enable debounce Writing a 1 to a bit in this register enables the debouncing circuitry. A signal must be valid for two periods of an external clock before it is internally processed.

GPIO_DBCLK_DIV_EN_L

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	RW	0x0000	dbclk_div_en_low Controls whether to use the internal divided clock when debounce function is enabled for an external signal of the lower 16 bits of I/O Port. 1'b0: Disable divider for debounce clock 1'b1: Enable divider for debounce clock Whenever a 1 is written to a bit of this register, the clock divided from dbclk is used as debounce clock; otherwise, the original dbclk is used. The clock divide factor depends on the register dbclk_div_con. The values programmed in this register for this particular bit are not considered when the corresponding bit of the register debounce_low is set to 0.

GPIO_DBCLK_DIV_EN_H

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	RW	0x0000	dbclk_div_en_high Controls whether to use the internal divided clock when debounce function is enabled for an external signal of the upper 16 bits of I/O Port. 1'b0: Disable divider for debounce clock 1'b1: Enable divider for debounce clock Whenever a 1 is written to a bit of this register, the clock divided from dbclk is used as debounce clock; otherwise, the original dbclk is used. The clock divide factor depends on the register dbclk_div_con. The values programmed in this register for this particular bit are not considered when the corresponding bit of the register debounce_high is set to 0.

GPIO_DBCLK_DIV_CON

Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31:24	RO	0x00	reserved
23:0	RW	0x000001	dbclk_div_con dbclk_div = dbclk / (dbclk_div_con + 1)

GPIO_INT_STATUS

Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	int_status Interrupt status of I/O Port.

GPIO_INT_RAWSTATUS

Address: Operational Base + offset (0x0058)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	int_rawstatus Interrupt raw status of I/O Port (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	R/W SC	0x0000	port_eoi_low Controls the clearing of edge type interrupts from the lower 16 bits of I/O Port. 1'b0: Nothing 1'b1: Clear edge-sensitive interrupt When a 1 is written into a corresponding bit of this register, the interrupt is cleared and the bit is self cleared at once. Writing to this register has no effect on level-sensitive interrupts. All interrupts are cleared when I/O Port is not configured for interrupts.

GPIO_PORT_EOI_H

Address: Operational Base + offset (0x0064)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	R/W SC	0x0000	port_eoi_high Controls the clearing of edge type interrupts from the upper 16 bits of I/O Port. 1'b0: Nothing 1'b1: Clear edge-sensitive interrupt When a 1 is written into a corresponding bit of this register, the interrupt is cleared and the bit is self cleared at once. Writing to this register has no effect on level-sensitive interrupts. All interrupts are cleared when I/O Port is not configured for interrupts.

GPIO_EXT_PORT

Address: Operational Base + offset (0x0070)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	ext_port This register always reflects the value of the signals on the external I/O Port.

GPIO_VER_ID

Address: Operational Base + offset (0x0078)

Bit	Attr	Reset Value	Description
31:0	RO	0x01000c2b	ver_id Version ID.

31.5 Interface Description

Table 31-1 GPIO Interface Description

Module Pin	Direction	Pad Name	IOMUX Setting
GPIO0 Interface			
gpio0_port[3:0]	I/O	GPIO0_A[3:0]	PMUGRF_GPIO0A_IOMUX_L[15:0]=16'h0
gpio0_port[7:4]	I/O	GPIO0_A[7:4]	PMUGRF_GPIO0A_IOMUX_H[15:0]=16'h0

Module Pin	Direction	Pad Name	IOMUX Setting
gpio0_port[11:8]	I/O	GPIO0_B[3:0]	PMUGRF_GPIO0B_IOMUX_L[15:0]=16'h0
gpio0_port[15:12]	I/O	GPIO0_B[7:4]	PMUGRF_GPIO0B_IOMUX_H[15:0]=16'h0
gpio0_port[19:16]	I/O	GPIO0_C[3:0]	PMUGRF_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[31:28]	I/O	GPIO0_D[7:4]	GRF_GPIO0D_IOMUX_H[15:0]=16'h0
GPIO1 Interface			
gpio1_port[3:0]	I/O	GPIO1_A[3:0]	GRF_GPIO1A_IOMUX_L[15:0]=16'h0
gpio1_port[7:4]	I/O	GPIO1_A[7:4]	GRF_GPIO1A_IOMUX_H[15:0]=16'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[23:20]	I/O	GPIO1_C[7:4]	GRF_GPIO1C_IOMUX_H[15:0]=16'h0
gpio1_port[27:24]	I/O	GPIO1_D[3:0]	GRF_GPIO1D_IOMUX_L[15:0]=16'h0
gpio1_port[31:28]	I/O	GPIO1_D[7:4]	GRF_GPIO1D_IOMUX_H[15:0]=16'h0
GPIO2 Interface			
gpio2_port[3:0]	I/O	GPIO2_A[3:0]	GRF_GPIO2A_IOMUX_L[15:0]=16'h0
gpio2_port[7:4]	I/O	GPIO2_A[7:4]	GRF_GPIO2A_IOMUX_H[15:0]=16'h0
gpio2_port[11:8]	I/O	GPIO2_B[3:0]	GRF_GPIO2B_IOMUX_L[15:0]=16'h0
gpio2_port[15:12]	I/O	GPIO2_B[7:4]	GRF_GPIO2B_IOMUX_H[15:0]=16'h0
gpio2_port[19:16]	I/O	GPIO2_C[3:0]	GRF_GPIO2C_IOMUX_L[15:0]=16'h0
gpio2_port[23:20]	I/O	GPIO2_C[7:4]	GRF_GPIO2C_IOMUX_H[15:0]=16'h0
gpio2_port[27:24]	I/O	GPIO2_D[3:0]	GRF_GPIO2D_IOMUX_L[15:0]=16'h0
gpio2_port[31:28]	I/O	GPIO2_D[7:4]	GRF_GPIO2D_IOMUX_H[15:0]=16'h0
GPIO3 Interface			
gpio3_port[3:0]	I/O	GPIO3_A[3:0]	GRF_GPIO3A_IOMUX_L[15:0]=16'h0
gpio3_port[7:4]	I/O	GPIO3_A[7:4]	GRF_GPIO3A_IOMUX_H[15:0]=16'h0
gpio3_port[11:8]	I/O	GPIO3_B[3:0]	GRF_GPIO3B_IOMUX_L[15:0]=16'h0
gpio3_port[15:12]	I/O	GPIO3_B[7:4]	GRF_GPIO3B_IOMUX_H[15:0]=16'h0
gpio3_port[19:16]	I/O	GPIO3_C[3:0]	GRF_GPIO3C_IOMUX_L[15:0]=16'h0
gpio3_port[23:20]	I/O	GPIO3_C[7:4]	GRF_GPIO3C_IOMUX_H[15:0]=16'h0
gpio3_port[27:24]	I/O	GPIO3_D[3:0]	GRF_GPIO3D_IOMUX_L[15:0]=16'h0
gpio3_port[31:28]	I/O	GPIO3_D[7:4]	GRF_GPIO3D_IOMUX_H[15:0]=16'h0
GPIO4 Interface			
gpio4_port[1:0]	I/O	GPIO4_A[1:0]	GRF_GPIO4A_IOMUX_L[7:0]=8'h0

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

31.6 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.