SOPHON

SG2300X Datasheet

Revision: v1.3

Date: 08/10/2022

Revision History

Revision	Issue Date	Author	Comments
0.1	2022/04/12	Jun Wang	Initial Draft
0.2	2022/07/07	Jun Wang	Update package information
0.3	2022/07/13	Jun Wang	Update eFuse Bit Map description
0.33	2022/07/18	Xin Zhang	Complete Appendix
1.0	2022/07/19	Jun Wang	Initial Release
1.1	2022/08/08	Peng Gao	Update overview
1.2	2022/08/09	Peng Gao	Update TPU descriptions Add key feature sections Remove process information
1.3	2022/08/10	Peng Gao	Update power descriptions

Table of Contents

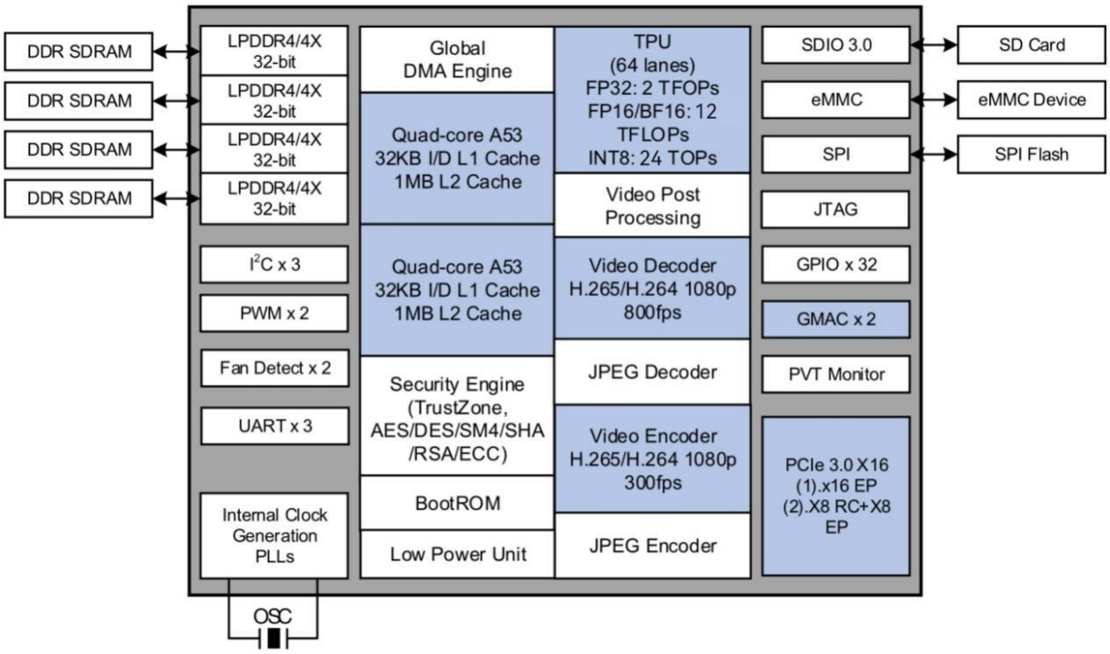
1 Overview

SG2300X is the fourth-generation tensor processor unit launched by SOPHON for deep learning inference application. It continues to provide more abundant, economic and adaptive AI computing power for all scenarios. All popular CNN / RNN / DNN network topologies can be highly accelerated when executing on SG2300X.

SG2300X can provide 24TOPs peak performance for 8-bit integer convolution/GEMM operation, 12TFLOPs peak performance for half precision floating point or bfloat16 operation, 2TFLOPs peak performance for single precision floating point operation. 16MB on chip memory resource enables good data reusability for performance optimization.

We also provide powerful software SDK. All the applications developed based on the mainstream deep learning frameworks (such as ONNX / Pytorch / Caffe / Tensorflow) could be ported to SG2300X platform easily.

The functional block diagram of SG2300X is shown below.


DDR SDRAM
DDR SDRAM
DDR SDRAM
DDR SDRAM
LPDDR4/4X 32-bit
LPDDR4/4X 32-bit
LPDDR4/4X 32-bit
LPDDR4/4X 32-bit
I^2C x 3
PWM x 2
Fan Detect x 2
UART x 3
Internal Clock Generation PLLs
OSC
Global DMA Engine
Quad-core A53 32KB I/D L1 Cache 1MB L2 Cache
Quad-core A53 32KB I/D L1 Cache 1MB L2 Cache
Security Engine (TrustZone, AES/DES/SM4/SHA /RSA/ECC)
BootROM
Low Power Unit
TPU (64 lanes) FP32: 2 TFOPs FP16/BF16: 12 TFLOPs INT8: 24 TOPs
Video Post Processing
Video Decoder H.265/H.264 1080p 800fps
JPEG Decoder
Video Encoder H.265/H.264 1080p 300fps
JPEG Encoder
SDIO 3.0
eMMC
SPI
JTAG
GPIO x 32
GMAC x 2
PVT Monitor
PCIe 3.0 X16 (1).x16 EP (2).X8 RC+X8 EP
SD Card
eMMC Device
SPI Flash

1

Figure 1 SG2300X Overview

1.1 Key Features

- **TPU Maximum Performance**
  - 24TOPs int8 for convolution/GEMM
  - 12TFLOPs fp16/bf16 for convolution/GEMM
  - 2TFLOPS fp32 for convolution/GEMM
  - 2454 fps Resnet50 network inference with 4 batch
  - 864 fps BERT-base network inference with 4 batch
  - 65 fps BERT-large network inference with 4 batch
- **Multimedia Codec Typical Performance**
  - H.265/H.264 decoder; decoder performance 1080p 800fps
  - H.265/H.264 encoder; encoder performance 1080p 300fps
- **Maximum Power when running 32 channel 1080P 25fps H.265/H.264 video structuring analysis:**
  - 12.8W @Junction Temperature 40°C
  - 16.6W @Junction Temperature 60°C
  - 20.2W @Junction Temperature 85°C

1.2 Detailed Features

- **Application Processor Subsystem**
  - 2 CPU clusters
  - 2 CPU clusters
  - Each cluster includes RISC Quad Core
  - Frequency 2.3GHz
  - 32KB L1 I Cache and D Cache
  - With 1MB L2 Cache
  - Core-sight support
  - Boot method: SPI Flash / 256KB ROM/EMMC/SD/I2C/UART
- **TPU Subsystem**
  - 64 lanes in a single chip
  - 256KB local memory for each lane
  - Max Frequency is 1000MHz
  - Peak Performance (TPU running at 1000MHz):
    - 24TOPs int8 convolution/GEMM
    - 12TFLOPs fp16/bf16 convolution/GEMM

        - 2TFLOPS fp32 convolution/GEMM
        - 8TOPS int8 vector operation
        - 4TFLOPS fp16/bf16 vector operation
        - 2TFLOPS fp32 vector operation

- **Memory Interface**
    - LPDDR4/LPDDR4x support
    - Highest data rate: 4266Mbps
    - Memory data bus width Per channel: 32bit
    - Four independent DDR channels
    - Two-rank supported
    - Maximum memory capacity up to 16GB
- **Video Subsystem**
    - H.265/H.264 decoder; decoder performance 1080p 800fps
    - H.265/H.264 encoder; encoder performance 1080p 300fps
    - Post-processing Unit for Color space conversion / Cropping / Scaling
    - JPEG Decoder performance 1080p 600fps
    - JPEG Encoder performance 1080p 600fps
- **PCIe**
    - Gen3 X16 EP support, Max speed 128Gbps or Gen3 X8 RC+EP, each max speed 64Gbps
    - Configurable Max Payload Size
- **Ethernet**
    - 2 Ethernet interfaces
    - 10/100/1000Mbps
    - RGMII support
    - IEEE 802.3u Media Access Controller
- **SDIO Controller**
    - SDIO 3.0 support
    - 4-bit data interface for SD cards, WIFI, Bluetooth
    - Supports card interrupts for SD cards and SDIO devices
- **eMMC**
    - 4.5/5.1 flash storage
    - 4-bit bus width
    - 32GB storage capacity
    - Maximum bus frequency 200MHz
- **SPI Flash Controller**
    - Synchronous, serial, full duplex transfer

- SPI mode 0,1,2,3 supported
  - Configurable frame length 2 ~ 16 bits
  - LSB or MSB First Tx/Rx
  - Dual/Quad I/O read/write Flash supported
  - Direct Memory Mapping Read supported
  - Boot from SPI Flash support
  - DMA transfer supported
- **UART**
  - Three UART interfaces
  - FIFO mode and register mode
  - Fractional Baud rate generator; supports a wide range of target baud rates from 2400 to 230.4K
  - Independent RX and TX DMA requests
  - DMA mode support
- **I2C**
  - Three I2C interfaces
  - Two-wire I2C serial interface
  - Three speeds:
    - Standard mode (0 to 100 Kb/s)
    - Fast mode (≤ 400 Kb/s)
    - High-speed mode (≤ 3.4 Mb/s)
  - Master or slave I2C operation
  - 7- or 10-bit addressing
  - Bulk transmit mode supported
  - Handles Bit and Byte waiting at all bus speeds
- **Other IOs**
  - Two PWM interfaces and two fan speed detection interfaces
  - 32 GPIOs
- **Security**
  - Security scheme for AI model and user data support
  - AES/DES/SM4/SHA/RSA/ECC accelerator
  - True random number generator support
  - Secure key storage scheme support
  - Secure boot support
  - Trust-zone support
- **Hardware algorithm accelerator**
  - Classify and statistics accelerate support

  - Sorting algorithm accelerate support
  - Data gather accelerate support
  - Base64 encoder/decoder accelerate support

- **Electrical**
  - Digital Core Voltage: 0.80V +/- 10%, 0.65V +/- 10%
  - I/O Voltage: 3.3/1.8V +/-5%
  - Less than 15W in typical operation
  - Operating Temperature: -40C min, 105C max

2 Function Specification

2.1 Boot Mode and Chip Mode

SG2300X boot sequence has two stages.

First stage, A53 fetches the first instruction from off-chip SPI NOR Flash or on-chip ROM.

- BOOT_SEL[3] = 1’b0: boot from internal ROM. A53 fetch the first instruction from internal ROM. The boot loader code is embedded in the ROM. After ROM boot, corresponding device will be ready for Linux kernel loading.
- BOOT_SEL[3] = 1’b1: boot from SPI NOR FLASH. This is mainly for debug purpose.

A53 fetches the first instruction from the off-chip SPI NOR Flash with address 0x0600_0000. The main code will be later loaded to AXI SRAM or DDR. Flash is read only, write to Flash will lead to unexpected error. After Flash boot, corresponding device will be ready for Linux kernel loading.

Second stage, A53 will load Linux kernel.

- BOOT_SEL[2:0] = 3’b000: loading next bootloader from SPI NOR Flash, but try SD boot first (note1)
- BOOT_SEL[2:0] = 3’b001: loading next bootloader from eMMC's boot0 partition, but try SD boot first (note1)
- BOOT_SEL[2:0] = 3’b010: loading next bootloader from AXI SRAM, loaded by dbg_I2C/PCIe/JTAG, but try SD boot first (note1)
- BOOT_SEL[2:0] = 3’b011: this is not used.
- BOOT_SEL[2:0] = 3’b100: loading next bootloader from SPI NOR Flash, without trying SD boot first
- BOOT_SEL[2:0] = 3’b101: loading next bootloader from eMMC's boot0 partition, without trying SD boot first
- BOOT_SEL[2:0] = 3’b110: loading next bootloader from AXI SRAM, loaded by dbg_I2C/PCIe/JTAG, without trying SD boot first
- BOOT_SEL[2:0] = 3’b111: A53 will enter WFI directly, do nothing else.

Note

1. If BOOT_SEL[2] = 1'b0, the software will check whether there is SD card inserted and whether there is firmware image in the SD card first. If true, it will use this firmware in SD card to boot up, otherwise it will check BOOT_SEL[1:0] for firmware source.

2. When booting from SPI Flash, chip IO SPIF_CLK_SEL1 is used to determine clock frequency of SPI interface as shown in this table.

Table 1 SPI Flash Clock Frequency

SPIF_CLK_SE	SPI Flash Clock

L1	Frequency
0	10MHz
1	25MHz

SG2300X chip mode can be selected by set specific IO into different constant value as shown in this table.

Table 2 SG2300X Chip Mode

Mode	TEST_EN	BOOT_SEL1	BOOT_SEL0	MODE_SEL2	MODE_SEL1	MODE_SEL0	Note
PTEST mode	1	0	0	1	X	X	X means the value can be determined by test requirement
DDR PHY test mode	1	0	1	1	X	X	
Efuse ext mode	1	1	0	1	X	X	
Latch up mode	1	1	1	1	X	X	
Normal Speed Function Mode	0	X	X	X	0	0	
Fast Speed Function Mode	0	X	X	X	0	1	
Safe Speed Function Mode	0	X	X	X	1	0	
Bypass Speed Function Mode	0	X	x	x	1	1	

IO MODE_SEL2 is used to control whether or not enable fast reset mode in function mode.

- 1'b0: disable fast reset mode
- 1'b1: enable fast reset mode

In fast reset mode, SG2300X will bypass reset de-bounce stage. And disable fast reset mode is only used for debug.

2.2 Memory Map

Table 3 SG2300X Memory Map

Start Address	End Address	On-Chip	Off-Chip	Memory Size (Byte)
0:0000:0000	0:01FF:FFFF	RESERVED		32M
0:0200:0000	0:0200:7FFF	AP Lite0 GIC		32K
0:0200:8000	0:0200:8FFF	HAU GDE REG		4K
0:0200:9000	0:0200:9FFF	HAU SORT REG		4K
0:0200:A000	0:0200:AFFF	HAU NMS REG		4K
0:0200:B000	0:0200:BFFF	TSDMA REG		4K

0:0200:C000	0:05FF:FFFF	RESERVED		63.953125M
0:0600:0000	0:06FF:FFFF		Serial Flash	16M
0:0700:0000	0:070F:FFFF	ROM		1M
0:0710:0000	0:07FF:FFFF	RESERVED		15M
0:0800:0000	0:08FF:FFFF	TPU Local Memory		16M
0:0900:0000	0:0900:FFFF	TPU Static Memory		64K
0:0A00:0000	0:0FFF:FFFF	RESERVED		96M
0:1000:0000	0:101F:FFFF	AXI SRAM		2M
0:1020:0000	0:11FF:FFFF	RESERVED		30M
0:1200:0000	0:1200:7FFF	PKA		32K
0:1200:8000	0:1200:8FFF	SPACC		4K
0:1204:0000	0:4FFF:FFFF	RESERVED		991.75M
0:5000:0000	0:5000:7FFF	GIC		32K
0:5000:8000	0:5000:83FF	AXI Fab24 Performance Monitor		1K
0:5000:8400	0:5000:87FF	AXI Fab25 Performance Monitor		1K
0:5000:8800	0:5000:8BFF	Video system 0 AXI Fab5 Performance Monitor		1K
0:5000:8C00	0:5000:8FFF	Video system 1 AXI Fab5 Performance Monitor		1K
0:5000:9000	0:5000:DFFF	RESERVED		20K
0:5000:E000	0:5000:E3FF	TOP Security Regs		1K
0:5000:E400	0:5000:E7FF	PCIe Security Regs		1K
0:5000:E800	0:5000:EBFF	TPU Security Regs		1K
0:5000:EC00	0:5000:EFFF	Video subsystem 0 Security Regs		1K
0:5000:F000	0:5000:F3FF	Video subsystem 1 Security Regs		1K
0:5000:F400	0:5000:FFFF	RESERVED		3K
0:5001:0000	0:5001:7FFF	TOP Registers		32K
0:5001:8000	0:5001:83FF	TRNG		1K
0:5001:8400	0:5001:9FFF	RESERVED		7K
0:5001:A000	0:5001:BFFF	I2C 0		8K
0:5001:C000	0:5001:DFFF	I2C 1		8K
0:5001:E000	0:5001:FFFF	I2C 2		8K
0:5002:0000	0:5002:1FFF	RESERVED		8K
0:5002:2000	0:5002:2FFF	OS Timer		4K
0:5002:3000	0:5002:3FFF	Interrupt Controller0		4K
0:5002:4000	0:5002:5FFF	RESERVED		8K
0:5002:6000	0:5002:6FFF	Watch Dog		4K
0:5002:7000	0:5002:73FF	GPIO0		1K
0:5002:7400	0:5002:77FF	GPIO1		1K
0:5002:7800	0:5002:7BFF	GPIO2		1K
0:5002:7C00	0:5002:7FFF	RESERVED		1K
0:5002:8000	0:5002:8FFF	Efuse Controller		4K
0:5002:9000	0:5002:93FF	PWM0		1K
0:5002:9400	0:5002:AFFF	RESERVED		7K
0:5002:B000	0:5002:BFFF	Interrupt Controller1		4K
0:5002:C000	0:5002:CFFF	Interrupt Controller2		4K
0:5002:D000	0:5002:DFFF	Interrupt Controller3		4K
0:5002:E000	0:5002:FFFF	RESERVED		8K
0:5003:0000	0:5003:FFFF	Video subsystem 0 JPEG Codec 0		64K

0:5004:0000	0:5004:FFFF	RESERVED		64K
0:5005:0000	0:5005:FFFF	Video subsystem 0 Video Decoder Wave 0		64K
0:5006:0000	0:5006:FFFF	RESERVED		64K
0:5007:0000	0:5007:0FFF	Video subsystem 0 VPP		4K
0:5007:1000	0:500A:FFFF	RESERVED		252K
0:500B:0000	0:500B:FFFF	Video subsystem 1 JPEG Codec 0		64K
0:500C:0000	0:500C:FFFF	RESERVED		64K
0:500D:0000	0:500D:FFFF	Video subsystem 1 Video Decoder Wave 0		64K
0:500E:0000	0:500E:FFFF	RESERVED		64K
0:500F:0000	0:500F:0FFF	Video subsystem 1 VPP		4K
0:500F:1000	0:500F:FFFF	RESERVED		60K
0:5010:0000	0:5010:0FFF	SD Controller 0 (eMMC)		4K
0:5010:1000	0:5010:1FFF	SD Controller 1 (SDIO)		4K
0:5010:2000	0:5010:7FFF	RESERVED		24K
0:5010:8000	0:5010:BFFF	Gigabit Ethernet 0		16K
0:5010:C000	0:5010:FFFF	Gigabit Ethernet 1		16K
0:5011:0000	0:5011:0FFF	System DMA		4K
0:5011:1000	0:5011:1FFF	Video system 0 JPEG0 ATT		4K
0:5011:2000	0:5011:2FFF	Video system 0 DEC0 ATT		4K
0:5011:3000	0:5011:3FFF	Video system 0 VPP ATT		4K
0:5011:4000	0:5011:7FFF	RESERVED		16K
0:5011:8000	0:5011:9FFF	UART0		8K
0:5011:A000	0:5011:BFFF	UART1		8K
0:5011:C000	0:5011:DFFF	UART2		8K
0:5011:E000	0:5012:0FFF	RESERVED		12K
0:5012:1000	0:5012:1FFF	Video system 1 JPEG0 ATT		4K
0:5012:2000	0:5012:2FFF	Video system 1 DEC0 ATT		4K
0:5012:3000	0:5012:3FFF	Video system 1 VPP ATT		4K
0:5012:4000	0:5012:5FFF	RESERVED		8K
0:5012:6000	0:5013:5FFF	Video Encoder Wave521		64K
0:5013:6000	0:5013:6FFF	Video Encoder ATT		4K
0:5013:7000	0:501F:FFFF	RESERVED		804K
0:5020:0000	0:502F:FFFF	CCI		1M
0:5030:0000	0:57FF:FFFF	RESERVED		125M
0:5800:0000	0:5800:0FFF	Global DMA		4K
0:5800:1000	0:5800:1FFF	TPU Register		4K
0:5800:2000	0:5800:2FFF	MMU		4K
0:5800:3000	0:5800:3FFF	Chip Link DMA		4K
0:5800:5000	0:5F5F:FFFF	RESERVED		117.98046875M
0:5F60:0000	0:5FFF:FFFF	PCIE config		10M
0:6000:0000	0:67FF:FFFF	RESERVED		128M
0:6800:0000	0:69FF:FFFF	DDR2_A register		32M
0:6A00:0000	0:6BFF:FFFF	DDR0_A register		32M
0:6C00:0000	0:6DFF:FFFF	DDR0_B register		32M
0:6E00:0000	0:6FFF:FFFF	DDR1 register		32M
0:7000:0000	0:77FF:FFFF	RESERVED		128M
0:7800:0000	0:7800:0FFF	Mirror Global DMA		4K
0:7800:1000	0:7800:1FFF	Mirror TPU Register		4K

0:7800:2000	0:7800:2FFF	Mirror MMU		4K
0:7800:3000	0:7800:3FFF	Mirror Chip Link DMA		4K
0:7800:4000	0:7FFF:FFFF	RESERVED	RESERVED	127.984375M
0:8000:0000	0:FFFF:FFFF	Remote PCIE config		2G
1:0000:0000	1:FFFF:FFFF		DDR0A Memory Range	4G
2:0000:0000	2:FFFF:FFFF		DDR0B Memory Range	4G
3:0000:0000	3:FFFF:FFFF		DDR1 Memory Range	4G
4:0000:0000	4:FFFF:FFFF		DDR2 Memory Range	4G

2.3 DDR Interleaved Access

1. 4K Interleave Access Overview

SG2300X supports interleaved access by spreading memory transfers across different memory controllers.

The granularity of interleave is 4K. For example, if the interleave group is made up of two memory controllers, odd 4K access will be routed to one controller and even 4K access will be allocated to the other.

2. Interleave Control

(1). Interleave Enable/Disable

The interleave access on DDR is enabled by default. SW is able to disable Interleave Access by clearing bit[3] of Top Control Register (0x50010008).

(2). Interleave Mode

SG2300X supports two Interleave Modes:

[1]. Twin Mode:

This is the default Interleave Mode. The interleave operation is only conducted within DDR0_A and DDR0_B while DDR1 and DDR2 are NOT involved.

[2]. Quadruplets Mode:

The interleave group is made up of all four memory controllers (DDR0_A, DDR0_B, DDR_1, DDR_2). Note that, in this mode, Video will NOT access DDR. The Interleave Mode is selected by programming bit[4] of Top Control Register (0x50010008).

The following figure gives user an illustration on how TOP fabric interconnect spreads the memory transfers on different DDR controllers under certain interleave control.

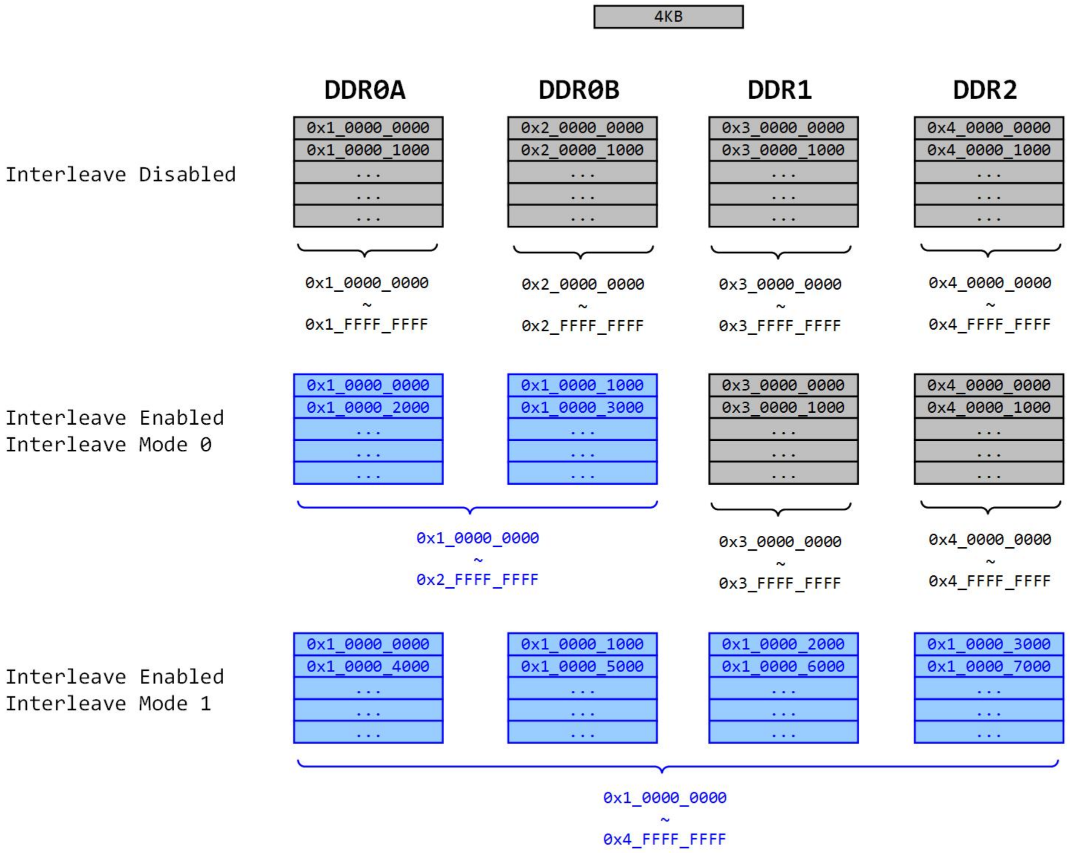


Figure 2 DDR Interleave

2.4 Interrupt

SG2300X Interrupt Table is shown below.

Table 4 SG2300X Interrupt

Interrupt Description	A53 Interrupt ID	AP Lite IRQ ID	AP Lite FIQ ID	PCIe Interrupt ID
SYSDMA Interrupt	32	0		0
UART0 Interrupt	33	1		1
UART0 DMA Tx Interrupt	34	2		2
UART0 DMA Rx Interrupt	35	3		3
UART1 Interrupt	36	4		4
UART1 DMA Tx Interrupt	37	5		5
UART1 DMA Rx Interrupt	38	6		6
UART2 Interrupt	39	7		7
UART2 DMA Tx Interrupt	40	8		8
UART2 DMA Rx Interrupt	41	9		9
UART CHN Tx Interrupt	42	10		10
ETH0 Interrupt	43	11		11

ETH1 Interrupt	44	12		12
SD Wakeup Interrupt	45	13		13
SD Interrupt	46	14		14
EMMC Wakeup Interrupt	47	15		15
EMMC Interrupt	48	16		16
PKA Interrupt	49	17		17
SPACC Interrupt	50	18		18
SF Interrupt	51	19		19
GPIO0 Interrupt	52	20		20
GPIO1 Interrupt	53	21		21
GPIO2 Interrupt	54	22		22
WDT Interrupt	55	23	0	23
TIMER Interrupt	56	24	1	24
I2C0 Interrupt	57	25		25
I2C1 Interrupt	58	26		26
I2C2 Interrupt	59	27		27
VD0 JPEG0 Interrupt	60	28		28
VD0 JPEG0 ATT Interrupt	61	29		29
VD0 VPP Interrupt	62	30	2	30
VD0 Wave0 Interrupt	63	31		31
Previous: vsys_0_wave1_intr	64	32		32
Previous: vsys_0_vmmu_intr	65	33		33
TSDMA Interrupt	66	34		34
HAU NMS Interrupt	67	35		35
HAU SORT Interrupt	68	36		36
TPU PIO Empty Interrupt	69	37		37
TPU PIO Half Empty Interrupt	70	38		38
TPU PIO Quart Empty Interrupt	71	39		39
TPU PIO One Empty Interrupt	72	40		40
HAU GDE Interrupt	73	41		41
GDMA Interrupt	74	42	3	42
TPU Interrupt	75	43	4	43
PCIe Interrupt	76	44		44
SMMU Interrupt	77	45		45
CDMA Interrupt	78	46	5	46
DDR2 PHY Interrupt	79	47		47
General Purpose Interrupt 0	80			
General Purpose Interrupt 1	81			
General Purpose Interrupt 2	82			
General Purpose Interrupt 3	83			
General Purpose Interrupt 4	84			
General Purpose Interrupt 5	85			
General Purpose Interrupt 6	86			
General Purpose Interrupt 7	87			
General Purpose Interrupt 8		48		48
General Purpose Interrupt 9		49		49

General Purpose Interrupt 10		50		50
General Purpose Interrupt 11		51		51
General Purpose Interrupt 12		52		52
General Purpose Interrupt 13		53		53
General Purpose Interrupt 14		54		54
General Purpose Interrupt 15		55		55
DDR0A Interrupt	88	56		56
DDR0B Interrupt	89	57		57
DDR1 Interrupt	90	58		58
DDR2 Interrupt	91	59		59
DDR0A PHY Interrupt	92	60		60
DDR0B PHY Interrupt	93	61		61
DDR1 PHY Interrupt	94	62		62
Reserved	95	NA		NA
VE Wave Interrupt	96	63[0]		63[0]
VE Wave ATT Interrupt	97	63[1]		63[1]
TOP Security Interrupt	140	63[32]		63[32]
VD0 VIDEO Security Interrupt	141	63[33]		63[33]
TPU SE Security Interrupt	142	63[34]		63[34]
PCIe SE Security Interrupt	143	63[35]		63[35]
VD1 VIDEO Security Interrupt	144	63[36]		63[36]
DDR0A Done Interrupt	145	63[37]		63[37]
DDR0B Done Interrupt	146	63[38]		63[38]
DDR1 Done Interrupt	147	63[39]		63[39]
General Purpose Interrupt 16	148			
General Purpose Interrupt 17	149			
General Purpose Interrupt 18	150			
General Purpose Interrupt 19	151			
General Purpose Interrupt 20	152			
General Purpose Interrupt 21	153			
General Purpose Interrupt 22	154			
General Purpose Interrupt 23	155			
General Purpose Interrupt 24		63[40]		63[40]
General Purpose Interrupt 25		63[41]		63[41]
General Purpose Interrupt 26		63[42]		63[42]
General Purpose Interrupt 27		63[43]		63[43]
General Purpose Interrupt 28		63[44]		63[44]
General Purpose Interrupt 29		63[45]		63[45]
General Purpose Interrupt 30		63[46]	6	63[46]
General Purpose Interrupt 31		63[47]	7	63[47]
VD1 JPEG0 Interrupt	156	63[48]		63[48]
VD1 JPEG0 ATT Interrupt	157	63[49]		63[49]
VD1 VPP Interrupt	158	63[50]		63[50]
VD1 Wave0 Interrupt	159	63[51]		63[51]
VD1 Wave0 ATT Interrupt	160	63[52]		63[52]
VD1 VPP ATT Interrupt	161	63[53]		63[53]
Reserved	162	63[54]		63[54]

Reserved	163	63[55]		63[55]
Reserved	164	63[56]		63[56]
Reserved	165	63[57]		63[57]
Reserved	166	63[58]		63[58]
Reserved	167	63[59]		63[59]
Reserved	168	63[60]		63[60]
DDR2 Interrupt	169	63[61]		63[61]
TRNG Interrupt	170	63[62]		63[62]

2.5 Clock

2.5.1 Board Level Clocking

2.5.1.1 Board Level Clocking Summary

Three 25MHz crystal oscillators are required to provide clock source for SG2300X internal PLLs.

As a backup option, three 25MHz clock gen chips are suggested to be deployed on test board. They should be connected to PLL_CLK_IN, PLL_CLK_IN_DPLL0 and PLL_CLK_IN_DPLL1.

For ethernet, gigabit Ethernet PHY provides RX clock to the gigabit Ethernet MAC inside SG2300X chip.

For PCIE, PCIe PHY needs 2 100MHz reference clock inputs. One is for PCIe EP, the other is for PCIe RC.

The following figure provides the illustration of SG2300X board level clocking scheme.

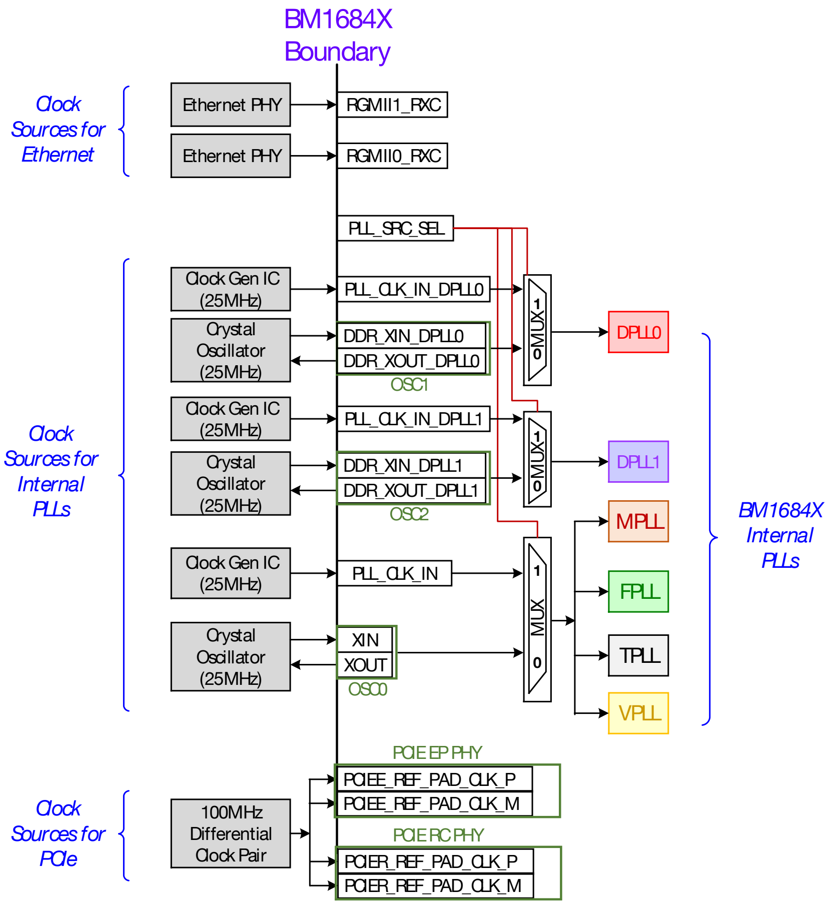


Figure 3 SG2300X board level clocking scheme

2.5.2 Chip Clock Scheme

2.5.2.1 PLL Reference Clock

As shown in the following figure, the IO named PLL_SRC_SEL is used to select the source for PLL's reference clock.

Besides this IO, SW is also capable of programming Memory-mapped registers for controlling the DPLL0/1's reference clock source.

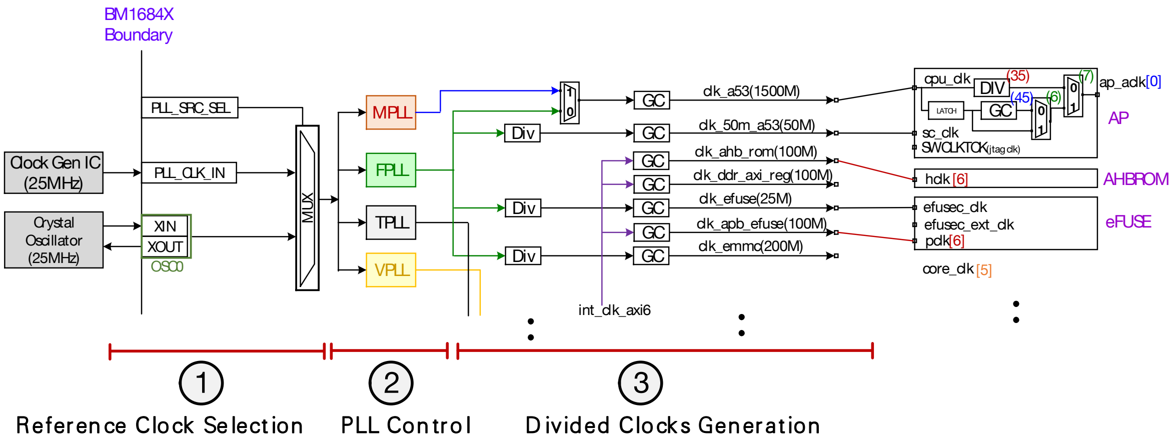


Figure 4 PLL Reference Clock Propagation

2.5.3 PLL Clock Scheme Overview in SG2300X

Different parts inside the chip works on different frequencies. As there is no "one-size-fits-all" PLL, SG2300X instantiates 6 PLLs to satisfy logic's clock requirements. (Do not take PLL inside vendor IPs, such as DDR and PCIE, into account).

- **MPLL**: the name is short for Main PLL. The output clocks of this PLL are mainly used in AP subsystem.
- **TPLL**: the name is short for TPU PLL. The output clocks of this PLL are mainly used in TPU subsystem.
- **VPLL**: the name is short for Video PLL. The output clocks of this PLL are mainly used in Video subsystem.
- **FPLL**: the name is short for Fixed PLL. This PLL generates fixed frequency clock, with output clock at 1.0 GHz or 2.0GHz. The output clocks of this PLL are mainly used in data and configuration bus.
- **DPLL0/1**: the name is short for DDR PLL. The output clocks of this PLL are mainly used in DDR subsystem.

And in order to reconfigure PLL clock frequency on the fly, MPLL, TPLL, VPLL and DPLL0/1 use FPLL as a backup. For details of this function, please refer to section 5 in this page.

2.5.4 PLL Control and Clock Generation

This section will focus on how to control the PLL so as to generate the wanted clock. Four topics will be discussed:

1. The specification of the PLL: which input parameters can alter the output frequency
2. The attributes of PLL's input reference clock.
3. The interface for SW to control the PLL.
4. The clock gating logic on PLL's output clock.

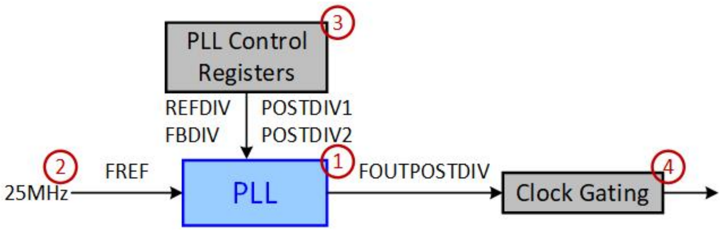


Figure 5 PLL Control

2.5.5 PLL Specification

For PLL, the Output Clock Frequency is calculated as:

*FOUTPOSTDIV = FREF*FBDIV / REFDIV / (POSTDIV1 * POSTDIV2)*

In above equation:

- **FREF**: Reference Clock Input (10MHz to 800MHz). SG2300X uses 25MHz reference clock.
- **FOUTPOSTDIV**: Output Clock (16MHz to 3200MHz)
- **REFDIV**: Reference divide value (1 to 63)
- **FBDIV**: Feedback divide value (16 to 320)
- **POSTDIV1**: Post Divide 1 setting (1 to 7)
- **POSTDIV2**: Post Divide 2 setting (1 to 7)

The selection of above variables should fit in the legal range. Additionally, other check points inside PLL are listed here:

- FOUTVCO = FREF * FBDIV / REFDIV （800MHz to 3200MHz）
- FREF / REFDIV > 10

2.5.5.1 Input Reference Clock

For reference clock, it is used in reset sequence. Only after certain reset sequence (1.5ms), PLL starts to work.

Since the 25MHz reference clock may come from crystal, the duty cycle/transition quality are not satisfying. So, all clocks for IP use PLL output directly or after division by logic.

2.5.5.2 PLL Control

Software together with a dedicated hardware module are in charge of the PLL control, especially the modification of PLL DIV values (REFDIV, FBDIV, POSTDIV1, POSTDIV2).

After Power-On Reset, embedded hardware is able to select the proper initial REFDIV, FBDIV, POSTDIV1 and POSTDIV2 values so that each PLL will generated clocks with expected frequency based on current chip mode.

During runtime, user can alter PLL's output by programming DIV values inside PLL Control Registers. Two modes are provided for programing PLL configurations: (1). Normal Config Mode; (2). Fast Config Mode

(1). Normal Config Mode

Every PLL DIV value can be modified in this mode. However, Normal Config Mode will take longer time as PLL Power Down is mandatory in this mode.

Suggested Programming Sequence for PLL Normal Config Mode:

Let's take TPLL configuration as an example.

Step1: Gate PLL output by clearing PLL Clock Enable Control Reg (0x500100C4) bit[1]

Step2: Modify TPLL Control Register (0x500100EC)

Step3: Polling PLL Status Register (0x500100C0) until: (1). PLL is locked again (bit[9] == 1) and (2). Updating sequence is finished (bit[1] == 0)

Step4: Un-gate PLL output clock by Setting PLL Clock Enable Control Reg (0x500100C4) bit[1]

When user programs the PLL Control Registers, internal hardware starts the following sequence:

1. The updating_pll_val bit is asserted immediately after user writes to PLL Control Registers, and user can check the value of this bit in PLL status register.
2. After hardware logic prepares the new DIV value for PLL, PLL's PD (Global Power Down) signal will be toggled so that PLL will work on the updated value.
3. PLL Lock goes high again when PLL's output is stable on new frequency.
4. Besides LOCK signal from PLL, internal logic will also wait for 240us then determine the modification sequence is finished and de-assert "updating_pll_val" bit.

User should keep polling PLL Status Register so as to ensure "updating_pll_val" bit field is de-asserted and whole sequence is finished. When the sequence is ongoing, internal logic will prevent initiating another modification.

(2). Fast Config Mode

When user programs the PLL Control Register, if MSB==1, it indicates it's a Fast Config. For Fast Config Mode, PLL will NOT experience a Power-Down Sequence when changing PLL frequency. This will achieve 1000x speedup versus Normal Config Mode. As you my expect, limitation exists in this mode. The limitation is that you can only change the value of FBDIV.

2.5.5.3 Gating on PLL output Clock

When users modify the frequency of a PLL, the output frequency may overshoot/exceed the expected frequency before PLL finally gets stable. This will lead to unwanted behavior or errors.

So output clock of PLL should be gated during configuration.

The generation of Clock Enable signal is shown below:

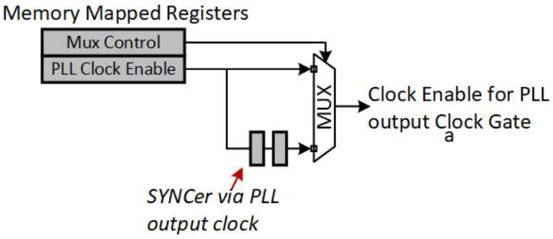

Memory Mapped Registers
Mux Control
PLL Clock Enable
MUX
Clock Enable for PLL output Clock Gate
a
SYNCer via PLL output clock

Figure 6 PLL Output Clock

User is able to control the above logic (Enable/MUX) by programming PLL Clock Enable Control Register (0x5001_00C4).

Both original PLL Clock Enable register and its synced version can be selected as Clock Enable PLL. This is because when PLL's frequency overshoots, the synchronizer may fail to work. There has to be a backup path.

Note that the address of register for controlling PLL Gating shall not be the same as those mentioned in previous section. Cos, once you touch the DIV related register, PLL will be powered-down.

2.5.6 Clock Structure

SG2300X TOP level clock structure is shown in the following figures

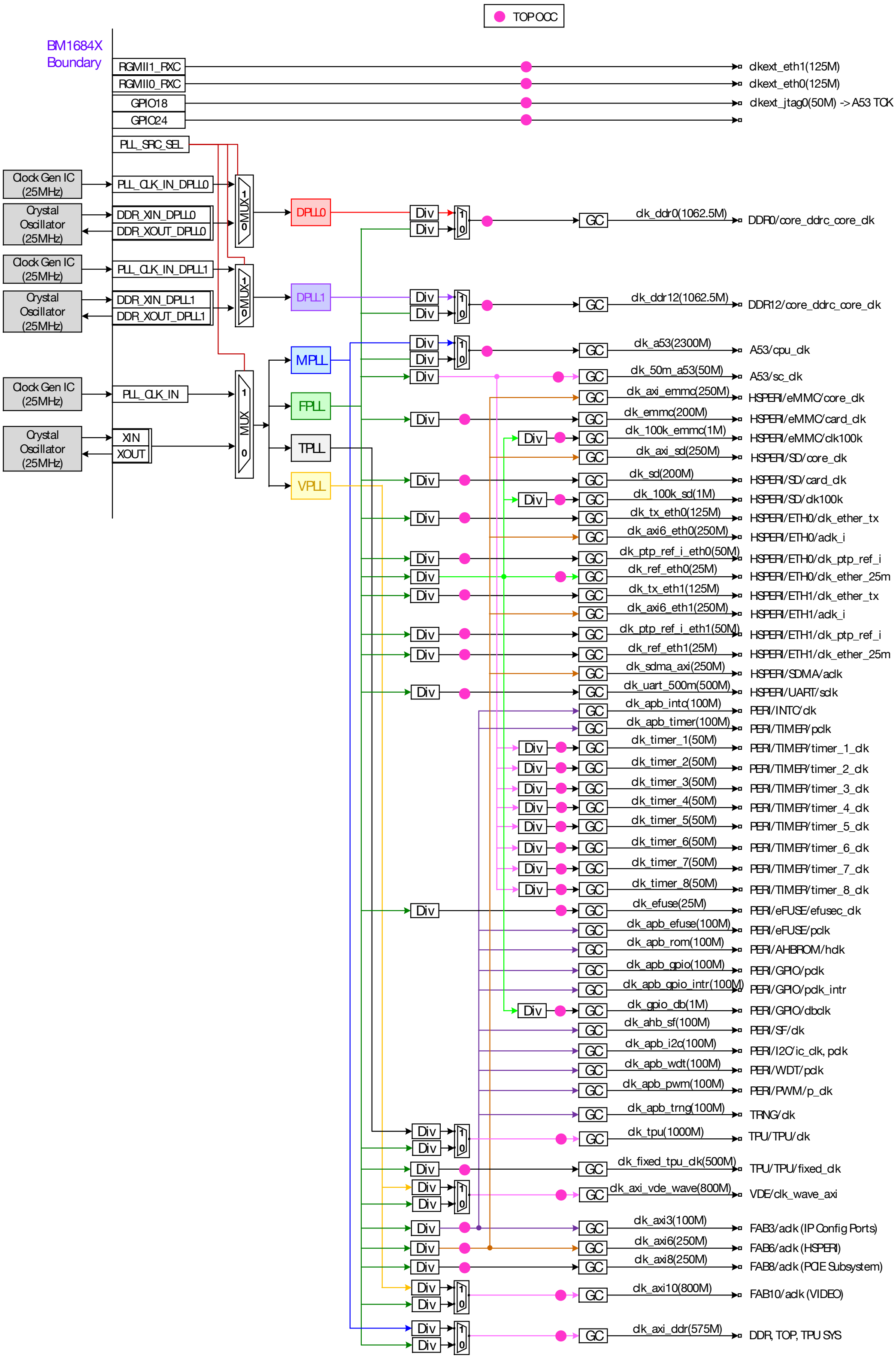


Figure 7 Clock Structure

2.5.7 Clock Frequency Definition

Please check the following table for SG2300X internal clock frequency information.

Table 5 Clock Frequency

Module Name	Clock Name	Clock frequency at normal mode (MHz)	Clock frequency at fast mode (MHz)	Clock frequency at safe mode (MHz)	Clock frequency at bypass mode (MHz)
PLL	**MPLL**	1825	2300	1150	25
PLL	**TPLL**	750	1000	500	25
PLL	**VPLL**	640	800	400	25
PLL	**FPLL**	1000	2000	1000	25
PLL	**DPLL0**	600	1062.5	462.5	25
PLL	**DPLL1**	600	1062.5	462.5	25
A53	clk_a53	1825	2300	1150	25
A53	clk_50m_a53	50	50	50	25
EMMC	clk_axi_emmc	250	250	250	25
EMMC	clk_emmc	100	100	100	25
EMMC	clk_100k_emmc	0.1	0.1	0.1	0.1
SD	clk_axi_sd	250	250	250	25
SD	clk_sd	100	100	100	25
SD	clk_100k_sd	0.1	0.1	0.1	0.1
ETH0	clk_tx_eth0	125	125	125	25
ETH0	clk_axi6_eth0	250	250	250	25
ETH0	clk_ptp_ref_i_eth0	50	50	50	25
ETH0	clk_ref_eth0	25	25	25	25
ETH1	clk_tx_eth1	125	125	125	25
ETH1	clk_axi6_eth1	250	250	250	25
ETH1	clk_ptp_ref_i_eth1	50	50	50	25
ETH1	clk_ref_eth1	25	25	25	25
SDMA	clk_sdma_axi	250	250	250	25
UART	clk_uart_500m	500	500	250	25
INTC	clk_apb_intc	100	100	100	25
TIMER	clk_apb_timer	100	100	100	25
TIMER	clk_timer_1	50	50	50	25
TIMER	clk_timer_2	50	50	50	25
TIMER	clk_timer_3	50	50	50	25
TIMER	clk_timer_4	50	50	50	25
TIMER	clk_timer_5	50	50	50	25
TIMER	clk_timer_6	50	50	50	25
TIMER	clk_timer_7	50	50	50	25
TIMER	clk_timer_8	50	50	50	25
EFUSE	clk_efuse	25	25	25	25
EFUSE	clk_apb_efuse	100	100	100	25
AHBROM	clk_ahb_rom	100	100	100	25
GPIO	clk_apb_gpio	100	100	100	25

GPIO	clk_apb_gpio_intr	100	100	100	25
GPIO	clk_gpio_db	0.1	0.1	0.1	0.1
SF	clk_ahb_sf	100	100	100	25
IIC	clk_apb_i2c	100	100	100	25
IIC	clk_apb_i2c	100	100	100	25
WDT	clk_apb_wdt	100	100	100	25
PWM	clk_apb_pwm	100	100	100	25
TRNG	clk_apb_trng	100	100	100	25
MMU	clk_axi8_mmu	250	250	250	25
PCIE	clk_axi8_pcie	250	250	250	25
CDMA	clk_axi8_cdma	250	250	250	25
ARM	clk_arm	456.25	575	287.5	25
AXISRAM	clk_axisram	456.25	575	287.5	25
GDMA	clk_gdma	456.25	575	287.5	25
TPU	clk_tpu	750	1000	500	25
TPU	clk_fixed_tpu_clk	500	500	500	25
TPU	clk_tpu_for_tpu_only	750	1000	500	25
HAU	clk_hau_ngs	750	1000	500	25
TSDMA	clk_tsdma	750	1000	500	25
dbg_i2c	clk_axi_dbg_i2c	456.25	575	287.5	25
PKA	clk_axi_pka	456.25	575	287.5	25
SPACC	clk_axi_spacc	456.25	575	287.5	25
VDE_WAVE521	clk_axi_vde_wave	640	800	400	25
VDE_WAVE521	clk_apb_vde_wave	100	100	100	25
VDE_AXI_BRG	clk_axi_vde_axi_bridge	640	800	400	25
VD0_WAVE0	clk_apb_vd0_wave0	100	100	100	25
VD0_WAVE0	clk_axi_vd0_wave0	640	800	400	25
VD0_WAVE1	clk_apb_vd0_wave1	100	100	100	25
VD0_WAVE1	clk_axi_vd0_wave1	640	800	400	25
VD0_JPEG0	clk_apb_vd0_jpeg0	100	100	100	25
VD0_JPEG0	clk_axi_vd0_jpeg0	640	800	400	25
VD0_JPEG1	clk_apb_vd0_jpeg1	100	100	100	25
VD0_JPEG1	clk_axi_vd0_jpeg1	640	800	400	25
VD0_VPP	clk_apb_vd0_vpp	100	100	100	25
VD0_VPP	clk_axi_vd0_vpp	640	800	400	25
VD1_WAVE0	clk_apb_vd1_wave0	100	100	100	25
VD1_WAVE0	clk_axi_vd1_wave0	640	800	400	25
VD1_WAVE1	clk_apb_vd1_wave1	100	100	100	25
VD1_WAVE1	clk_axi_vd1_wave1	640	800	400	25
VD1_JPEG0	clk_apb_vd1_jpeg0	100	100	100	25
VD1_JPEG0	clk_axi_vd1_jpeg0	640	800	400	25
VD1_JPEG1	clk_apb_vd1_jpeg1	100	100	100	25
VD1_JPEG1	clk_axi_vd1_jpeg1	640	800	400	25
VD1_VPP	clk_apb_vd1_vpp	100	100	100	25

VD1_VPP	clk_axi_vd1_vpp	640	800	400	25
FABRIC_AXI3	clk_axi3	100	100	100	25
FABRIC_AXI6	clk_axi6	250	250	250	25
FABRIC_AXI8	clk_axi8	250	250	250	25
FABRIC_AXI10	clk_axi10	640	800	400	25
FABRIC_AXI_DDR	clk_axi_ddr	456.25	575	287.5	25
DDR0	clk_ddr0	600	1062.5	462.5	25
DDR12	clk_ddr12	600	1062.5	462.5	25

2.5.8 Generation of divided clocks

This section describes how divided clocks are generated from PLLs. Basically, two types of division logics are implemented in SG2300X based on the application.

2.5.8.1 Type 1: single divider

Most clocks sent to IPs are generated under this type. The structure is straightforward, the source of the logic is fast clock from PLL, it enters a divider and outputs the slow clock. Additionally, a clock gating clock is attached as the last component in this logic.

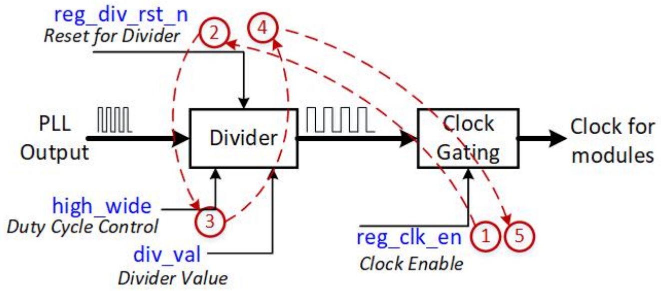


Figure 8 Single Divider

Users can control the Clock Divide Factor and whether High Level is wider in a cycle via memory mapped register “reg_div_val”, “reg_high_wide”, “sel_reg_div_val” and “sel_high_wide” as shown below. “initial_high_wide” and “initial_div_val” are fixed value decoded based on chip working mode.

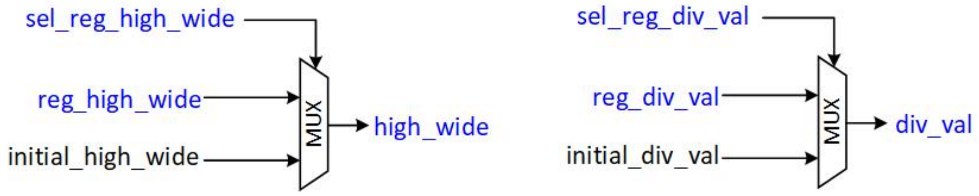


Figure 9 Divider Control Selection

The sequence of clock frequency modification is:

1. Gate the output clock by pull low clock enable memory-mapped register. Please wait for at

least two clock cycle time of current output clock, so as to ensure the clock gate logic already took effect.

2. Assert “reg_div_rst_n” to reset the Divider.
3. Modify the value of Clock Divide Factor (and High Wide if needed).
4. De-assert the “reg_div_rst_n” to restore divided clock with new frequency.
5. Un-gate the output clock.

2.5.8.2 *Type 2: mux out from two dividers*

Few clocks' generation need two dividers with clock mux as shown in the following figure. By doing this, the output clock can switch between two frequencies without any pause. Also, the output clock can cover wider range of frequency when it selects from different PLLs.

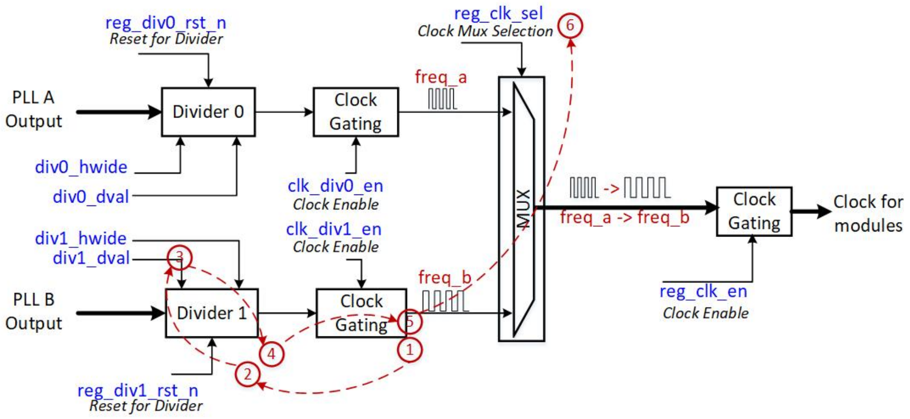


Figure 10 Type 2 Divider

Most part of the logic is similar to type 1 logic and already mentioned in above session. The clock mux is controlled by memory-mapped register.

For clocks generated with type 2 logic, the sequence of hot switch from Frequency A to Frequency B is shown below:

1. Gate the target branch clock by pull low clock enable MMR. Please wait for at least two clock cycle time of current output clock, so as to ensure the clock gate logic already took effect.
2. Assert “reg_div_rst_n” to reset the Divider.
3. Modify the value of Clock Divide Factor and High Wide if needed.
4. De-assert the “reg_div_rst_n” to restore divided clock with new frequency.
5. Un-gate the clock of target branch.
6. Change the value of “reg_clk_a53_sel” to alter the output of clock mux.

Note:

DDR clock should be gated when changing frequency. So the corresponding A53's instruction should be stored in AXI SRAM instead of DDR at that time.

2.6 Reset

2.6.1 SG2300X Reset sequence

SG2300X system reset is controlled by RESET IO; the timing sequence is shown in the following figure. The IO Reset signal is debounced by 25MHz clock (PLL reference clock) for about 30ms to avoid false reset caused by glitch. This 30ms will also ensure the reference clock input of PLL goes stable after chip power on, which needs at least 1ms. When the root reset is released, the PLLs get out of power down mode and start to work.

Reset for clock divider, reset for the system and reset for the AP will be released one by one after: 1. all PLLs are locked; 2. BISR (Built-In-Self-Repair) process is finished. RESET IO have Schmitt trigger to protect RESET IO from glitches.

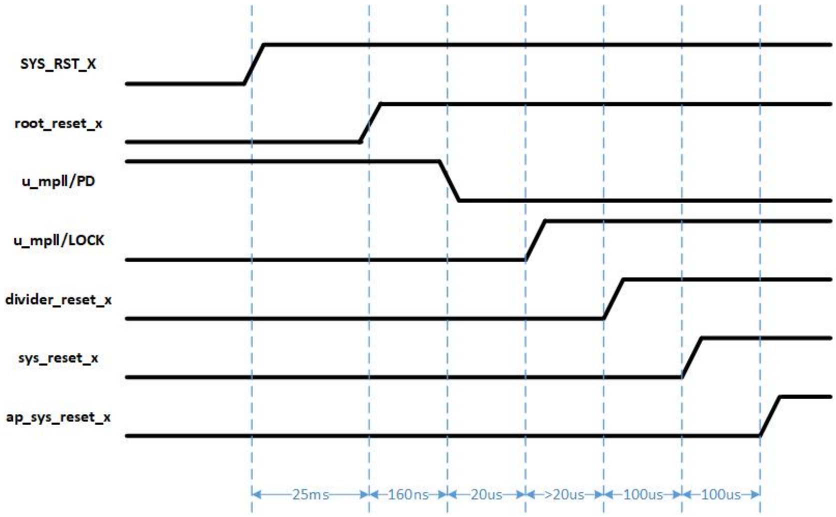


Figure 11 SG2300X Reset Sequence

There are some software resets controlled by register: watch dog reset, and ARM register controlled reset.

These resets may cause PLLs, reset for clock divider, reset for the system and reset for the AP active and the whole chip will be reset, except a little logic of REFCLK， IO reset.

These software resets must be active for at least 1us before taking effect so as to ensure PLLs are reset correctly.

2.7 Secure Design

2.7.1 Secure Engine

SG2300X supports following secure engines:

- **SPACC**:
  - AES128/192/256; DES/TDES; SM4
  - SHA-1/SHA-256; SM3
- **PKA**: ECC, RSA
- **TRNG**: True Random Number Generator

2.7.2 Security Master Firewall

The Security Master Firewall is located between IP's Master Port and Fabric's Slave Port.

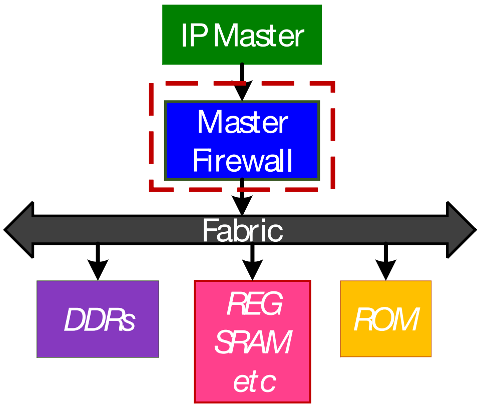


Figure 12 Security Master Firewall

The purpose of Security Master Firewall is to protect memory space from illegal transactions sent by this master.

When Security Master Firewall receives a transaction from Master, it will:

1. Replace the secure-bit (AW/RPORT[1]) of this request based on register configuration. By doing this, Master's original secure access can be replaced with non-secure access. Note that by default, the secure-bit coming from A53's masters will not be replaced. But for other masters, their secure bits will be replaced as non-secure by default.
2. Check the attribute of target address:
   a) The memory space of target address can be marked as forbidden or inaccessible, this is to ensure ROM and EFUSE and not reachable after boot. In this case, neither secure nor non-secure access can reach this memory space.
   b) If the memory space of target address is marked as secure, only secure access (replaced secure-bit == 1'b0) can reach this memory space.

3. For illegal access, the transaction will be discarded, and a DECERR response is returned. Meanwhile, an interrupt will be reported. Also "Address", "R/W" and "Burst Length" information of this transaction will be recorded in Security Register Files.

There is total 16 configurable spaces in system register.

Note that above function is only enabled when eFUSE-ADDR[0]-(Bit[1]|Bit[0]) (double bit) is programmed to 1. After power-up, dedicated hardware module will load the Addr[0]-(Bit[1]|Bit[0]) from eFUSE to determine whether to bypass Security Master Firewall in the following run time.

2.7.3 Configurable Security Register

There are system registers which indicate each master port of IP is secure or non-secure.

If a IP has to distinguish between secure and non-secure for each transaction, it should deal with this itself.

To ensure security, all master security parameter is default non-secure except A53, which is boot CPU.

Note that three memory spaces are in secure domain in default.

(1). Space for Configurable Security Registers (0x5000 E000 - 0x5000 FFFF)

(2). Space for ROM (0x0700 0000 - 0x070F FFFF)

(3). Space for eFUSE (0x5002 8000 - 0x5002 8FFF)

2.7.3.1 Security Register Description

SG2300X owns five sets of Security Registers for sake of floorplan and routing.

Each SE_REGs will control the masters inside that domain. User are encouraged to program all five domains every time, cos if you only set the registers in TOP, then masters in other four domains will not be manipulated.

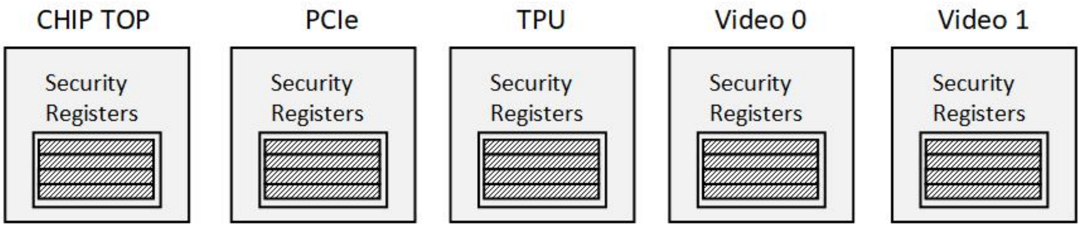


Figure 13 Security Registers

2.7.4 Secure Boot and Secure Debug

Several bits (double bit scheme) inside eFUSE Addr[0] are used to control whether to enable or disable the following functions:

(1). JTAG Debug Disable: bit[3]|bit[2]

When this bit is "1", on chip debug interfaces like JTAG are disabled.

An extra bit (JTAG Enable) is also provided in Configurable Security Register. JTAG function is disabled only when eFUSE is programmed (1'b1) and JTAG Enable bit is also cleared (1'b0).

(2). Onchip Boot: bit[5]||bit[4]

When this bit is "1", CPU core is forced to boot from ON-CHIP software (known as bootrom). In this condition, Hardware will clear BOOTSEL[3] so as to prevent booting from SPI Flash.

In TEST mode (TESTEN == 1), efuse can be accessed through IOs.

To ensure security, logic forbids the lower half of efuse accessed by IOs.

2.7.5 Secure Key

SG2300X provides a new function called Customer Secure Key. With Secure Key function, customer can save a 256-bit key and use this key in SPACC's encryption and decryption.

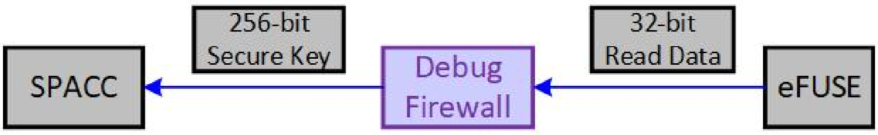


Figure 14 Custom Secure Key

To use this function, customer is required to program following contents into eFUSE in advance:

1. One bit (double-bit backup) to indicate whether Secure Key function is enabled
2. 256-bit secure key (double-bit backup)

(For eFUSE Bit Map, please refer to Appendix A: SG2300X eFuse Bit Map)

In power-on reset, Debug Firewall will automatically load the information from eFUSE to check whether Secure Key function is enabled.

If so, it will continue read out all 256-bit Secure Key and parse the Key together with Key Valid Flag to SPACC.

Then, Debug Firewall informs eFUSE to protect (block) the space of Secure Key. From now on, no device can ever access this space.

When SPACC receives the Secure Key, it set a bit inside Memory-Mapped Register. SW gets the updated status and starts to use Secure Key by polling this bit.

Note that in the whole run-time, SW is NOT able to read the content of Secure Key by any means, but purely use the Key.

2.8 Process, Voltage, Temperature Monitors

SG2300X has process monitor, voltage monitor and temperature monitor functions.

TEMP_P and TEMP_N are used as temperature monitor pins.

By setting different register value of 0x5001_0014 bit[4:0], there are total 12 points in SG2300X chip can be monitored. The relationship is shown in this table.

Table 6 Temperature Monitors

Selected Temperature Monitors	Register value of 0x5001_0014 bit[4:0]
tpu_sys	7
tpu	11
pcie_subsystem	2
ddr0a	3
ddr0b	4
ddr1	5
ddr2	6
cluster0/u_cortexa53	0
cluster1/u_cortexa53	1
videosubsystem0	8
videosubsystem1	9
Chip top	10
Total Number	**12**

VMON_P and VMON_N are used as voltage monitor pins.

By setting different register value of 0x5001_0014 bit[12:8], there are total 19 points in SG2300X chip can be monitored. The relationship is shown in this table.

Table 7 Voltage Monitors

Selected Voltage Monitors	Register value of 0x5001_0014 bit[12:8]
tpu_sys	15
pcie_subsystem	10
ddr0a	11
ddr0b	12
ddr1	13
ddr2	14
cluster0/u_cortexa53	4
cluster0/g_ca53_cpu[0]	0
cluster0/g_ca53_cpu[1]	1
cluster0/g_ca53_cpu[2]	2
cluster0/g_ca53_cpu[3]	3
cluster1/u_cortexa53	9

cluster1/g_ca53_cpu[0]	5
cluster1/g_ca53_cpu[1]	6
cluster1/g_ca53_cpu[2]	7
cluster1/g_ca53_cpu[3]	8
videosubsystem_0	16
videosubsystem_1	17
chip top	18
Total Number	**19**

There are process monitors in SG2300X.

To use process monitor, the software sequence is:

Step1: Set reg_pm_en（0x5001_0018 bit[3]） to 1'b1

Step2: Configure reg_pm_select（0x5001_0018 bit[2:1]）to select VT

2'd0 for lvt；2'd1 for svt；2'd2 for ulvt

Step3: Set reg_pm_start（0x5001_0018 bit[0]）

Step4: Read toreg_pm_count as shown in this table.

Table 8 Process Monitors

Selected Process Monitors	Register value of 0x5001_0014 bit[12:8]
tpu (reserved)	0x5001_011C[15:0] 0x5001_0120[15:0] 0x5001_0124[15:0] 0x5001_0128[15:0] 0x5001_012C[15:0] 0x5001_0130[15:0] 0x5001_0134[15:0] 0x5001_0138[15:0] 0x5001_013C[15:0]
pcie_subsystem	0x5001_0108[15:0]
ddr0a	0x5001_010C[15:0]
ddr0b	0x5001_0110[15:0]
ddr1	0x5001_0114[15:0]
ddr2	0x5001_0118[15:0]
cluster0/u_cortexa53	0x5001_0100[15:0]
cluster1/u_cortexa53	0x5001_0104[15:0]
videosubsystem_0	0x5001_0140[15:0]
videosubsystem_0	0x5001_0144[15:0]
SG2300X_core	0x5001_0148[15:0]

2.9 Performance Monitors

Three types of Performance Monitor are designed to cover different monitor operation of data flow:

- AXI Performance Monitor
- TPU Performance Monitor
- GDMA Performance Monitor

2.9.1 AXI Performance Monitor

2.9.1.1 Overview

AXI Bus Performance Monitors are located inside AXI Fab5, AXI Fab24 and AXI Fab25 so as to check the condition (count the number of bytes) of AXI data transfer within a certain period of time.

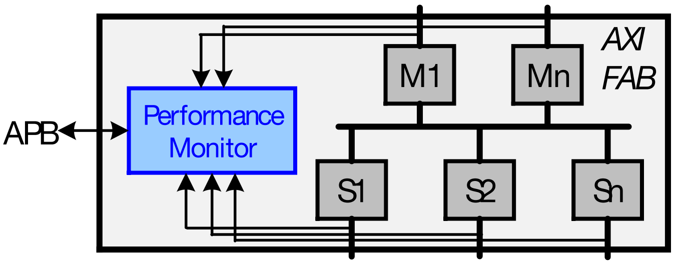


Figure 15 AXI Bus Performance Monitors

As shown in the above figure, the Monitor has the access to every Master/Slave port of the fabric. However, the Monitor only listens to one selected port based on configuration. User configures setting and reads out result through an APB interface.

2.9.1.2 Function Description

Monitor counts the data amount (number of bytes) of read and write simultaneously based on the period set by user.

1. Monitor Modes

This Monitor supports two Monitor-Modes:

(1). **Routine** Monitor Mode: in this mode, after user sets the period, the Monitor Module keeps counting the number of bytes for continues periods and tries to save all records also the latest record.

(2). **Peak** Monitor Mode: in this mode, after user sets the period, the Monitor Module keeps counting the number of bytes for continues periods and only records the largest value.

2. Reporting Methods

(1). For Routine Monitor Mode: Monitor Module will record result in both Trace Mode and Sampling Mode as shown below:

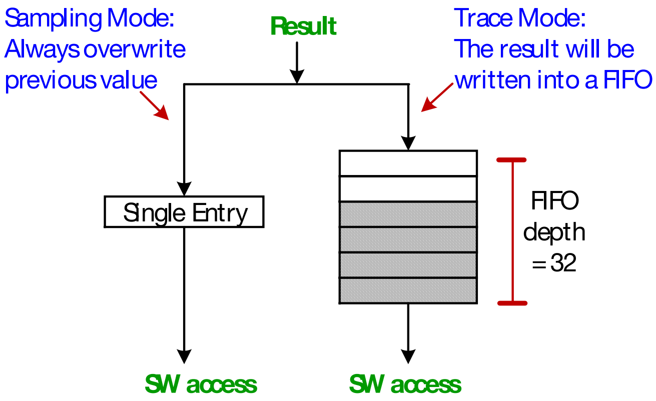


Figure 16 Result Report

Note that, if the FIFO is full as software didn't read out the result, Monitor will keep pushing in and overflow FIFO. Apart from the results in FIFO, Monitor should also provide SW the available number of entries inside the FIFO.

(2). For Peak Monitor Mode: monitor will only save the largest value for SW.

2.9.1.3 Register Definition

Please refer to Appendix B - Register Table for more details.

2.9.2 TPU Performance Monitor

2.9.2.1 Overview

TPU Performance Monitor is located inside TPU. The purpose of this module is to check the condition of TPU's computation efficiency, number of instructions, etc within a certain period of time. After one period, Monitor is able to send out the results to DDR.

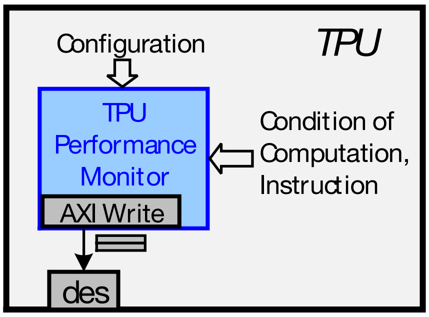


Figure 17 TPU Performance Monitor

Unlike AXI performance Monitor, the configuration for TPU Monitor is provided by TPU directly.

2.9.2.2 Function Description

2.9.2.2.1 Monitor Enable and Disable

SW can enable or disable this monitor by programming TPU registers.

2.9.2.2.2 Monitor Items

The "period" concept in TPU Performance Monitor is not the same as AXI Performance Monitor. In TPU performance monitor, "period" is the life cycle of an instruction.

When Monitor is enabled, a 32-bit counter is running all the time. The value of this counter is used as timestamp.

During a period, the following information should be recorded:

1) Start Time of Instruction
2) End Time of Instruction
3) Instruction ID
4) Computation Load
5) Number of Read Instruction
6) Number of Read Stall Cycles
7) Number of Write Instruction

2.9.2.2.3 Result Reporting

When a period ends, Monitor should send out the results to DDR.

1. The data pattern of results is shown below:

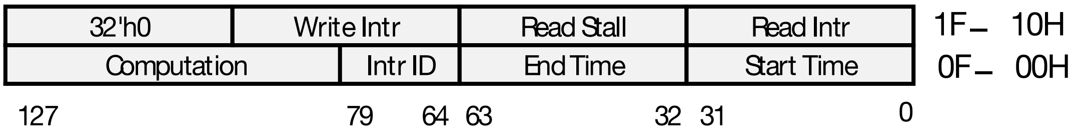


Figure 18 TPU PM Result Reporting

2. For the address of AXI access, software configures the range (start address, end address) through TPU memory mapped registers.

2.9.2.3 Register Definition

Please refer to TPU Register Description for more details.

2.9.3 GDMA Performance Monitor

2.9.3.1 Overview

GDMA Performance Monitor is located inside GDMA. The purpose of this module is to check the condition of GDMA's transfer efficiency within a certain period of time. After one period, Monitor is able to send out the results via AXI AW/W Channel belongs to GDMA's des master port.

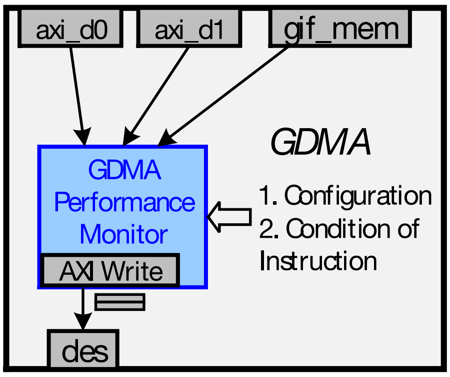


Figure 19 GDMA Performance Monitor

Unlike AXI performance Monitor, the configuration for GMDA Monitor is provided by GDMA directly. The Memory-Mapped Register should be instantiated by GDMA designer.

2.9.3.2 Function Description

2.9.3.2.1 Monitor Enable and Disable

User can enable or disable this monitor by programming GDMA registers.

2.9.3.2.2 Monitor Items

The "period" concept in GDMA Performance Monitor is same as TPU Performance Monitor.

32-bit timing counter is also included.

During a period, the following information should be recorded:

1) Start Time of Instruction

2) End Time of Instruction

3) Instruction ID

4) Data Throughput: During a period, data transfer throughput will be checked. Monitor has 9 32-bit counters for data amount detection: AXI_D0_AW, AXI_D0_W, AXI_D0_AR, AXI_D1_AW, AXI_D1_W, AXI_D1_AR, GIF_MEM_AW, GIF_MEM_W, GIF_MEM_AR.

a) For AXI: Once Monitor detects a Read/Write request (aw/rvalid & aw/rready), it calculates the data transfer amount based on burst length and burst size and adds the value to read or write counter.

Additionally, for write data, Monitor should also count the real write bytes based on WVALID, WREADY, WSTRB. Read and write value are logged simultaneously with three 32-bit counters.

b) For GIF: The behavior is similar to AXI, except (mwrite/mread & saccept) is used to indicate Write/Read Request, while represents the byte count. Write strobe is involved in third counter.

5) Valid Transfer Cycles

6) Stall Cycles

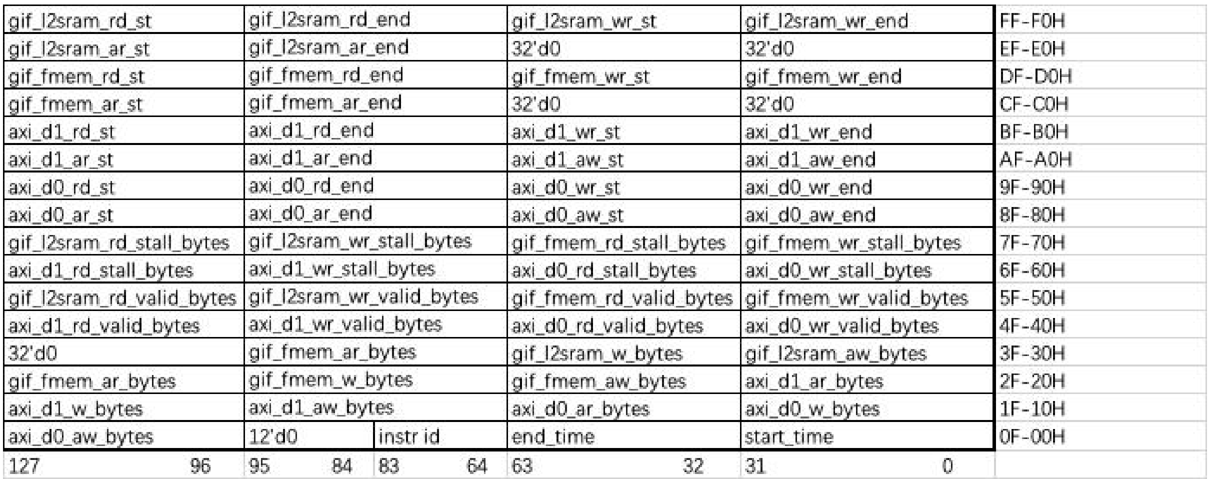
7) Start Time of Each Channel

8) End Time of Each Channel

2.9.3.2.3Result Reporting

The Result Reporting Scheme is similar to TPU performance Monitor. The result will be written out via AXI write channel.

The data pattern of results is shown below:



gif_l2sram_rd_st	gif_l2sram_rd_end		gif_l2sram_wr_st	gif_l2sram_wr_end	FF-F0H
gif_l2sram_ar_st	gif_l2sram_ar_end		32'd0	32'd0	EF-E0H
gif_fmem_rd_st	gif_fmem_rd_end		gif_fmem_wr_st	gif_fmem_wr_end	DF-D0H
gif_fmem_ar_st	gif_fmem_ar_end		32'd0	32'd0	CF-C0H
axi_d1_rd_st	axi_d1_rd_end		axi_d1_wr_st	axi_d1_wr_end	BF-B0H
axi_d1_ar_st	axi_d1_ar_end		axi_d1_aw_st	axi_d1_aw_end	AF-A0H
axi_d0_rd_st	axi_d0_rd_end		axi_d0_wr_st	axi_d0_wr_end	9F-90H
axi_d0_ar_st	axi_d0_ar_end		axi_d0_aw_st	axi_d0_aw_end	8F-80H
gif_l2sram_rd_stall_bytes	gif_l2sram_wr_stall_bytes		gif_fmem_rd_stall_bytes	gif_fmem_wr_stall_bytes	7F-70H
axi_d1_rd_stall_bytes	axi_d1_wr_stall_bytes		axi_d0_rd_stall_bytes	axi_d0_wr_stall_bytes	6F-60H
gif_l2sram_rd_valid_bytes	gif_l2sram_wr_valid_bytes		gif_fmem_rd_valid_bytes	gif_fmem_wr_valid_bytes	5F-50H
axi_d1_rd_valid_bytes	axi_d1_wr_valid_bytes		axi_d0_rd_valid_bytes	axi_d0_wr_valid_bytes	4F-40H
32'd0	gif_fmem_ar_bytes		gif_l2sram_w_bytes	gif_l2sram_aw_bytes	3F-30H
gif_fmem_ar_bytes	gif_fmem_w_bytes		gif_fmem_aw_bytes	axi_d1_ar_bytes	2F-20H
axi_d1_w_bytes	axi_d1_aw_bytes		axi_d0_ar_bytes	axi_d0_w_bytes	1F-10H
axi_d0_aw_bytes	12'd0	instr id	end_time	start_time	0F-00H
127 96	95 84	83 64	63 32	31 0	

Figure 20 GDMA PM Result Reporting

2.9.3.3 Register Definition

Please refer to GDMA Register Description for more details.

2.10 Top Registers

Please check Appendix B for details.

3 Power Domain and Power Sequence

SG2300X implements multi-voltage design to minimize the power consumption. Different blocks will work at various voltage based on their application. The following figure shows the partition of block with different voltage.

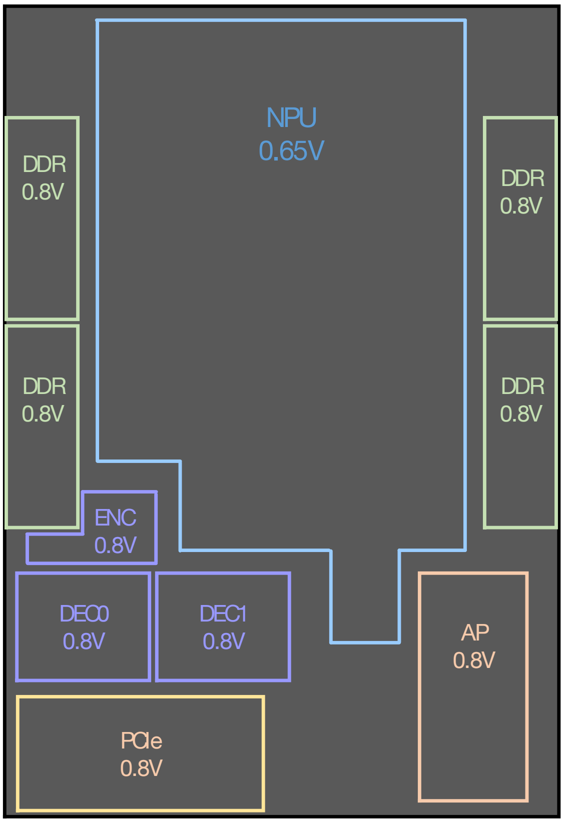


Figure 21 SG2300X Power Domain

In this section, topics will be elaborated related to power domain, power up/down sequence.

1. Chip Power Domain: describes the relationship between bump and power domain.
2. Chip Power Up Sequence: the steps about powering up the chip
3. Block Power Down Sequence: the steps about shutting down blocks.

3.1.1 Power domain

The following table shows the relationship between Power Bump and Voltage. Additionally, whether a block controlled by this Power Bump can be power down in application is also shown.

Table 9 SG2300X Power Domain

Power Bump	Voltage (V)	Variation	Can be power

		(%)	down in application?
VDD_TPU	0.65	10%	N
VDDC	0.8	10%	N
VDDIO_FPLL VDDIO_MPLL VDDIO_TPLL VDDIO_VPLL VDDIO_DPLL0 VDDIO_DPLL1	0.8	10%	N
DDR0A_VDD_DDR DDR1_VDD_DDR DDR2_VDD_DDR DDR0B_VDD_DDR	0.8	-7% ~+10%	N
DDR0A_VDDQ DDR1_VDDQ DDR2_VDDQ DDR0B_VDDQ	1.1	5%	N
DDR0A_VDDQLP DDR1_VDDQLP DDR2_VDDQLP DDR0B_VDDQLP	0.6	5%	N
DDR0A_VAA DDR1_VAA DDR2_VAA DDR0B_VAA	1.8	10%	N
VDDIO_EMMC_18	1.8	10%	N
VDDIO_EMMC_33	3.3	10%	N
PCIE_vph_phy0 PCIE_vph_phy1	1.8	Supply Voltage at PHY Bumps -7%~+10% AC noise: < 3% Vpp of DC level	Y
PCIE_vp_phy0 PCIE_vp_phy1	0.8	Supply Voltage at PHY Bumps -7%~+5% AC noise: < 5% Vpp of DC level	Y
PCIE_vptx0_phy0 PCIE_vptx1_phy0 PCIE_vptx2_phy0 PCIE_vptx3_phy0 PCIE_vptx4_phy0 PCIE_vptx5_phy0	0.8	Supply Voltage at PHY Bumps -7%~+5%	Y

PCIE_vptx6_phy0 PCIE_vptx7_phy0 PCIE_vptx0_phy1 PCIE_vptx1_phy1 PCIE_vptx2_phy1 PCIE_vptx3_phy1 PCIE_vptx4_phy1 PCIE_vptx5_phy1 PCIE_vptx6_phy1 PCIE_vptx7_phy1		AC noise: < 5% Vpp of DC level	
PCIE_vpdig_phy0 PCIE_vpdig_phy1	0.8	Supply Voltage at PHY Bumps -7%~+5% AC noise: < 5% Vpp of DC level	Y
VDDIO_PCIE_PAD_33	3.3	10%	N
VDDIO_PCIE_PAD_18	1.8	10%	N
VDDIO_RGM_33	1.8	10%	N
VDDIO_RGM_18	1.8	10%	N
VDDIO_R VDDIO_B	1.8	10%	N
VDDIO_SENSOR	1.8	10%	N
VDDIO_OSC_DDR0	1.8	10%	N
VDDIO_OSC_DDR1	1.8	10%	N
VQPS	1.8	5%	Y

3.1.2 SG2300X Power Up Sequence

0ms DDR*_VAA (1.8V) / VDDIO18(all the other 1.8V except DDR 1.8V and VQPS 1.8V: VDDIO_OSC_DDR0, VDDIO_OSC_DDR1, VDDIO_R, VDDIO_SENSOR, VDDIO_RGM_33, VDDIO_RGM_18, VDDIO_B) →

1ms VDDC (0.8V) →

2ms VDDIO_PCIE_PAD_33 / VDDIO_EMMC_33 (3.3V) →

3ms VDD_DDR (0.8V DDRPHY+PUB) / VDDIO_FPLL (0.8V) / VDDIO_MPLL (0.8V) / VDDIO_TPLL (0.8V) / VDDIO_VPLL (0.8V) / VDDIO_DPLL0 (0.8V) / VDDIO_DPLL1 (0.8V) →

4ms P08_PWR_GOOD 0-1 →

6ms PCIE_PWR_GOOD 0-1 →

7ms VDD_TPU(0.65V) →

8ms TPU_PWR_GOOD 0-1 →

9ms	DDR*_VDDQ (1.1V) →
10ms	DDR*_VDDQLP (1.1V or 0.6V) →
12ms	TPUMEM_PWR_GOOD 0-1 →
13ms	VQPS （1.8V） →
15ms	release IO reset →
45ms	DDR_PWR_GOOD 0-1 (30ms after release IO reset →

3.1.3 Block Power Down Sequence

After power up, the sequence of powering down a block:

(1). Gate the clock of this block

(2). Clear the Power Good Inputs of this block to 0 (use memory mapped register)

(3). Assert reset of this block

(4). Remove the power supply to this block

If the user wants to recover the power supply of a block, the sequence should be:

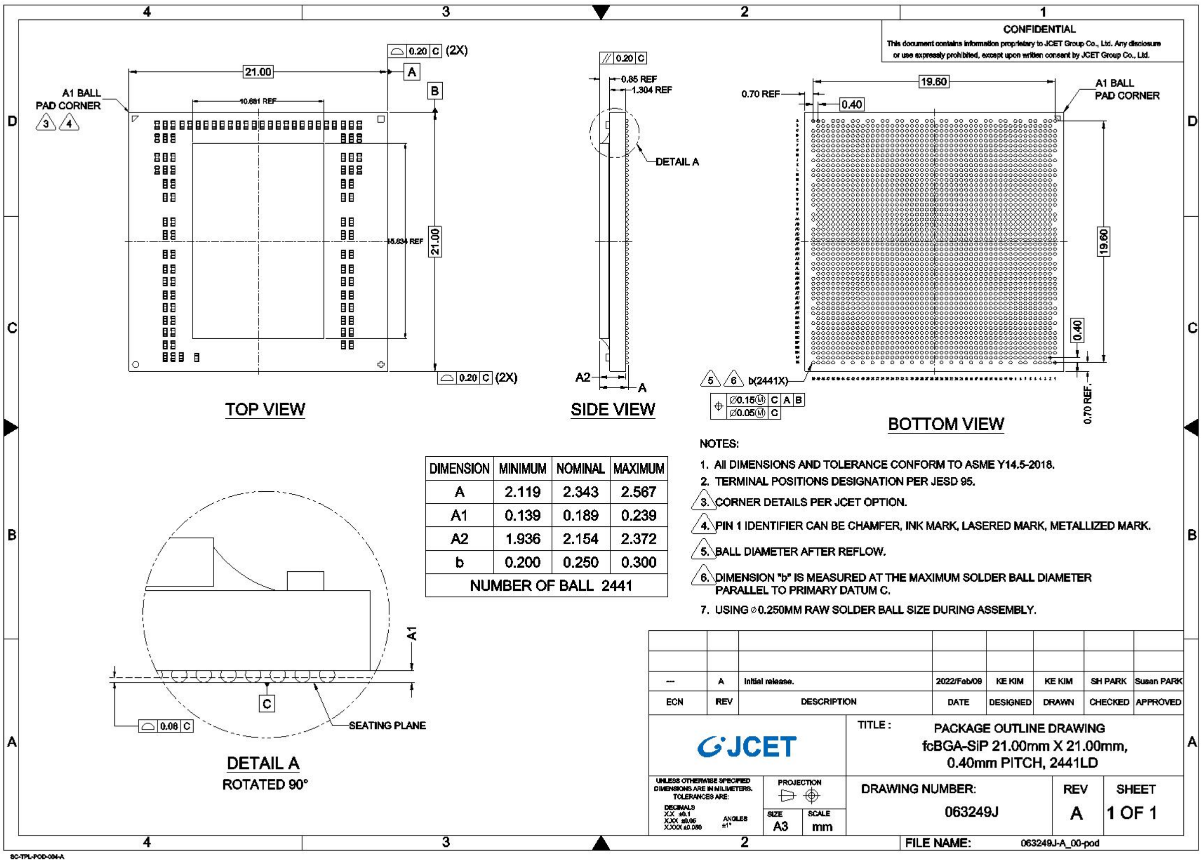
(1). Restore the power supply (controls the MCU on PCB)

(2). De-assert reset of this block

(3). Set Power Good Inputs of this block to 1.

(4). Enable the clock of this block

4 Package Specifications



CONFIDENTIAL

This document contains information proprietary to JCET Group Co., Ltd. Any disclosure or use expressly prohibited, except upon written consent by JCET Group Co., Ltd.

DIMENSION	MINIMUM	NOMINAL	MAXIMUM
A	2.119	2.343	2.567
A1	0.139	0.189	0.239
A2	1.936	2.154	2.372
b	0.200	0.250	0.300
NUMBER OF BALL 2441			

NOTES:

1. All DIMENSIONS AND TOLERANCE CONFORM TO ASME Y14.5-2018.
2. TERMINAL POSITIONS DESIGNATION PER JESD 95.
3. CORNER DETAILS PER JCET OPTION.
4. PIN 1 IDENTIFIER CAN BE CHAMFER, INK MARK, LASERED MARK, METALLIZED MARK.
5. BALL DIAMETER AFTER REFLOW.
6. DIMENSION "b" IS MEASURED AT THE MAXIMUM SOLDER BALL DIAMETER PARALLEL TO PRIMARY DATUM C.
7. USING ⌀0.250MM RAW SOLDER BALL SIZE DURING ASSEMBLY.

---	A	Initial release.	2022/Feb/09	KE KIM	KE KIM	SH PARK	Susan PARK
ECN	REV	DESCRIPTION	DATE	DESIGNED	DRAWN	CHECKED	APPROVED

JCET

TITLE : PACKAGE OUTLINE DRAWING
fcBGA-SiP 21.00mm X 21.00mm,
0.40mm PITCH, 2441LD

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN MILIMETERS. TOLERANCES ARE:
DECIMALS
X.X ±0.1
X.XX ±0.05
X.XXX ±0.050
ANGLES ±1°

PROJECTION

SIZE	SCALE
A3	mm

DRAWING NUMBER:	REV	SHEET
063249J	A	1 OF 1

FILE NAME: 063249J-A_00-pod

5 Pin Information

5.1 General Purpose IO

Digital IO Name	Value of reg_pin_mux_sel under none-test mode (If the table cell is blank, it indicates this condition is not legal)				Default Pull Up or Pull Down	Voltage	Description
	2'b00	2'b01	2'b10	2'b11			
PCIEE_CLKREQ_X	PCIEE_CLKREQ_X				Pull Up	3.3/1.8	PCIe Clock Request X (default voltage is 3.3V)
PCIER_CLKREQ_X	PCIER_CLKREQ_X				Pull Up	3.3/1.8	PCIe Clock Request X (default voltage is 3.3V)
PCIEE_RST_X	PCIEE_RST_X				Pull Down	3.3/1.8	PCIe Reset X (default voltage is 3.3V)
PCIER_RST_X	PCIER_RST_X				Pull Down	3.3/1.8	PCIe Reset X (default voltage is 3.3V)
PCIEE_WAKEUP_X_OUT	PCIEE_WAKEUP_X_OUT				Pull Up	3.3/1.8	PCIe Wakeup X OUT (default voltage is 3.3V)
PCIER_WAKEUP_X_OUT	PCIER_WAKEUP_X_OUT				Pull Up	3.3/1.8	PCIe Wakeup X OUT (default voltage is 3.3V)
SPIF_CLK_SEL1	SPIF_CLK_SEL1	GPIO32			N.A	1.8	SPI Flash clock select[0]
SPIF_WP_X	SPIF_WP_X	GPIO33			N.A	1.8	SPI Flash Write Protect
SPIF_HOLD_X	SPIF_HOLD_X	GPIO34			N.A	1.8	SPI Flash Hold
SPIF_SDI	SPIF_SDI	GPIO35			N.A	1.8	SPI NAND data input
SPIF_CS_X	SPIF_CS_X	GPIO36			N.A	1.8	SPI NAND chip select
SPIF_SCK	SPIF_SCK	GPIO37			N.A	1.8	SPI NAND clock
SPIF_SDO	SPIF_SDO	GPIO38			N.A	1.8	SPI NAND data output
EMMC_WP	EMMC_WP				N.A	1.8	eMMC write protect signal
EMMC_CD_X	EMMC_CD_X				Pull Up	1.8	eMMC card detect signal, low active
EMMC_RST_X	EMMC_RST_X				N.A	1.8	eMMC card reset signal
EMMC_PWR_EN	EMMC_PWR_EN				N.A	1.8	eMMC card power enable signal
SDIO_CD_X	SDIO_CD_X	GPIO39			Pull Up	1.8	SDIO card detect signal, low active
SDIO_WP	SDIO_WP	GPIO40			N.A	1.8	SDIO write protect signal
SDIO_RST_X	SDIO_RST_X	GPIO41			N.A	1.8	SDIO card reset signal

SDIO_PWR_EN	SDIO_PWR_EN	GPIO42			N.A	1.8	SDIO card power enable signal
RGMII0_TXD0	GPIO43	RGMII0_TXD0			N.A	1.8	RGMII transmit data
RGMII0_TXD1	GPIO44	RGMII0_TXD1			N.A	1.8	RGMII transmit data
RGMII0_TXD2	GPIO45	RGMII0_TXD2			N.A	1.8	RGMII transmit data
RGMII0_TXD3	GPIO46	RGMII0_TXD3			N.A	1.8	RGMII transmit data
RGMII0_TXCTRL	GPIO47	RGMII0_TXCTRL			N.A	1.8	RGMII transmit control
RGMII0_RXD0	GPIO48	RGMII0_RXD0			N.A	1.8	RGMII receive data
RGMII0_RXD1	GPIO49	RGMII0_RXD1			N.A	1.8	RGMII receive data
RGMII0_RXD2	GPIO50	RGMII0_RXD2			N.A	1.8	RGMII receive data
RGMII0_RXD3	GPIO51	RGMII0_RXD3			N.A	1.8	RGMII receive data
RGMII0_RXCTRL	GPIO52	RGMII0_RXCTRL			N.A	1.8	RGMII receive control
RGMII0_TXC	GPIO53	RGMII0_TXC			N.A	1.8	RGMII transmit clock
RGMII0_RXC	GPIO54	RGMII0_RXC			N.A	1.8	RGMII receive clock
RGMII0_REFCLKO	GPIO55	RGMII0_REFCLKO			N.A	1.8	Reference clock output
RGMII0_IRQ	GPIO56	RGMII0_IRQ			N.A	1.8	Interrupt request from PHY
RGMII0_MDC	GPIO57	RGMII0_MDC			N.A	1.8	RGMII management clock
RGMII0_MDIO	GPIO58	RGMII0_MDIO			N.A	1.8	RGMII management data IO
RGMII1_TXD0	GPIO59	RGMII1_TXD0			N.A	1.8	RGMII transmit data
RGMII1_TXD1	GPIO60	RGMII1_TXD1			N.A	1.8	RGMII transmit data
RGMII1_TXD2	GPIO61	RGMII1_TXD2			N.A	1.8	RGMII transmit data
RGMII1_TXD3	GPIO62	RGMII1_TXD3			N.A	1.8	RGMII transmit data
RGMII1_TXCTRL	GPIO63	RGMII1_TXCTRL			N.A	1.8	RGMII transmit control
RGMII1_RXD0	GPIO64	RGMII1_RXD0			N.A	1.8	RGMII receive data
RGMII1_RXD1	GPIO65	RGMII1_RXD1			N.A	1.8	RGMII receive data
RGMII1_RXD2	GPIO66	RGMII1_RXD2			N.A	1.8	RGMII receive data
RGMII1_RXD3	GPIO67	RGMII1_RXD3			N.A	1.8	RGMII receive data
RGMII1_RXCTRL	GPIO68	RGMII1_RXCTRL			N.A	1.8	RGMII receive control
RGMII1_TXC	GPIO69	RGMII1_TXC			N.A	1.8	RGMII transmit clock
RGMII1_RXC	GPIO70	RGMII1_RXC			N.A	1.8	RGMII receive clock
RGMII1_REFCLKO	GPIO71	RGMII1_REFCLKO			N.A	1.8	Reference clock output
RGMII1_IRQ	GPIO72	RGMII1_IRQ			N.A	1.8	RGMII interrupt request from PHY
RGMII1_MDC	GPIO73	RGMII1_MDC			N.A	1.8	RGMII management clock
RGMII1_MDIO	GPIO74	RGMII1_MDIO			N.A	1.8	RGMII management data

							IO
PWM0	PWM0	GPIO75			N.A	1.8	Outputs of PWM0
PWM1	PWM1	GPIO76			N.A	1.8	Outputs of PWM1
FAN0	GPIO77	FAN0			N.A	1.8	Outputs of PWM0
FAN1	GPIO78	FAN1			N.A	1.8	Outputs of PWM1
IIC0_SDA	IIC0_SDA(OD)	GPIO79			N.A	1.8	IIC0 master data wire / slave data wire
IIC0_SCL	IIC0_SCL(OD)	GPIO80			N.A	1.8	IIC0 master clock / slave clock
IIC1_SDA	IIC1_SDA(OD)	GPIO81			N.A	1.8	IIC1 master data wire / slave data wire
IIC1_SCL	IIC1_SCL(OD)	GPIO82			N.A	1.8	IIC1 master clock / slave clock
IIC2_SDA	GPIO83	IIC2_SDA(OD)			N.A	1.8	IIC2 master data wire / slave data wire
IIC2_SCL	GPIO84	IIC2_SCL(OD)			N.A	1.8	IIC2 master clock / slave clock
UART0_TX	UART0_TX	GPIO85			Pull Up	1.8	UART0 transmit data
UART0_RX	UART0_RX	GPIO86			Pull Up	1.8	UART0 receive data
UART1_TX	UART1_TX	GPIO87			Pull Up	1.8	UART1 transmit data
UART1_RX	UART1_RX	GPIO88			Pull Up	1.8	UART1 receive data
UART2_TX	GPIO89	UART2_TX			Pull Up	1.8	UART2 transmit data
UART2_RX	GPIO90	UART2_RX			Pull Up	1.8	UART2 receive data
GPIO0	GPIO0		DEBUG_0		Pull Down	1.8	GPIO
GPIO1	TPUMEM_PWR_GOOD	GPIO1	DEBUG_1		Pull Down	1.8	GPIO
GPIO2	PCIE_PWR_GOOD	GPIO2	DEBUG_2		Pull Down	1.8	GPIO
GPIO3	TPU_PWR_GOOD	GPIO3	DEBUG_3		Pull Down	1.8	GPIO
GPIO4	PLL_LOCKO	VD0_WAVE0_UART_TX	DEBUG_4	GPIO4	Pull Down	1.8	GPIO
GPIO5	GPIO5	VD0_WAVE0_UART_RX	DEBUG_5		Pull Down	1.8	GPIO
GPIO6	GPIO6	VD0_WAVE1_UART_TX	DEBUG_6		Pull Down	1.8	GPIO
GPIO7	GPIO7	VD0_WAVE1_UART_RX	DEBUG_7		Pull Down	1.8	GPIO
GPIO8	GPIO8		DEBUG_8		Pull Down	1.8	GPIO
GPIO9	GPIO9	VD1_WAVE0_UART_TX	DEBUG_9		Pull Down	1.8	GPIO
GPIO10	GPIO10	VD1_WAVE0_UART_RX	DEBUG_10		Pull Down	1.8	GPIO
GPIO11	GPIO11	VD1_WAVE1_UART_TX	DEBUG_11		Pull Down	1.8	GPIO
GPIO12	GPIO12	VD1_WAVE1_UART_RX	DEBUG_12		Pull Down	1.8	GPIO
GPIO13	GPIO13	VDE_WAVE_UART_TX	DEBUG_13	UART0_RTS	Pull Down	1.8	GPIO

GPIO14	GPIO14	VDE_WAVE_UART_RX	DEBUG_14	UART0_CTS	Pull Down	1.8	GPIO
GPIO15	JTAG_1_2_SEL	UART1_RTS	DEBUG_15	GPIO15	Pull Down	1.8	GPIO
GPIO16	GPIO16	UART1_CTS	DEBUG_16		Pull Down	1.8	GPIO
GPIO17	JTAG0_TDO	GPIO17	DEBUG_17		Pull Down	1.8	GPIO
GPIO18	JTAG0_TCK	GPIO18	DEBUG_18		Pull Down	1.8	GPIO
GPIO19	JTAG0_TDI	GPIO19	DEBUG_19		Pull Down	1.8	GPIO
GPIO20	JTAG0_TMS	GPIO20	DEBUG_20		Pull Down	1.8	GPIO
GPIO21	JTAG0_TRST_X	GPIO21	DEBUG_21		Pull Up	1.8	GPIO
GPIO22	JTAG0_SRST_X	GPIO22	DEBUG_22		Pull Up	1.8	GPIO
GPIO23	JTAG_1_2_TDO	GPIO23	DEBUG_23		Pull Down	1.8	GPIO
GPIO24	JTAG_1_2_TCK	GPIO24	DEBUG_24		Pull Down	1.8	GPIO
GPIO25	JTAG_1_2_TDI	GPIO25	DEBUG_25		Pull Down	1.8	GPIO
GPIO26	JTAG_1_2_TMS	GPIO26	DEBUG_26		Pull Down	1.8	GPIO
GPIO27	JTAG_1_2_TRST_X	GPIO27	DEBUG_27		Pull Up	1.8	GPIO
GPIO28	JTAG_1_2_SRST_X	GPIO28	DEBUG_28		Pull Up	1.8	GPIO
GPIO29	DBG_I2C_SCL(OD)	GPIO29	DEBUG_29		N.A	1.8	GPIO
GPIO30	DBG_I2C_SDA(OD)	GPIO30	DEBUG_30		N.A	1.8	GPIO
GPIO31	DBG_I2C_SDA_OE	GPIO31	DEBUG_31		Pull Down	1.8	GPIO
BOOT_SEL0	BOOT_SEL0				Pull Down	1.8	Boot mode select0
BOOT_SEL1	BOOT_SEL1				Pull Down	1.8	Boot mode select1
BOOT_SEL2	BOOT_SEL2				Pull Down	1.8	Boot mode select2
BOOT_SEL3	BOOT_SEL3				Pull Down	1.8	Boot mode select3
MODE_SEL0	MODE_SEL0				Pull Down	1.8	Mode Select0
MODE_SEL1	MODE_SEL1				Pull Down	1.8	Mode Select1
MODE_SEL2	MODE_SEL2				Pull Down	1.8	Mode Select2
MODE_SEL3	MODE_SEL3				Pull Down	1.8	Mode Select3
PLL_CLK_IN	PLL_CLK_IN				N.A	1.8	PLL reference clock input
PLL_CLK_IN_DPLL0	PLL_CLK_IN_DPLL0				N.A	1.8	DPLL0 reference clock input
PLL_CLK_IN_DPLL1	PLL_CLK_IN_DPLL1				N.A	1.8	DPLL1 reference clock input
PLL_SRC_SEL	PLL_SRC_SEL				Pull Down	1.8	PLL source reference clock select
SYS_RST_X	SYS_RST_X				Pull Up	1.8	System Reset, active-low
TEST_EN	TEST_EN				Pull Down	1.8	TEST Mode Enable
BISR_BYP	BISR_BYP				Pull Down	1.8	BISR Bypass
P08_PWR_GOOD	P08_PWR_GOOD				Pull Down	1.8	P08 Power Good

OSC_DS3	OSC_DS3				Pull Down	1.8	OSC PAD DS Bit[3]
DDR_PWR_GOOD	DDR_PWR_GOOD				Pull Down	1.8	DDR Power Good (for ISO Control)

6 Part Numbering and Product Version Information

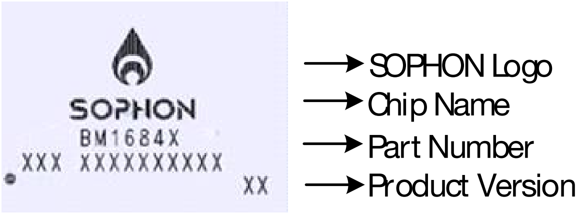


Figure 22 SG2300X Identification

Following figure shows the part order numbering scheme for SG2300X. Refer to SOPHON Field Application Engineers (FAEs) or representatives for further information when ordering parts.

One example is: AAA S1H222AK0G.

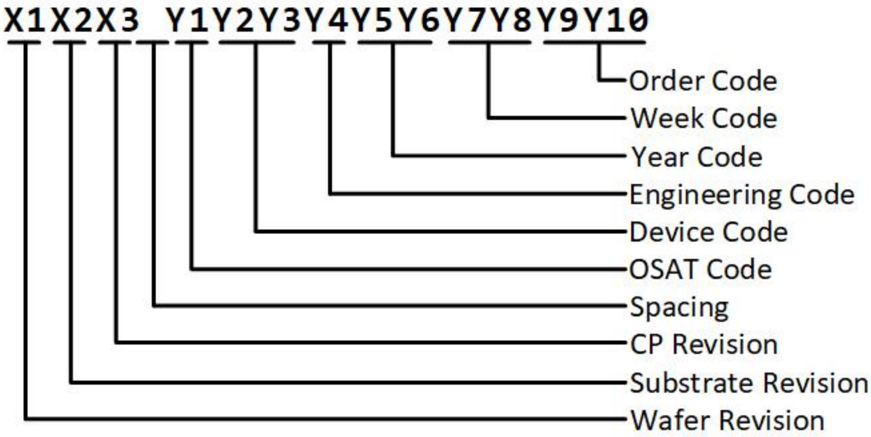


Figure 23 SG2300X Part Number

Table 10 SG2300X Product Version

Product name	**Product version name**	
	Grade	**Version**
SG2300X	S：Standard	0~9
SG2300X	P：Pro	0~9

7 Appendix A: SG2300X eFuse Bit Map

Table 11 SG2300X eFuse Bit Map

Category	Cell Index (32-bit each)	Content	Data Type
top half 1. can NOT be accessed by external 2. cell 0 and 1 does NOT support SRAM read mode, each odd bit and the even bit next to it are backup of each other. 3. secure key is not readable or writable in secure mode	0	secure config bits (and its double bits) bits[1] | [0]: hardware - security master firewall enable bits[3] | [2]: hardware - JTAG debug disable bits[5] | [4]: hardware - on-chip boot bits[7] | [6]: software - secure boot enable	fixed area
	1	Customer Secure Config Bit bit[1] | bit[0]: secure key enable	
	2 - 9	customer secure key (256 bits)	
	10 - 17	backup of customer secure key (256 bits)	
	18 - 25	root public key digest (256 bits)	
	26 - 33	backup of root public key digest (256 bits)	
	34 - 39	ATE info	
	40 - 45	backup of ATE info	
	46	SLT result (1byte flag, 3bytes result)	variable length area software will check the 1st byte of each cell as a flag (0xF?), to find content of interest each content may occur more than once, the one with largest cell index should be considered as the newest and valid one.
	47	backup of SLT result	
	48	product S/N (1byte flag, 3bytes serial number)	
	49	backup of product S/N	
	50	MAC address 1 (1byte flag, 3bytes address)	
	51	backup of MAC address 1	
	52	MAC address 2 (1byte flag, 3bytes address)	
	53	backup of MAC address 2	
	54	customer ID (1byte flag and 3bytes payload in each	

	- 57	cell, ID is 64bit max) This field is usually used by customers' software to identify different batches of products. Software should only execute on specified set of chips.	
	58 - 61	backup of customer ID	
	62		
	63		
bottom half can be accessed by external	64 - 79	MBIST (and its double bits) (512 bits)	fixed area
	80	lane repair (and its double bits)	
	81		
	82	dummy/debug cell for secure key	
	83		variable length area
	84	PCIe patch table region ends here	
	85 - 126		
	127	PCIe single card patch table region starts here 1. from tail to head 2. single card begin with a cell value 0x0b0b0b0b, end with a cell value 0x0e0e0e0e 3. multi card begin with a cell value 0x2b2b2b2b, end with a cell value 0x2e2e2e2e 4. always begin from cell index in odd number	

8 Appendix B: SG2300X Top Register Description

8.1 SG2300X Top Register Summary

Table 12 SG2300X Top Register Summary

Function Group	RegFile	Reg Description	Base Address
AXI Performance Monitor	TOP AXI Fab24 Performance Monitor	Collect transfer performance related to TPU - DDR, PCIe - DDR	0x5000_8000
	TOP AXI Fab25 Performance Monitor	Collect transfer performance related to Video DEC/ENC - DDR	0x5000_8400
	Video system 0 AXI Performance Monitor	Collect transfer performance inside Video Subsystem 0	0x5000_8800
	Video system 1 AXI Performance Monitor	Collect transfer performance inside Video Subsystem 1	0x5000_8C00
Security Related Registers	Security Registers for TOP	Control the security attributes of components inside TOP	0x5000_E000
	Security Registers for PCIe	Control the security attributes of components inside PCIe Subsystem	0x5000_E400
	Security Registers for TPU	Control the security attributes of components inside TPU Subsystem	0x5000_E800
	Security Registers for Video 0	Control the security attributes of components inside Video Subsystem0	0x5000_EC00
	Security Registers for Video 1	Control the security attributes of components inside Video Subsystem1	0x5000_F000
TOP MISC Control and Status	Top MISC Control and Status	Control and Status related to vendor IPs or general purpose	0x5001_0000
IO/Pinmux Control and Status	IO and Pinmux	Control and Status related to BM1684 Chip IO and Pinmux	0x5001_0400
Clock Generation Control and Status	Clock Generation	Control and Status related to Clock Generation	0x5001_0800
Reset Generation Control and	Reset Generation	Control and Status related to Reset Generation	0x5001_0C00

Status			

8.2 AXI Performance Monitor Registers

8.2.1 AXI Performance Monitor Registers Summary

Table 13 AXI Performance Monitor Registers Summary

Offset	Register Name	Default	Attribute
0x000	Period Setting Register	0	RW
0x004	Port Select Register	0	RW
0x008	Reserved for future	0	RO
0x00C	Reserved for future	0	RO
0x010	Peak AW Byte Result Register	0	RO
0x014	Peak WR Byte Result Register	0	RO
0x018	Peak AR Byte Result Register	0	RO
0x01C	Routine AW Byte Sampling Result Register	0	RO
0x020	Routine WR Byte Sampling Result Register	0	RO
0x024	Routine AR Byte Sampling Result Register	0	RO
0x028	Routine AW Byte Trace Result Register	0	RO
0x02C	Routine WR Byte Trace Result Register	0	RO
0x030	Routine AR Byte Trace Result Register	0	RO
0x034	Routine AW Byte Available Trace Results	0	RO
0x038	Routine WR Byte Available Trace Results	0	RO
0x03C	Routine AR Byte Available Trace Results	0	RO
0x040	Peak AW Transaction Result Register	0	RO
0x044	Peak AR Transaction Result Register	0	RO
0x048	Routine AW Transaction Sampling Result Register	0	RO
0x04C	Routine AR Transaction Sampling Result Register	0	RO
0x050	Routine AW Transaction Trace Result Register	0	RO
0x054	Routine AR Transaction Trace Result Register	0	RO
0x058	Routine AW Transaction Available Trace Results	0	RO
0x05C	Routine AR Transaction Available Trace Results	0	RO

8.2.2 AXI Performance Monitor Register Description

8.2.2.1 Period Setting Register (0x000)

Bit	Attribute	Default	Description

31:0	RW	0	Set the threshold of Monitor's Down Counter Once a non-zero is written into this register, Monitor starts counting.

8.2.2.2 Port Select Register (0x004)

Bit	Attribute	Default	Description
31:9	RO	0	Reserved
8	W1C	0	Clear Peak Register Once SW sets this bit to one, Peak AW/WR/AR Result Registers will be cleared. This bit is auto-cleared.
7:0	RW	0	Select port which Monitor will check. This register should be set before Period Setting Register. 0 – Master Port 1 (m1) … N – Slave Port N (sN)

8.2.2.3 Peak AW Byte Result Register (0x010)

Bit	Attribute	Default	Description
31:0	RO	0	Byte Counting Result of Peak Mode AW Channel. After SW writes Period Setting Register, HW will report the Peak Result after each period through this register.

8.2.2.4 Peak WR Byte Result Register (0x014)

Bit	Attribute	Default	Description
31:0	RO	0	Byte Counting Result of Peak Mode WR Channel. After SW writes Period Setting Register, HW will report the Peak Result after each period through this register.

8.2.2.5 Peak AR Byte Result Register (0x018)

Bit	Attribute	Default	Description
31:0	RO	0	Byte Counting Result of Peak Mode AR Channel. After SW writes Period Setting Register, HW will report the Peak Result after each period through this register.

8.2.2.6 Routine AW Byte Sampling Result Register (0x01C)

Bit	Attribute	Default	Description
31:0	RO	0	Byte Counting Result of Routine Mode (Sampling Method) for AW channel. After SW writes Period Setting Register, HW will report the Routine Sampling Result after each period through this register. When SW reads this register, the value will be auto-cleared.

8.2.2.7 Routine WR Byte Sampling Result Register (0x020)

Bit	Attribute	Default	Description
31:0	RO	0	Byte Counting Result of Routine Mode (Sampling Method) for WR channel. After SW writes Period Setting Register, HW will report the Routine Sampling Result after each period through this register. When SW reads this register, the value will be auto-cleared.

8.2.2.8 Routine AR Byte Sampling Result Register (0x024)

Bit	Attribute	Default	Description
31:0	RO	0	Byte Counting Result of Routine Mode (Sampling Method) for AR channel. After SW writes Period Setting Register, HW will report the Routine Sampling Result after each period through this register. When SW reads this register, the value will be auto-cleared.

8.2.2.9 Routine AW Byte Trace Result Register (0x028)

Bit	Attribute	Default	Description
31:0	RO	0	Byte Counting Result of Routine Mode (Trace Method) for AW channel. After SW writes Period Setting Register, HW will report the Routine Trace Result after each period through this register. Actually, this register is the RD port of a FIFO.

8.2.2.10 Routine WR Byte Trace Result Register (0x02C)

Bit	Attribute	Default	Description
31:0	RO	0	Byte Counting Result of Routine Mode (Trace Method) for AW channel. After SW writes Period Setting Register, HW will report the Routine Trace Result after each period through this register. Actually, this register is the RD port of a FIFO.

8.2.2.11 Routine AR Byte Trace Result Register (0x030)

Bit	Attribute	Default	Description
31:0	RO	0	Byte Counting Result of Routine Mode (Trace Method) for AR channel. After SW writes Period Setting Register, HW will report the Routine Trace Result after each period through this register. Actually, this register is the RD port of a FIFO.

8.2.2.12 Routine AW Byte Available Trace Results (0x034)

Bit	Attribute	Default	Description
31:8	RO	0	Reserved
7:0	RO	0	The number of entries now are available in Routine AW Byte Counting Trace Result Register.

8.2.2.13 Routine WR Byte Available Trace Results (0x038)

Bit	Attribute	Default	Description
31:8	RO	0	Reserved
7:0	RO	0	The number of entries now are available in Routine WR Byte Counting Trace Result Register.

8.2.2.14 Routine AR Byte Available Trace Results (0x03C)

Bit	Attribute	Default	Description
31:8	RO	0	Reserved
7:0	RO	0	The number of entries now are available in Routine AR Trace Byte Counting Result Register.

8.2.2.15 Peak AW Transaction Result Register (0x040)

Bit	Attribute	Default	Description
31:0	RO	0	Transaction Counting Result of Peak Mode AW Channel. After SW writes Period Setting Register, HW will report the Peak Result after each period through this register.

8.2.2.16 Peak AR Transaction Result Register (0x044)

Bit	Attribute	Default	Description
31:0	RO	0	Transaction Counting Result of Peak Mode AR Channel. After SW writes Period Setting Register, HW will report the Peak Result after each period through this register.

8.2.2.17 Routine AW Transaction Sampling Result Register (0x048)

Bit	Attribute	Default	Description
31:0	RO	0	Transaction Counting Result of Routine Mode (Sampling Method) for AW channel. After SW writes Period Setting Register, HW will report the Routine Sampling Result after each period through this register. When SW reads this register, the value will be auto-cleared.

8.2.2.18 Routine AR Transaction Sampling Result Register (0x04C)

Bit	Attribute	Default	Description
31:0	RO	0	Transaction Counting Result of Routine Mode (Sampling Method) for AR channel. After SW writes Period Setting Register, HW will report the Routine Sampling Result after each period through this register. When SW reads this register, the value will be auto-cleared.

8.2.2.19 Routine AW Transaction Trace Result Register (0x050)

Bit	Attribute	Default	Description
31:0	RO	0	Transaction Counting Result of Routine Mode (Trace Method) for AW channel. After SW writes Period Setting Register, HW will report the Routine Trace Result after each period through this register. Actually, this register is the RD port of a FIFO.

8.2.2.20 Routine AR Transaction Trace Result Register (0x054)

Bit	Attribute	Default	Description
31:0	RO	0	Transaction Counting Result of Routine Mode (Trace Method) for AR channel. After SW writes Period Setting Register, HW will report the Routine Trace Result after each period through this register. Actually, this register is the RD port of a FIFO.

8.2.2.21 Routine AW Transaction Available Trace Results (0x058)

Bit	Attribute	Default	Description
31:8	RO	0	Reserved
7:0	RO	0	The number of entries now are available in Routine AW Transaction Counting Trace Result Register.

8.2.2.22 Routine AR Transaction Available Trace Results (0x05C)

Bit	Attribute	Default	Description
31:8	RO	0	Reserved
7:0	RO	0	The number of entries now are available in Routine AR Trace Transaction Counting Result Register.

8.3 Security Related Registers

8.3.1 Security Related Registers Summary

Table 14 Security Related Registers Summary

Offset	Register Name	Default	Attribute
0x000	Global Memory Space Secure Flag Control	FFFF_FFFF	RW
0x004	Global Memory Space Forbidden Flag Control	0	RW
0x008	Security Master Firewall Secure Bit Control	0	RW
0x00C	Security Master Firewall Secure Bit Value	0	RW
0x010	Security Master Firewall Interrupt Status	0	RO
0x014	Clear Security Master Firewall Interrupt	0	RW
0x018	Security Debug Firewall Control	0	RW
0x020	Start Address of Global Memory Space 0	0x5000E	RW
0x024	End Address of Global Memory Space 0	0x5000F	RW
0x028	Start Address of Global Memory Space 1	0x7000	RW
0x02C	End Address of Global Memory Space 1	0x70FF	RW
0x030	Start Address of Global Memory Space 2	0x50028	RW
0x034	End Address of Global Memory Space 2	0x50028	RW
0x038 - 0x09C	Start/End Address of Global Memory Space 3~15	0	RW
0x200	Master Firewall 0 Interrupt Information Reg 0	0	RO
0x204	Master Firewall 0 Interrupt Information Reg 1	0	RO
0x208 - 0x2FC	Master Firewall 1-31 Interrupt Information Reg 0/1	0	RO

8.3.2 Security Related Register Description

8.3.2.1 Global Memory Space Secure Flag Control (0x000)

Bit	Attribute	Default	Description
31:16	RO	0	Reserved
15:0	RW	16'h7	Secure Flag Each bit in this field represents the Secure Attribute of one memory space defined by 0x020 ~ 0x0C. Bit[0] controls Memory Space 0 while bit[15] controls Memory Space 15. 0: Memory Space can be reached by both non-secure/secure access. 1: Memory Space can only be reached by secure access.

8.3.2.2 Global Memory Space Forbidden Flag Control (0x004)

Bit	Attribute	Default	Description
31:16	RO	0	Reserved
15:0	RW	0	Forbidden Flag Each bit in this field represents the Attribute of one memory space defined by 0x020 ~ 0x9C. Bit[0] controls Memory Space 0 while bit[15] controls Memory Space 15. 0: Memory Space can be reached via correct Secure or Non-secure access. 1: Memory Space cannot be reached under either Secure or Non-secure access. Note that, once a bit is asserted, it remains high until Chip Reset or Power Down.

8.3.2.3 Security Master Firewall Secure Bit Control (0x08)

Bit	Attribute	Default	Description

31:0	RW	FFFF_FFFC(TOP)
		FFFF_FFFF(others)
			Control the AW/RPROT[1] bit of following masters.
			0: The Secure Bit of Master's transaction is untouched.
			1: The Secure Bit of Master's transaction will be replaced by corresponding bit in Security Master Firewall Secure Bit Value (0x0C)

Bit Map:

	Location			
Bit	**TOP**	**PCIE**	**TPU**	**Video**
0	a53_sys	CDMA_d0	GDMA_d0	WAVE0
1	a53_mem	CDMA_d1	GDMA_d1	N/A
2	EMMC	PCIE_data	GDMA_des	JPEG0
3	SD	MMU_des	TPU_des	VPP
4	SDMA_UART		HAU_axim0	
5	ETH0		HAU_axim1	
6	ETH1		HAU_des	
7	dbg_io		TSDMA_d1	
8	dbg_mem		TSDMA_d0	
9	video_enc		APLITE_d1	
10			APLITE_d0	
11				
12 - 31				

8.3.2.4 Security Master Firewall Secure Bit Value (0x0C)

Bit	Attribute	Default	Description
31:0	RW	FFFF_FFFF	0: New Secure Bit of Master's transaction is 0 (Secure Access). 1: New Secure Bit of Master's transaction is 1 (Non-secure Access). Bit Map is same as "Security Master Firewall Secure Bit Control (0x08)"

8.3.2.5 Security Master Firewall Interrupt Status (0x10)

Bit	Attribute	Default	Description
31:0	RO	0	Interrupt Status of corresponding Masters. The detailed information of each master is recorded in 0x200 ~ 0x2FC Bit Map is same as “Security Master Firewall Secure Bit Control （0x08)”

8.3.2.6 Clear Security Master Firewall Interrupt (0x14)

Bit	Attribute	Default	Description
31:0	RW	0	Write 1 to certain bit will clear the corresponding interrupt status in “Security Master Firewall Interrupt Status （0x010)”. Note that, this bit is not auto-cleared, after SW confirmed target bit in “Security Master Firewall Interrupt Status （0x010)” is cleared, SW also needs to clear this register so as to re-active the interrupt function inside Master Firewall. Bit Map is same as “Security Master Firewall Secure Bit Control （0x08)”

8.3.2.7 Security Debug Firewall Control (0x18)

Bit	Attribute	Default	Description
31:1	RW	0	Reserved
0	RW	0	JTAG Enable JTAG function is disabled only when eFUSE security bit is programmed and this bit is cleared. Only valid in TOP Security Registers.

8.3.2.8 Start Address of Global Memory Space 0 (0x020)

Bit	Attribute	Default	Description
31:23	RO	0	Reserved
22:0	RW	0x5000E	[34:12] of Memory Space Start Address

8.3.2.9 End Address of Global Memory Space 0 (0x024)

Bit	Attribute	Default	Description
31:23	RO	0	Reserved
22:0	RW	0x5000F	[34:12] of Memory Space End Address

8.3.2.10 Start Address of Global Memory Space 1 (0x028)

Bit	Attribute	Default	Description
31:23	RO	0	Reserved
22:0	RW	0x7000	[34:12] of Memory Space Start Address

8.3.2.11 End Address of Global Memory Space 1 (0x02C)

Bit	Attribute	Default	Description
31:23	RO	0	Reserved
22:0	RW	0x70FF	[34:12] of Memory Space End Address

8.3.2.12 Start Address of Global Memory Space 2 (0x030)

Bit	Attribute	Default	Description
31:23	RO	0	Reserved
22:0	RW	0x50028	[34:12] of Memory Space Start Address

8.3.2.13 End Address of Global Memory Space 2 (0x034)

Bit	Attribute	Default	Description
31:23	RO	0	Reserved
22:0	RW	0x50028	[34:12] of Memory Space End Address

8.3.2.14 Start/End Address of Global Memory Space 3~15 (0x038~0x9C)

Bit	Attribute	Default	Description
31:23	RO	0	Reserved
22:0	RW	0	[34:12] of Memory Space Start/End Address Here is an example on using Start/End Address registers: If you want to check 0x0000~0x0FFF, then you should program 0x0 to both START/END register. If you write 0x0000 to START and 0x1000 to END, then address space 0x0000000~0x1000FFF will be checked.

8.3.2.15 Master Firewall 0 Interrupt Information Reg 0 (0x200)

Bit	Attribute	Default	Description
31:0	RO	0	Addr[31:0] of illegal transaction which triggered the interrupt.

8.3.2.16 Master Firewall 0 Interrupt Information Reg 1 (0x204)

Bit	Attribute	Default	Description
31:8	RO	0	Reserved

7:4	RO	0	Burst Length of illegal transaction which triggered the interrupt.
3	RO	0	Direction of illegal transaction which triggered the interrupt. 0: Read 1: Write
2:0	RO	0	Addr[34:32] of illegal transaction which triggered the interrupt.

8.3.2.17 Master Firewall 1-31 Interrupt Information Reg 0/1 （0x208 ~ 0x2FC)

Bit	Attribute	Default	Description
31:0	RO	0	Interrupt Information

8.4 TOP MISC Control and Status Registers

8.4.1 TOP MISC Control and Status Registers Summary

Table 14 TOP MISC Control and Status Registers Summary

Offset	Register Name	Reg Description
0x0	CHIP_VER	Chip Version
0x4	CONFIG_INFO	Conf information
0x8	TOP_CTRL_REG	Top Control Register
0xC	AP_WFI_STS_REG	AP WFI Status Register
0x10	WARM_RST_CTRL	Warm Reset Control and Status
0x14	VMON_TMON_MUX_SEL	VMON/TMON Mux Select
0x18	PMON_CTRL_REG	Process Monitor Control Register
0x1C	PAD_MS_CTRL_REG	PAD MS Control Register
0x20	RVBAR_ADDR_L	ARM boot start address
0x24	RVBAR_ADDR_H	ARM boot start address
0x28	AP_CLK_CTRL	AP clock control
0x2C	AP_DBG_I2C_ID	AP Debug I2C ID
0x30	VD_STS_REG	Video Status Register
0x34	ARM9_ADDR_REMAP_REG	ARM9 Address Remapping Register
0x38	PWR_GOOD_STS	Power Good Status
0x3C	AP_LITE_BASE_ADDR_REG	AP Lite Reset Vector Base Address Register
0x40	DEV_LOCK_REG0	Device Lock Register0
0x44	DEV_LOCK_REG1	Device Lock Register1
0x48	DEV_LOCK_REG2	Device Lock Register2
0x4C	DEV_LOCK_REG3	Device Lock Register3
0x50	DEV_LOCK_REG4	Device Lock Register4
0x54	DEV_LOCK_REG5	Device Lock Register5
0x58	DEV_LOCK_REG6	Device Lock Register6
0x5C	DEV_LOCK_REG7	Device Lock Register7
0x60	JPEG_ADDR_REMAP_REG	JPEG Address Remap Register
0x64	VD_ADDR_REMAP_REG	Video Address Remap Register
0x68	DDR_REFRESH_CTRL_REG	DDR Refresh Control Registers

0x70	OSC0_CTRL_REG	OSC Cell0 Control Register
0x74	OSC1_CTRL_REG	OSC Cell1 Control Register
0x78	OSC2_CTRL_REG	OSC Cell2 Control Register
0x80	GP_REG0	General Purpose register0
0x84	GP_REG1	General Purpose register1
0x88	GP_REG2	General Purpose register2
0x8C	GP_REG3	General Purpose register3
0x90	GP_REG4	General Purpose register4
0x94	GP_REG5	General Purpose register5
0x98	GP_REG6	General Purpose register6
0x9C	GP_REG7	General Purpose register7
0xA0	GP_REG8	General Purpose register8
0xA4	GP_REG9	General Purpose register9
0xA8	GP_REG10	General Purpose register10
0xAC	GP_REG11	General Purpose register11
0xB0	GP_REG12	General Purpose register12
0xB4	GP_REG13	General Purpose register13
0xB8	GP_REG14	General Purpose register14
0xBC	GP_REG15	General Purpose register15
0xC0	PLL_STAT	PLL Status
0xC4	PLL_CLK_EN_REG	PLL Clock Enable Control
0xE8	MPLL_CTL	Main PLL Control
0xEC	TPLL_CTL	TPU PLL control
0xF0	FPLL_CTL	fixed PLL control
0xF4	VPLL_CTL	Video PLL control
0xF8	DPLL0_CTL	DDR PLL 0 control
0xFC	DPLL1_CTL	DDR PLL 1 control
0x100	PMON_CNTR_REG0	Process Monitor Counter Register0
0x104	PMON_CNTR_REG1	Process Monitor Counter Register1
0x108	PMON_CNTR_REG2	Process Monitor Counter Register2
0x10C	PMON_CNTR_REG3	Process Monitor Counter Register3
0x110	PMON_CNTR_REG4	Process Monitor Counter Register4
0x114	PMON_CNTR_REG5	Process Monitor Counter Register5
0x118	PMON_CNTR_REG6	Process Monitor Counter Register6
0x11C	PMON_CNTR_REG7	Process Monitor Counter Register7
0x120	PMON_CNTR_REG8	Process Monitor Counter Register8
0x124	PMON_CNTR_REG9	Process Monitor Counter Register9
0x128	PMON_CNTR_REG10	Process Monitor Counter Register10
0x12C	PMON_CNTR_REG11	Process Monitor Counter Register11
0x130	PMON_CNTR_REG12	Process Monitor Counter Register12
0x134	PMON_CNTR_REG13	Process Monitor Counter Register13
0x138	PMON_CNTR_REG14	Process Monitor Counter Register14
0x13C	PMON_CNTR_REG15	Process Monitor Counter Register15
0x140	PMON_CNTR_REG16	Process Monitor Counter Register16
0x144	PMON_CNTR_REG17	Process Monitor Counter Register17
0x148	PMON_CNTR_REG18	Process Monitor Counter Register18
0x180	NV_CNTR_DIS	None Volatile Counter Disable

0x184	NV_CNTR_LOW	None Volatile Counter Low
0x188	NV_CNTR_HIGH	None Volatile Counter High
0x18C	WDT_RST_STATUS	Watch Dog Reset Status
0x190	GP_REG14_SET	GP_REG14 Set Register
0x194	GP_REG14_CLR	GP_REG14 Clear Register
0x198	GP_REG15_SET	GP_REG15 Set Register
0x19C	GP_REG15_CLR	GP_REG15 Clear Register
0x1C0	ACG_EN_CTRL_REG0	Auto Clock Gating Enable Control Register 0
0x1C4	ACG_EN_CTRL_REG1	Auto Clock Gating Enable Control Register 1
0x1C8	ACG_EN_CTRL_REG2	Auto Clock Gating Enable Control Register 2
0x1CC	ACG_EN_CTRL_REG3	Auto Clock Gating Enable Control Register 3
0x1D0	ACG_EN_CTRL_REG4	Auto Clock Gating Enable Control Register 4
0x1D4	ACG_EN_CTRL_REG5	Auto Clock Gating Enable Control Register 5
0x1E0	GP_REG16	General Purpose register16
0x1E4	GP_REG17	General Purpose register17
0x1E8	GP_REG18	General Purpose register18
0x1EC	GP_REG19	General Purpose register19
0x1F0	GP_REG20	General Purpose register20
0x1F4	GP_REG21	General Purpose register21
0x1F8	GP_REG22	General Purpose register22
0x1FC	GP_REG23	General Purpose register23
0x200	GP_REG24	General Purpose register24
0x204	GP_REG25	General Purpose register25
0x208	GP_REG26	General Purpose register26
0x20C	GP_REG27	General Purpose register27
0x210	GP_REG28	General Purpose register28
0x214	GP_REG29	General Purpose register29
0x218	GP_REG30	General Purpose register30
0x21C	GP_REG31	General Purpose register31

8.4.2 TOP MISC Control and Status Register Description

8.4.2.1 Chip Version (0x0)

Field Name	Bit	Type	Reset Value	Field Description
CHIP_ID	[31:16]	RO	16’h1686	Chip ID
CHIP_VER	[15:0]	RO	16’h0	Chip version

8.4.2.2 Conf information (0x4)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:26]	RO	6'h0	Reserved

REG_DBG_SEL	[25:24]	RW	2'h0	reg_debug_sel register to select which group of debug signal to output 0: debug_dout = 32’h1684 1: debug_dout = pcie_dbg_signal 2: debug_dout = {22’h0, gdma_debug[9:0]} 3: debug_dout = {17’h0, tpu_next_postdiv1, tpu_next_fbdiv}
Reserved	[23:22]	RO	2'h0	Reserved
REG_JTAG_1_2_SEL_EN	[21]	RW	1'h0	reg_jtag_1_2_sel_en register to enable select JTAG_1_2_* pad source from register instead of IO (GPIO15).
REG_JTAG_1_2_SEL	[20]	RW	1'h0	reg_jtag_1_2_sel 0: JTAG1 signals are muxed out for JTAG_1_2_* pads 1: JTAG2 signals are muxed out for JTAG_1_2_* pads
REG_PLL_SRC_SEL1	[19]	RW	1'h0	reg_pll_src_sel1 Reserved in BM1684.
REG_PLL_SRC_SEL0	[18]	RW	1'h0	reg_pll_src_sel0 register to control pll_src_sel0 1: M/T/V/F PLL ref clock comes from CLK GEN IC 0: M/T/V/F PLL ref clock comes from OSC Reserved in BM1684.
REG_PLL_SRC_SEL_EN	[17]	RW	1'h0	reg_pll_src_sel_en register to enable control pll source from register instead of IO. When set to 1’b1, the pll source will be controlled by reg_pll_src_sel1 and reg_pll_src_sel0 instead of pll_src_sel_din1 and pll_src_sel_din0. Reserved in BM1684.
REG_DBG_SEL_DIN_AND	[16]	RW	1'h1	reg_dbg_sel_din_and register to mask dbg_sel_din, this register will do logic and with dbg_sel_din before it drives any logic.
EFUSE_CHECK_DONE	[15]	RO	1'h0	eFUSE Internal Check Done 0: eFUSE is doing Internal Check, it cannot be accessed by SW. 1: eFUSE has finished the Internal Check, free to access.
Reserved	[14:12]	RO	3'h0	Reserved
DBG_SEL_DIN	[11]	RO	1'h0	dbg_sel_din read only register for dbg_sel_din
VOL_SEL	[10]	RO	1'h0	mode_sel3 (voltage_sel) 0: 0.8V 1: 0.7V
PLL_SRC_SEL_DIN	[9]	RO	1'h0	pll_src_sel_din read only register for pll_src_sel
BOOT_SEL	[8:3]	RO	6'h0	boot_sel read only register for boot_sel IO
MODE_SEL	[2:0]	RO	3'h0	mode_sel read only register for mode_sel IO

8.4.2.3 Top Control Register (0x8)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:24]	RO	8'h0	Reserved
JTAG_MFR_ID_FOR_RV	[23:13]	RW	11'h0	jtag_mfr_id for riscv
DDR2_PLL_BYP_MODE	[12]	RW	1'h0	DDR2 PLL Bypass Mode: 0: Use DDR Core Clock 1: Use 1/4 DDR Core Clock

DDR1_PLL_BYP_MODE	[11]	RW	1'h0	DDR1 PLL Bypass Mode: 0: Use DDR Core Clock 1: Use 1/4 DDR Core Clock
DDR0B_PLL_BYP_MODE	[10]	RW	1'h0	DDR0B PLL Bypass Mode: 0: Use DDR Core Clock 1: Use 1/4 DDR Core Clock
DDR0A_PLL_BYP_MODE	[9]	RW	1'h0	DDR0A PLL Bypass Mode: 0: Use DDR Core Clock 1: Use 1/4 DDR Core Clock
DDR2_INTR_EN	[8]	RW	1'h0	DDR2 Interrupt Enable 0: Disable DDR Interrupt to AP 1. Enable DDR Interrupt to AP. AP sets this bit when DDR reset sequence is finished.
DDR1_INTR_EN	[7]	RW	1'h0	DDR1 Interrupt Enable 0: Disable DDR Interrupt to AP 1. Enable DDR Interrupt to AP. AP sets this bit when DDR reset sequence is finished.
DDR0B_INTR_EN	[6]	RW	1'h0	DDR0B Interrupt Enable 0: Disable DDR Interrupt to AP 1. Enable DDR Interrupt to AP. AP sets this bit when DDR reset sequence is finished.
DDR0A_INTR_EN	[5]	RW	1'h0	DDR0A Interrupt Enable 0: Disable DDR Interrupt to AP 1. Enable DDR Interrupt to AP. AP sets this bit when DDR reset sequence is finished.
DDR_INTLV_MODE	[4]	RW	1'h1	DDR 4KB Interleave Mode 0: Twin Mode. Only DDR0A and DDR0B are involved in interleave. DDR1 and DDR2 is not included. 1: Quadruplet Mode. The interleave operation will involve all DDR0A/DDR0B/DDR1/DDR2.
DDR_INTLV_EN	[3]	RW	1'h1	DDR 4KB Interleave Enable 1: Enable; 0: Disable
SW_ROOT_RESET_EN	[2]	RW	1'h0	SW Root Reset Enable 1: SW Reset is enabled (watchdog, AP Warm Reset, RISCV Warm Reset)
ITCM_AXI_EN	[1]	RW	1'h0	ITCM AXI Enable
ROM_BOOT_FINISH	[0]	RW	1'h0	AHB ROM Boot Finish Flag SW sets this bit to 1 after ROM Boot Finish, and ROM will be put into sleep mode.

8.4.2.4 AP WFI Status Register (0xC)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:27]	RO	5'h0	Reserved
CCI_SEL_WFI	[26]	RW	1'h0	CCI SEL WFI 0: Use AxValid as clock request of CCI 1: Use WFIL2 as clock request of CCI Reserved in BM1684.
CCI_INACT_CYCLE	[25:18]	RW	8'h8	CCI In-Active Cycle Clock will be gated after number of “CCI In-Active Cycle”. Reserved in BM1684.
AP_CLST1_ACINACT	[17]	RW	1'h0	AP system Cluster 1 ACINACTM: 0: cluster 1 may be snooped by external system 1: cluster 1 will not be snooped by external system
AP_CLST0_ACINACT	[16]	RW	1'h0	AP system Cluster 0 ACINACTM 0: cluster 0 may be snooped by external system 1: cluster 0 will not be snooped by external system

Reserved	[15:11]	RW	5'h0	Reserved
AP_LITE_WFI_ST	[10]	RO	1'h0	AP Lite WFI State
AP_CLST1_WFI_ST	[9]	RO	1'h0	AP system Cluster 1 WFI State
AP_CORE4TO7_WFI_ST	[8:5]	RO	4'h0	AP system Core4-7 WFI State
AP_CLST0_WFI_ST	[4]	RO	1'h0	AP system Cluster 0 WFI State
AP_CORE0TO3_WFI_ST	[3:0]	RO	4'h0	AP system Core0-3 WFI State

8.4.2.5 Warm Reset Control and Status (0x10)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:18]	RO	14'h0	Reserved
RV_WARM_RST_ACT	[17]	RO	1'h0	RISCV Warm Reset Active signal. This bit reflect the current status of RISCV Warm Reset Active signal (riscv_debug_ndreset_act).
CLR_RV_WARM_RST_ACT	[16]	RW	1'h0	Clear RISCV Warm Reset Active signal. Writing 1 into this bit will clear the RISCV Warm Reset Active signal (riscv_debug_ndreset_act)
Reserved	[15:3]	RO	13'h0	Reserved
JTAG_WARM_RST_DIS	[2]	RW	1'h0	JTAG Warm Reset Disable 1: Disable Warm Request from JTAG SRST Pin
AP_WARM_RST_ACT	[1]	RO	1'h0	AP System Warm Reset Active signal. This bit reflect the current status of AP System Warm Reset Active signal (ap_sys_warm_rst_act).
CLR_AP_WRAM_RST_ACT	[0]	RW	1'h0	Clear AP System Warm Reset Active signal. Writing 1 into this bit will clear the AP System Warm Reset Active signal (ap_sys_warm_rst_act)

8.4.2.6 VMON/TMON Mux Select (0x14)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:16]	RO	16'h0	Reserved
Reserved	[15:13]	RO	3'h0	Reserved
VOL_MUX_SEL	[12:8]	RW	5'h0	Voltage Mux Select
Reserved	[7:5]	RO	3'h0	Reserved
TEMP_MUX_SEL	[4:0]	RW	5'h0	Temperature Mux Select

8.4.2.7 Process Monitor Control Register (0x18)

Field Name	Bit	Type	Reset Value	Field Description
PMON_CNT	[31:16]	RO	16'h0	toreg_pm_count

				count value of process monitor
Reserved	[15:4]	RO	12'h0	Reserved
PMON_EN	[3]	RW	1'h0	reg_pm_en Enable signal for process monitor clock
PMON_SEL	[2:1]	RW	2'h0	reg_pm_select Selection of process monitor
PMON_START	[0]	RW	1'h0	reg_pm_start Start trigger of process monitor Sequence: Step1: Set reg_pm_en Step2: Configure reg_pm_select Step3: Set reg_pm_start Step4: Read toreg_pm_count

8.4.2.8 PAD MS Control Register (0x1C)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:4]	RO	28'h0	Reserved
MS_GRP	[3:0]	RW	4'h4	[3]: MS group 3 (0: 3V mode; 1: 1.8V mode) [2]: MS group 2 (0: 3V mode; 1: 1.8V mode) [1]: MS group 1 (0: 3V mode; 1: 1.8V mode) [0]: MS group 0 (0: 3V mode; 1: 1.8V mode)

8.4.2.9 ARM boot start address (0x20)

Field Name	Bit	Type	Reset Value	Field Description
AP_RVBAR_L	[31:0]	RO	32'h0	ap_rvbaraddr[31:0] ARM boot start address. After Power-On Reset, ARM boot start address can be selected from: (1). When SPI boot selected, ap_rvbaraddr = 40’h00_0600_0000 (2). When ROM boot selected, ap_rvbaraddr = 40’h00_0700_0000 SW is able to configure this boot address register during runtime. And by doing this, once ARM Warm Reset is triggered, ARM will boot from the new configured start address.

8.4.2.10 ARM boot start address (0x24)

Field Name	Bit	Type	Reset Value	Field Description
AP_RVBAR_H	[31:0]	RO	32'h0	ap_rvbaraddr[39:32] ARM boot start address. After Power-On Reset, ARM boot start address can be selected from: (1). When SPI boot selected, ap_rvbaraddr = 40’h00_0600_0000 (2). When ROM boot selected, ap_rvbaraddr = 40’h00_0700_0000 SW is able to configure this boot address register during runtime. And by doing this, once ARM Warm Reset is

				triggered, ARM will boot from the new configured start address.

8.4.2.11 AP clock control (0x28)

Field Name	Bit	Typ e	Reset Value	Field Description
Reserved	[31:5]	RO	27'h0	Reserved
ARM_CLK_UP_DONE	[4]	RO	1'h1	reg_arm_clk_up_done done status of updating arm clock divider when this bit is 1'b1, software can trigger another configuration of arm clock divider.
ARM_CLK_DIV_UP	[3]	RW	1'h0	reg_arm_clk_div_up update trigger of arm clock divider
ARM_CLK_HWIDE	[2]	RW	1'h0	reg_arm_clk_highwide highwide for arm clock divider
ARM_CLK_DIV	[1:0]	RW	2'h0	reg_arm_clk_div divider for arm clock

8.4.2.12 AP Debug I2C ID (0x2C)

Field Name	Bit	Typ e	Reset Value	Field Description
Reserved	[31:8]	RO	24'h0	Reserved
AP_DBG_I2C_ID	[7:0]	RW	8'h0	AP System Debug I2C ID

8.4.2.13 Video Status Register (0x30)

Field Name	Bit	Typ e	Reset Value	Field Description
Reserved	[31:5]	RO	27'h0	Reserved
VDE_IDLE	[4]	RO	1'h1	Video Encoder Subsystem wave521_idle 0: wave 521 is not idle 1: wave 521 is idle
VD1_WAVE1_IDLE	[3]	RO	1'h1	Video System 1 wave1_idle 0: wave 1 is not idle 1: wave 1 is idle
VD1_WAVE0_IDLE	[2]	RO	1'h1	Video System 1 wave0_idle 0: wave 0 is not idle 1: wave 0 is idle
VD0_WAVE1_IDLE	[1]	RO	1'h1	Video System 0 wave1_idle 0: wave 1 is not idle 1: wave 1 is idle
VD0_WAVE0_IDLE	[0]	RO	1'h1	Video System 0 wave0_idle 0: wave 0 is not idle 1: wave 0 is idle

8.4.2.14 ARM9 Address Remapping Register (0x34)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:4]	RO	28'h0	Reserved
ARM9_ADDR_REMAP	[3:0]	RW	4'h2	ARM9 Address Remapping Register. Remap Scheme: new[34:0] = ori[31] ? {remap, ori[30:0]} : {4'h0: ori[30:0]} Note: by default, when ARM9 r/w 0x8000_0000, the address should be remapped to 0x1_0000_0000 in DDR0.

8.4.2.15 Power Good Status (0x38)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:20]	RO	12'h0	Reserved
DDR_PWR_GOOD_VAL	[19]	RO	1'h0	DDR Power Good Value after mux between IO value and register value.
DDR_PWR_GOOD_VAL_IO	[18]	RO	1'h0	DDR Power Good Value from IO
DDR_PWR_GOOD_VAL_REG	[17]	RW	1'h0	DDR Power Good Status Register Value When "DDR Power Good Status Select" is 1, this bit will be selected as value for ISO cell.
DDR_PWR_GOOD_SEL	[16]	RW	1'h0	DDR Power Good Status Select: 0: Select IO (DDR_PWR_GOOD) as value for ISO cell. 1: Select register (bit[17] of this register) as value for ISO cell.
PCIE_PWR_GOOD_VAL	[15]	RO	1'h0	PCIE Power Good Value after mux between IO value and register value.
PCIE_PWR_GOOD_VAL_IO	[14]	RO	1'h0	PCIE Power Good Value from IO
PCIE_PWR_GOOD_VAL_REG	[13]	RW	1'h0	PCIE Power Good Status Register Value When "PCIE Power Good Status Select" is 1, this bit will be selected as value for ISO cell.
PCIE_PWR_GOOD_SEL	[12]	RW	1'h0	PCIE Power Good Status Select: 0: Select IO (PCIE_PWR_GOOD) as value for ISO cell. 1: Select register (bit[13] of this register) as value for ISO cell.
TPU_PWR_GOOD_VAL	[11]	RO	1'h0	TPU Power Good Value after mux between IO value and register value.
TPU_PWR_GOOD_VAL_IO	[10]	RO	1'h0	TPU Power Good Value from IO
TPU_PWR_GOOD_VAL_REG	[9]	RW	1'h0	TPU Power Good Status Register Value When "TPU Power Good Status Select" is 1, this bit will be selected as value for ISO cell.
TPU_PWR_GOOD_SEL	[8]	RW	1'h0	TPU Power Good Status Select: 0: Select IO (TPU_PWR_GOOD) as value for ISO cell. 1: Select register (bit[9] of this register) as value for ISO cell.
TPUMEM_PWR_GOOD_VAL	[7]	RO	1'h0	TPU_MEM Power Good Value after mux between IO value and register value.
TPUMEM_PWR_GOOD_VAL_IO	[6]	RO	1'h0	TPU_MEM Power Good Value from IO
TPUMEM_PWR_GOOD	[5]	RW	1'h0	TPU_MEM Power Good Status Register Value When "TPU_MEM Power Good Status Select" is 1, this bit

_VAL_REG				will be selected as value for ISO cell.
TPUMEM_PWR_GOOD_SEL	[4]	RW	1'h0	TPU_MEM Power Good Status Select: 0: Select IO (TPU_MEM_PWR_GOOD) as value for ISO cell. 1: Select register (bit[5] of this register) as value for ISO cell.
P08_PWR_GOOD_VAL	[3]	RO	1'h0	P08 Power Good Value after mux between IO value and register value.
P08_PWR_GOOD_VAL_IO	[2]	RO	1'h0	P08 Power Good Value from IO
P08_PWR_GOOD_VAL_REG	[1]	RW	1'h0	P08 Power Good Status Register Value When "P08 Power Good Status Select" is 1, this bit will be selected as value for ISO cell.
P08_PWR_GOOD_SEL	[0]	RW	1'h0	P08 Power Good Status Select: 0: Select IO (P08_PWR_GOOD) as value for ISO cell. 1: Select register (bit[1] of this register) as value for ISO cell.

8.4.2.16 AP Lite Reset Vector Base Address Register (0x3C)

Field Name	Bit	Type	Reset Value	Field Description
AP_LITE_BASE_ADDR	[31:0]	RW	32'h8100 0000	reg_ap_lite_bar AP Lite Base Address = {reg_ap_lite_bar, 4'h0}

8.4.2.17 Device Lock Register0 (0x40)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:1]	RO	31'h0	Reserved
LOCK_CTRL_STS_0	[0]	RW	1'h0	Lock Control and Status (1). The read operation will return the value then assert this bit. (2). Write operation will de-assert the bit.

8.4.2.18 Device Lock Register1 (0x44)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:1]	RO	31'h0	Reserved
LOCK_CTRL_STS_1	[0]	RW	1'h0	Lock Control and Status (1). The read operation will return the value then assert this bit. (2). Write operation will de-assert the bit.

8.4.2.19 Device Lock Register2 (0x48)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:1]	RO	31'h0	Reserved
LOCK_CTRL_STS_2	[0]	RW	1'h0	Lock Control and Status (1). The read operation will return the value then assert this

				bit. (2). Write operation will de-assert the bit.

8.4.2.20 Device Lock Register3 (0x4C)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:1]	RO	31'h0	Reserved
LOCK_CTRL_STS_3	[0]	RW	1'h0	Lock Control and Status (1). The read operation will return the value then assert this bit. (2). Write operation will de-assert the bit.

8.4.2.21 Device Lock Register4 (0x50)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:1]	RO	31'h0	Reserved
LOCK_CTRL_STS_4	[0]	RW	1'h0	Lock Control and Status (1). The read operation will return the value then assert this bit. (2). Write operation will de-assert the bit.

8.4.2.22 Device Lock Register5 (0x54)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:1]	RO	31'h0	Reserved
LOCK_CTRL_STS_5	[0]	RW	1'h0	Lock Control and Status (1). The read operation will return the value then assert this bit. (2). Write operation will de-assert the bit.

8.4.2.23 Device Lock Register6 (0x58)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:1]	RO	31'h0	Reserved
LOCK_CTRL_STS_6	[0]	RW	1'h0	Lock Control and Status (1). The read operation will return the value then assert this bit. (2). Write operation will de-assert the bit.

8.4.2.24 Device Lock Register7 (0x5C)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:1]	RO	31'h0	Reserved

LOCK_CTRL_STS_7	[0]	RW	1'h0	Lock Control and Status (1). The read operation will return the value then assert this bit. (2). Write operation will de-assert the bit.

8.4.2.25 JPEG Address Remap Register (0x60)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31]	RO	1'h0	Reserved
VD1_JPEG1_RD_ADDR_REMAP	[30:28]	RW	3'h3	Video Subsystem1 JPEG1 Read Channel Address Remap Control
Reserved	[27]	RO	1'h0	Reserved
VD1_JPEG1_WR_ADDR_REMAP	[26:24]	RW	3'h3	Video Subsystem1 JPEG1 Write Channel Address Remap Control
Reserved	[23]	RO	1'h0	Reserved
VD1_JPEG0_RD_ADDR_REMAP	[22:20]	RW	3'h3	Video Subsystem1 JPEG0 Read Channel Address Remap Control
Reserved	[19]	RO	1'h0	Reserved
VD1_JPEG0_WR_ADDR_REMAP	[18:16]	RO	3'h3	Video Subsystem1 JPEG0 Write Channel Address Remap Control
Reserved	[15]	RW	1'h0	Reserved
VD0_JPEG1_RD_ADDR_REMAP	[14:12]	RO	3'h3	Video Subsystem0 JPEG1 Read Channel Address Remap Control
Reserved	[11]	RW	1'h0	Reserved
VD0_JPEG1_WR_ADDR_REMAP	[10:8]	RO	3'h3	Video Subsystem0 JPEG1 Write Channel Address Remap Control
Reserved	[7]	RW	1'h0	Reserved
VD0_JPEG0_RD_ADDR_REMAP	[6:4]	RO	3'h3	Video Subsystem0 JPEG0 Read Channel Address Remap Control
Reserved	[3]	RO	1'h0	Reserved
VD0_JPEG0_WR_ADDR_REMAP	[2:0]	RW	3'h3	Video Subsystem0 JPEG0 Write Channel Address Remap Control

8.4.2.26 Video Address Remap Register (0x64)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:27]	RO	5'h0	Reserved
VDE_REMAP_34TO32	[26:24]	RW	3'h1	reg_vde_remap_addr_34to32 register for address 34to32 of video encoder subsystem
VD1_REMAP_39to32	[23:16]	RW	8'h1	reg_vd1_remap_addr_39to32 register for address 39to32 of video system 1
Reserved	[15:8]	RO	8'h0	Reserved
VD0_REMAP_39to32	[7:0]	RW	8'h1	reg_vd0_remap_addr_39to32 register for address 39to32 of video system 0

8.4.2.27 DDR Refresh Control Registers (0x68)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:8]	RO	24'h0	Reserved
DDR2_MST_MODE	[7]	RW	1'h1	DDR2 Master Mode 1: Master Mode 0: Slave Mode
DDR2_REFRESH_MODE_BYP	[6]	RW	1'h1	DDR2 Refresh Mode Bypass
DDR1_MST_MODE	[5]	RW	1'h1	DDR1 Master Mode 1: Master Mode 0: Slave Mode
DDR1_REFRESH_MODE_BYP	[4]	RW	1'h1	DDR1 Refresh Mode Bypass
DDR0B_MST_MODE	[3]	RW	1'h1	DDR0 B Master Mode 1: Master Mode 0: Slave Mode
DDR0B_REFRESH_MODE_BYP	[2]	RW	1'h1	DDR0 B Refresh Mode Bypass
DDR0A_MST_MODE	[1]	RW	1'h1	DDR0 A Master Mode 1: Master Mode 0: Slave Mode
DDR0A_REFRESH_MODE_BYP	[0]	RW	1'h1	DDR0 A Refresh Mode Bypass

8.4.2.28 OSC Cell0 Control Register (0x70)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:9]	RO	23'h0	Reserved
OSC0_RD_CTRL	[8:7]	RW	2'h0	Damping Resistance (RD) control via RD[1:0]
OSC0_RF_CTRL	[6:5]	RW	2'h2	Feedback Resistance (RF) control via REF[1:0].
OSC0_DS3	[4]	RO	1'h1	Driving Strength(DS) Control – DS[3], this field is controlled by PAD OSC_DS3
OSC0_DS2TO0	[3:1]	RW	3'h7	Driving Strength(DS) Control – DS[2:0], this field is controlled by SW.
Reserved	[0]	RW	1'h0	Reserved

8.4.2.29 OSC Cell1 Control Register (0x74)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:9]	RO	23'h0	Reserved
OSC1_RD_	[8:7]	RW	2'h0	Damping Resistance (RD) control via RD[1:0]

CTRL				
OSC1_RF_CTRL	[6:5]	RW	2'h2	Feedback Resistance (RF) control via REF[1:0].
OSC1_DS3	[4]	RO	1'h1	Driving Strength(DS) Control – DS[3], this field is controlled by PAD OSC_DS3
OSC1_DS2TO0	[3:1]	RW	3'h7	Driving Strength(DS) Control – DS[2:0], this field is controlled by SW.
OSC1_XE	[0]	RW	1'h1	XE Control for Test Mode Scenario selection.

8.4.2.30 OSC Cell2 Control Register (0x78)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:9]	RO	23'h0	Reserved
OSC2_RD_CTRL	[8:7]	RW	2'h0	Damping Resistance (RD) control via RD[1:0]
OSC2_RF_CTRL	[6:5]	RW	2'h2	Feedback Resistance (RF) control via REF[1:0].
OSC2_DS3	[4]	RO	1'h1	Driving Strength(DS) Control – DS[3], this field is controlled by PAD OSC_DS3
OSC2_DS2TO0	[3:1]	RW	3'h7	Driving Strength(DS) Control – DS[2:0], this field is controlled by SW.
OSC2_XE	[0]	RW	1'h1	XE Control for Test Mode Scenario selection.

8.4.2.31 General Purpose register0 (0x80)

Field Name	Bit	Type	Reset Value	Field Description
GP_REG0	[31:0]	RW	32'h0	General Purpose register for SW usage.

8.4.2.32 General Purpose register1 (0x84)

Field Name	Bit	Type	Reset Value	Field Description
GP_REG1	[31:0]	RW	32'h0	General Purpose register for SW usage.

8.4.2.33 General Purpose register2 (0x88)

Field Name	Bit	Type	Reset Value	Field Description
GP_REG2	[31:0]	RW	32'h0	General Purpose register for SW usage.

8.4.2.34 General Purpose register3 (0x8C)

Field Name	Bit	Type	Reset Value	Field Description

GP_REG3	[31:0]	RW	32'h0	General Purpose register for SW usage.

8.4.2.35 General Purpose register4 (0x90)

Field Name	Bit	Typ e	Reset Value	Field Description
GP_REG4	[31:0]	RW	32'h0	General Purpose register for SW usage.

8.4.2.36 General Purpose register5 (0x94)

Field Name	Bit	Typ e	Reset Value	Field Description
GP_REG5	[31:0]	RW	32'h0	General Purpose register for SW usage.

8.4.2.37 General Purpose register6 (0x98)

Field Name	Bit	Typ e	Reset Value	Field Description
GP_REG6	[31:0]	RW	32'h0	General Purpose register for SW usage.

8.4.2.38 General Purpose register7 (0x9C)

Field Name	Bit	Typ e	Reset Value	Field Description
GP_REG7	[31:0]	RW	32'h0	General Purpose register for SW usage.

8.4.2.39 General Purpose register8 (0xA0)

Field Name	Bit	Typ e	Reset Value	Field Description
GP_REG8	[31:0]	RW	32'h0	General Purpose register for SW usage.

8.4.2.40 General Purpose register9 (0xA4)

Field Name	Bit	Typ e	Reset Value	Field Description
GP_REG9	[31:0]	RW	32'h0	General Purpose register for SW usage.

8.4.2.41 General Purpose register10 (0xA8)

Field Name	Bit	Typ e	Reset Value	Field Description

GP_REG10	[31:0]	RW	32'h0	General Purpose register for SW usage.

8.4.2.42 General Purpose register11 (0xAC)

Field Name	Bit	Type	Reset Value	Field Description
GP_REG11	[31:0]	RW	32'h0	General Purpose register for SW usage.

8.4.2.43 General Purpose register12 (0xB0)

Field Name	Bit	Type	Reset Value	Field Description
GP_REG12	[31:0]	RW	32'h0	General Purpose register for SW usage.

8.4.2.44 General Purpose register13 (0xB4)

Field Name	Bit	Type	Reset Value	Field Description
GP_REG13	[31:0]	RW	32'h0	General Purpose register for SW usage.

8.4.2.45 General Purpose register14 (0xB8)

Field Name	Bit	Type	Reset Value	Field Description
GP_REG14	[31:0]	RW	32'h0	General Purpose register for SW usage bit[31:16] are used to interrupt RISC-V. bit[15:0] are used to interrupt ARM926.

8.4.2.46 General Purpose register15 (0xBC)

Field Name	Bit	Type	Reset Value	Field Description
GP_REG15	[31:0]	RW	32'h0	General Purpose register for SW usage. bit[15:0] are used to interrupt A53.

8.4.2.47 PLL Status (0xC0)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:14]	RO	18'h0	Reserved
DPLL1_LOCK	[13]	RO	1'h0	DPLL1 LOCK
DPLL0_LOCK	[12]	RO	1'h0	DPLL0 LOCK

FPLL_LOCK	[11]	RO	1'h0	FPLL LCOK
VPLL_LOCK	[10]	RO	1'h0	VPLL LCOK
TPLL_LOCK	[9]	RO	1'h0	TPLL LOCK
MPLL_LOCK	[8]	RO	1'h0	MPLL LOCK
Reserved	[7:6]	RO	2'h0	Reserved
UPDT_DPLL1	[5]	RO	1'h0	updating_dpll1_val
UPDT_DPLL0	[4]	RO	1'h0	updating_dpll0_val
UPDT_FPLL	[3]	RO	1'h0	updating_fpll_val
UPDT_VPLL	[2]	RO	1'h0	updating_vpll_val
UPDT_TPLL	[1]	RO	1'h0	updating_tpll_val
UPDT_MPLL	[0]	RO	1'h0	updating_mpll_val

8.4.2.48 PLL Clock Enable Control (0xC4)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:17]	RO	15'h0	Reserved
TPLL_CONFIG_SEL	[16]	RO	1'h0	TPLL Configuration Selection: 0: Select TPLL configuration calculated by TOP 1: Select TPLL configuration calculated by TPU
Reserved	[15:14]	RO	2'h0	Reserved
DPLL1_CLK_EN_MUX	[13]	RW	1'h0	DPLL1 Clock Enable Mux Control 0: Select Unsynced PLL Clock Enable 1: Select Synced version of PLL Clock Enable
DPLL0_CLK_EN_MUX	[12]	RW	1'h0	DPLL0 Clock Enable Mux Control 0: Select Unsynced PLL Clock Enable 1: Select Synced version of PLL Clock Enable
FPLL_CLK_EN_MUX	[11]	RW	1'h0	FPLL Clock Enable Mux Control 0: Select Unsynced PLL Clock Enable 1: Select Synced version of PLL Clock Enable
VPLL_CLK_EN_MUX	[10]	RW	1'h0	VPLL Clock Enable Mux Control 0: Select Unsynced PLL Clock Enable 1: Select Synced version of PLL Clock Enable
TPLL_CLK_EN_MUX	[9]	RW	1'h0	TPLL Clock Enable Mux Control 0: Select Unsynced PLL Clock Enable 1: Select Synced version of PLL Clock Enable
MPLL_CLK_EN_MUX	[8]	RW	1'h0	MPLL Clock Enable Mux Control 0: Select Unsynced PLL Clock Enable 1: Select Synced version of PLL Clock Enable
Reserved	[7:6]	RO	2'h0	Reserved
DPLL1_CLK_EN	[5]	RW	1'h1	DPLL1 Clock Enable
DPLL0_CLK_EN	[4]	RW	1'h1	DPLL0 Clock Enable
FPLL_CLK_EN	[3]	RW	1'h1	FPLL Clock Enable
VPLL_CLK_EN	[2]	RW	1'h1	VPLL Clock Enable
TPLL_CLK_EN	[1]	RW	1'h1	TPLL Clock Enable
MPLL_CLK_EN	[0]	RW	1'h1	MPLL Clock Enable

8.4.2.49 Main PLL Control (0xE8)

Field Name	Bit	Type	Reset Value	Field Description
MPLL_FAST_CONFIG_EN	[31]	WO	1'h0	Fast Config Mode Enable 1: Enable Fast Config Mode. In this mode, only FBDIV can be modified, and there will be no PLL Power-Down sequence in PLL frequency update. 0: Disable Fast Config Mode.
Reserved	[30:28]	RO	3'h0	Reserved
MPLL_FBDIV	[27:16]	RW	12'h92	FBDIV. NOTE:Default value when BOOT_SEL3 == 0 (0.8v): Normal Mode: 146 Fast Mode: 92 Safe Mode: 92 BOOT_SEL3 == 1 (0.7v): Normal Mode: 46 Fast Mode: 69 Safe Mode: 69
Reserved	[15]	RO	1'h0	Reserved
MPLL_POSTDIV2	[14:12]	RW	3'h1	POSTDIV2. NOTE:Default value when BOOT_SEL3 == 0 (0.8v): Normal Mode: 1 Fast Mode: 1 Safe Mode: 1 BOOT_SEL3 == 1 (0.7v): Normal Mode: 1 Fast Mode: 1 Safe Mode: 1
Reserved	[11]	RO	1'h0	Reserved
MPLL_POSTDIV1	[10:8]	RW	3'h1	POSTDIV1. NOTE:Default value when BOOT_SEL3 == 0 (0.8v): Normal Mode: 1 Fast Mode: 1 Safe Mode: 1 BOOT_SEL3 == 1 (0.7v): Normal Mode: 1 Fast Mode: 1 Safe Mode: 1
Reserved	[7:5]	RO	3'h0	Reserved
MPLL_REFDIV	[4:0]	RW	5'h2	REFDIV. NOTE:Default value when BOOT_SEL3 == 0 (0.8v): Normal Mode: 2 Fast Mode: 1 Safe Mode: 2 BOOT_SEL3 == 1 (0.7v): Normal Mode: 1 Fast Mode: 1 Safe Mode: 2

8.4.2.50 TPU PLL control (0xEC)

Field Name	Bit	Type	Reset Value	Field Description
TPLL_FAST_CONFIG_E	[31]	WO	1'h0	Fast Config Mode Enable 1: Enable Fast Config Mode. In this mode, only FBDIV can be

N				modified, and there will be no PLL Power-Down sequence in PLL frequency update. 0: Disable Fast Config Mode.
Reserved	[30:28]	RO	3'h0	Reserved
TPLL_FBDIV	[27:16]	RW	12'h3C	FBDIV. NOTE:Default value when BOOT_SEL3 == 0 (0.8v): Normal Mode: 60 Fast Mode: 40 Safe Mode: 40 BOOT_SEL3 == 1 (0.7v): Normal Mode: 60 Fast Mode: 40 Safe Mode: 40
Reserved	[15]	RO	1'h0	Reserved
TPLL_POSTDIV2	[14:12]	RW	3'h1	POSTDIV2. NOTE:Default value when BOOT_SEL3 == 0 (0.8v): Normal Mode: 1 Fast Mode: 1 Safe Mode: 1 BOOT_SEL3 == 1 (0.7v): Normal Mode: 1 Fast Mode: 1 Safe Mode: 1
Reserved	[11]	RO	1'h0	Reserved
TPLL_POSTDIV1	[10:8]	RW	3'h5	POSTDIV1. NOTE:Default value when BOOT_SEL3 == 0 (0.8v): Normal Mode: 2 Fast Mode: 1 Safe Mode: 2 BOOT_SEL3 == 1 (0.7v): Normal Mode: 2 Fast Mode: 1 Safe Mode: 2
Reserved	[7:5]	RO	3'h0	Reserved
TPLL_REFDIV	[4:0]	RW	5'h1	REFDIV. NOTE:Default value when BOOT_SEL3 == 0 (0.8v): Normal Mode: 1 Fast Mode: 1 Safe Mode: 1 BOOT_SEL3 == 1 (0.7v): Normal Mode: 1 Fast Mode: 1 Safe Mode: 1

8.4.2.51 fixed PLL control (0xF0)

Field Name	Bit	Type	Reset Value	Field Description
FPLL_FAST_CONFIG_EN	[31]	WO	1'h0	Fast Config Mode Enable 1: Enable Fast Config Mode. In this mode, only FBDIV can be modified, and there will be no PLL Power-Down sequence in PLL frequency update. 0: Disable Fast Config Mode.
Reserved	[30:28]	RO	3'h0	Reserved
FPLL_FBDIV	[27:16]	RW	12'h50	FBDIV. NOTE:Default value whenNormal Mode: 80 Fast Mode: 160 Safe Mode: 80
Reserved	[15]	RO	1'h0	Reserved
FPLL_POST	[14:12]	RW	3'h1	POSTDIV2.

DIV2				
Reserved	[11]	RO	1'h0	Reserved
FPLL_POST DIV1	[10:8]	RW	3'h1	POSTDIV1.
Reserved	[7:5]	RO	3'h0	Reserved
FPLL_REFD IV	[4:0]	RW	5'h2	REFDIV.

8.4.2.52 Video PLL control (0xF4)

Field Name	Bit	Typ e	Reset Value	Field Description
VPLL_FAST _CONFIG_E N	[31]	WO	1'h0	Fast Config Mode Enable 1: Enable Fast Config Mode. In this mode, only FBDIV can be modified, and there will be no PLL Power-Down sequence in PLL frequency update. 0: Disable Fast Config Mode.
Reserved	[30:28]	RO	3'h0	Reserved
VPLL_FBDI V	[27:16]	RW	12'h80	FBDIV. NOTE:Default value BOOT_SEL3 == 0 (0.8v): Normal Mode: 128 Fast Mode: 64 Safe Mode: 32 BOOT_SEL3 == 1 (0.7v): Normal Mode: 32 Fast Mode: 48 Safe Mode: 36
Reserved	[15]	RO	1'h0	Reserved
VPLL_POST DIV2	[14:12]	RW	3'h1	POSTDIV2. NOTE:Default value when BOOT_SEL3 == 0 (0.8v): Normal Mode: 1 Fast Mode: 1 Safe Mode: 1 BOOT_SEL3 == 1 (0.7v): Normal Mode: 1 Fast Mode: 1 Safe Mode: 1
Reserved	[11]	RO	1'h0	Reserved
VPLL_POST DIV1	[10:8]	RW	3'h5	POSTDIV1. NOTE:Default value when BOOT_SEL3 == 0 (0.8v): Normal Mode: 5 Fast Mode: 1 Safe Mode: 2 BOOT_SEL3 == 1 (0.7v): Normal Mode: 2 Fast Mode: 2 Safe Mode: 3
Reserved	[7:5]	RO	3'h0	Reserved
VPLL_REFD IV	[4:0]	RW	5'h1	REFDIV. NOTE:Default value when BOOT_SEL3 == 0 (0.8v): Normal Mode: 1 Fast Mode: 2 Safe Mode: 1 BOOT_SEL3 == 1 (0.7v): Normal Mode: 1 Fast Mode: 1 Safe Mode: 1

8.4.2.53 DDR PLL 0 control (0xF8)

Field Name	Bit	Type	Reset Value	Field Description
DPLL0_FAST_CONFIG_EN	[31]	WO	1'h0	Fast Config Mode Enable 1: Enable Fast Config Mode. In this mode, only FBDIV can be modified, and there will be no PLL Power-Down sequence in PLL frequency update. 0: Disable Fast Config Mode.
Reserved	[30:28]	RO	3'h0	Reserved
DPLL0_FBDIV	[27:16]	RW	12'h30	FBDIV. NOTE:Default value BOOT_SEL3 == 0 (0.8v): Normal Mode: 48 Fast Mode: 85 Safe Mode: 37 BOOT_SEL3 == 1 (0.7v): Normal Mode: 36 Fast Mode: 255 Safe Mode: 111
Reserved	[15]	RO	1'h0	Reserved
DPLL0_POSTDIV2	[14:12]	RW	3'h1	POSTDIV2. NOTE:Default value when BOOT_SEL3 == 0 (0.8v): Normal Mode: 1 Fast Mode: 1 Safe Mode: 1 BOOT_SEL3 == 1 (0.7v): Normal Mode: 1 Fast Mode: 1 Safe Mode: 1
Reserved	[11]	RO	1'h0	Reserved
DPLL0_POSTDIV1	[10:8]	RW	3'h2	POSTDIV1. NOTE:Default value when BOOT_SEL3 == 0 (0.8v): Normal Mode: 2 Fast Mode: 1 Safe Mode: 2 BOOT_SEL3 == 1 (0.7v): Normal Mode: 2 Fast Mode: 4 Safe Mode: 4
Reserved	[7:5]	RO	3'h0	Reserved
DPLL0_REFDIV	[4:0]	RW	5'h1	REFDIV. NOTE:Default value whenBOOT_SEL3 == 0 (0.8v): Normal Mode: 1 Fast Mode: 2 Safe Mode: 1 BOOT_SEL3 == 1 (0.7v): Normal Mode: 1 Fast Mode: 2 Safe Mode: 2

8.4.2.54 DDR PLL 1 control (0xFC)

Field Name	Bit	Type	Reset Value	Field Description
DPLL1_FAST_CONFIG_EN	[31]	WO	1'h0	Fast Config Mode Enable 1: Enable Fast Config Mode. In this mode, only FBDIV can be modified, and there will be no PLL Power-Down sequence in PLL frequency update. 0: Disable Fast Config Mode.

Reserved	[30:28]	RO	3'h0	Reserved
DPLL1_FBDIV	[27:16]	RW	12'h30	FBDIV. NOTE:Default value BOOT_SEL3 == 0 (0.8v): Normal Mode: 48 Fast Mode: 85 Safe Mode: 37 BOOT_SEL3 == 1 (0.7v): Normal Mode: 36 Fast Mode: 255 Safe Mode: 111
Reserved	[15]	RO	1'h0	Reserved
DPLL1_POSTDIV2	[14:12]	RW	3'h1	POSTDIV2. NOTE:Default value when BOOT_SEL3 == 0 (0.8v): Normal Mode: 1 Fast Mode: 1 Safe Mode: 1 BOOT_SEL3 == 1 (0.7v): Normal Mode: 1 Fast Mode: 1 Safe Mode: 1
Reserved	[11]	RO	1'h0	Reserved
DPLL1_POSTDIV1	[10:8]	RW	3'h2	POSTDIV1. NOTE:Default value when BOOT_SEL3 == 0 (0.8v): Normal Mode: 2 Fast Mode: 1 Safe Mode: 2 BOOT_SEL3 == 1 (0.7v): Normal Mode: 2 Fast Mode: 4 Safe Mode: 4
Reserved	[7:5]	RO	3'h0	Reserved
DPLL1_REFDIV	[4:0]	RW	5'h1	REFDIV. NOTE:Default value when BOOT_SEL3 == 0 (0.8v): Normal Mode: 1 Fast Mode: 2 Safe Mode: 1 BOOT_SEL3 == 1 (0.7v): Normal Mode: 1 Fast Mode: 2 Safe Mode: 2

8.4.2.55 PMON_CNTR_REG0 (0x100)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:16]	RO	16'h0	Reserved
PMON_CNTR0	[15:0]	RO	16'h0	toreg_pm_count count value of process monitor

8.4.2.56 PMON_CNTR_REG1 (0x104)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:16]	RO	16'h0	Reserved
PMON_CNTR1	[15:0]	RO	16'h0	toreg_pm_count count value of process monitor

8.4.2.57 PMON_CNTR_REG2 (0x108)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:16]	RO	16'h0	Reserved
PMON_CNTR2	[15:0]	RO	16'h0	toreg_pm_count count value of process monitor

8.4.2.58 PMON_CNTR_REG3 (0x10C)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:16]	RO	16'h0	Reserved
PMON_CNTR3	[15:0]	RO	16'h0	toreg_pm_count count value of process monitor

8.4.2.59 PMON_CNTR_REG4 (0x110)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:16]	RO	16'h0	Reserved
PMON_CNTR4	[15:0]	RO	16'h0	toreg_pm_count count value of process monitor

8.4.2.60 PMON_CNTR_REG5 (0x114)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:16]	RO	16'h0	Reserved
PMON_CNTR5	[15:0]	RO	16'h0	toreg_pm_count count value of process monitor

8.4.2.61 PMON_CNTR_REG6 (0x118)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:16]	RO	16'h0	Reserved
PMON_CNTR6	[15:0]	RO	16'h0	toreg_pm_count count value of process monitor

8.4.2.62 PMON_CNTR_REG7 (0x11C)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:16]	RO	16'h0	Reserved

PMON_CNTR7	[15:0]	RO	16'h0	toreg_pm_count count value of process monitor

8.4.2.63 PMON_CNTR_REG8 (0x120)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:16]	RO	16'h0	Reserved
PMON_CNTR8	[15:0]	RO	16'h0	toreg_pm_count count value of process monitor

8.4.2.64 PMON_CNTR_REG9 (0x124)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:16]	RO	16'h0	Reserved
PMON_CNTR9	[15:0]	RO	16'h0	toreg_pm_count count value of process monitor

8.4.2.65 PMON_CNTR_REG10 (0x128)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:16]	RO	16'h0	Reserved
PMON_CNTR10	[15:0]	RO	16'h0	toreg_pm_count count value of process monitor

8.4.2.66 PMON_CNTR_REG11 (0x12C)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:16]	RO	16'h0	Reserved
PMON_CNTR11	[15:0]	RO	16'h0	toreg_pm_count count value of process monitor

8.4.2.67 PMON_CNTR_REG12 (0x130)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:16]	RO	16'h0	Reserved
PMON_CNTR12	[15:0]	RO	16'h0	toreg_pm_count count value of process monitor

8.4.2.68 PMON_CNTR_REG13 (0x134)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:16]	RO	16'h0	Reserved
PMON_CNTR13	[15:0]	RO	16'h0	toreg_pm_count count value of process monitor

8.4.2.69 PMON_CNTR_REG14 (0x138)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:16]	RO	16'h0	Reserved
PMON_CNTR14	[15:0]	RO	16'h0	toreg_pm_count count value of process monitor

8.4.2.70 PMON_CNTR_REG15 (0x13C)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:16]	RO	16'h0	Reserved
PMON_CNTR15	[15:0]	RO	16'h0	toreg_pm_count count value of process monitor

8.4.2.71 PMON_CNTR_REG16 (0x140)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:16]	RO	16'h0	Reserved
PMON_CNTR16	[15:0]	RO	16'h0	toreg_pm_count count value of process monitor

8.4.2.72 PMON_CNTR_REG17 (0x144)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:16]	RO	16'h0	Reserved
PMON_CNTR17	[15:0]	RO	16'h0	toreg_pm_count count value of process monitor

8.4.2.73 PMON_CNTR_REG18 (0x148)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:16]	RO	16'h0	Reserved

PMON_CNT R18	[15:0]	RO	16'h0	toreg_pm_count count value of process monitor

8.4.2.74 None Volatile Counter Disable (0x180)

Field Name	Bit	Typ e	Reset Value	Field Description
Reserved	[31:1]	RO	31'h0	Reserved
NV_CNTR_ DIS	[0]	RW	1'h0	None Volatile Counter Disable. When this bit is set high, the function of None Volatile Counter will be disabled, and cannot be enabled again unless power-down. The value of None Volatile Counter will be set to 64'hFFFF_FFFF_FFFF_FFFF.

8.4.2.75 None Volatile Counter Low (0x188)

Field Name	Bit	Typ e	Reset Value	Field Description
NV_CNTR_ LOW	[31:0]	RO	32'h0	Lower half of 64-bit None Volatile Counter. This counter starts counting after power-on-reset.

8.4.2.76 Watch Dog Reset Status (0x18C)

Field Name	Bit	Typ e	Reset Value	Field Description
Reserved	[31:1]	RO	31'h0	Reserved
WDT_RST_ HAPPENED	[0]	W1C	1'h0	Watch-Dog Reset Happened 1: Watch-Dog Reset happened This register is used to inidcate whether Watch-Dog Reset is happened. SW writes 1 to clear this bit.

8.4.2.77 GP_REG14 Set Register (0x190)

Field Name	Bit	Typ e	Reset Value	Field Description
GP_REG14 _SET	[31:0]	WO	32'h0	Write 1 into this register will also set the corresponding bit in GP_REG14. When SW writes value into this register, the behavior is shown as: GP_REG14 <= GP_REG14 | this_reg

8.4.2.78 GP_REG14 Clear Register (0x194)

Field Name	Bit	Typ e	Reset Value	Field Description
GP_REG14	[31:0]	WO	32'h0	Write 1 into this register will also clear the corresponding bit

_CLR				in GP_REG14. When SW writes value into this register, the behavior is shown as: GP_REG14 <= GP_REG14 & ~this_reg

8.4.2.79 GP_REG15 Set Register (0x198)

Field Name	Bit	Type	Reset Value	Field Description
GP_REG15_SET	[31:0]	WO	32'h0	Write 1 into this register will also set the corresponding bit in GP_REG15. When SW writes value into this register, the behavior is shown as: GP_REG15 <= GP_REG15 | this_reg

8.4.2.80 GP_REG15 Clear Register (0x19C)

Field Name	Bit	Type	Reset Value	Field Description
GP_REG15_CLR	[31:0]	WO	32'h0	Write 1 into this register will also clear the corresponding bit in GP_REG15. When SW writes value into this register, the behavior is shown as: GP_REG15 <= GP_REG15 & ~this_reg

8.4.2.81 Auto Clock Gating Enable Control Register 0 (0x1C0)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:14]	RO	18'h0	Reserved
HSPERI_ACG_EN	[13]	RW	1'h0	[HSPERI] High-Speed Peripheral Subsystem Auto Clock Gating Enable
VDE_ACG_EN	[12]	RW	1'h0	[VDE] Video Encoder Subsystem Auto Clock Gating Enable
EMMU_ACG_EN	[11]	RW	1'h0	[EMMU] EMMU Auto Clock Gating Enable
DDR_FAB_ACG_EN_800	[10]	RW	1'h0	[DDR] DDR Fabric Auto Clock Gating Enable for 800M Clock
DDR_FAB_ACG_EN_1150	[9]	RW	1'h0	[DDR] DDR Fabric Auto Clock Gating Enable for 1150M Clock
DDR_FAB_ACG_EN_575	[8]	RW	1'h0	[DDR] DDR Fabric Auto Clock Gating Enable for 575M Clock
VD1_FAB_ACG_EN	[7]	RW	1'h0	[VD1] Video Subsystem1 Fabric Auto Clock Gating Enable
VD0_FAB_ACG_EN	[6]	RW	1'h0	[VD0] Video Subsystem0 Fabric Auto Clock Gating Enable
PCIE_FAB_ACG_EN	[5]	RW	1'h0	[PCIe] PCIe Subsystem Fabric Auto Clock Gating Enable
TPU_FAB_ACG_EN	[4]	RW	1'h0	[TPU] TPU Subsystem Fabric Auto Clock Gating Enable

AP_FAB_ACG_EN	[3]	RW	1'h0	[AP] AP Subsystem Fabric Auto Clock Gating Enable
AXI_SRAM_ACG_EN	[2]	RW	1'h0	[TPU] TPU Subsystem AXI SRAM Auto Clock Gating Enable
TOP_FAB_ACG_EN	[1]	RW	1'h0	[TOP] Top Fabric Auto Clock Gating Enable
DDR_AW_W_ALGMNT_DIS	[0]	RW	1'h0	DDR AW W Alignment Disable 1: Disable DDR AW W Alignment 0: Enable DDR AW W Alignment (The write request will be sent to DDR only when the write data is also shown on DDR port.)

8.4.2.82 Auto Clock Gating Enable Control Register 1 (0x1C4)

Field Name	Bit	Type	Reset Value	Field Description
FAB25_ACG_IDLE_THR	[31:24]	RW	8'h40	[Top] Fabric 25 Auto Clock Gating Idle Threshold. After N cycles (N is defined by this register) of Fabric Idle, Fabric Low Power Controller will start Auto Clock Gating. This field can only be modified when bit[1] of Auto Clock Gating Enable Control Register 0 （0x01C0) is cleared. The function is only valid when bit[1] of Auto Clock Gating Enable Control Register 0 （0x01C0) is set.
FAB24_ACG_IDLE_THR	[23:16]	RW	8'h40	[Top] Fabric 24 Auto Clock Gating Idle Threshold. After N cycles (N is defined by this register) of Fabric Idle, Fabric Low Power Controller will start Auto Clock Gating. This field can only be modified when bit[1] of Auto Clock Gating Enable Control Register 0 （0x01C0) is cleared. The function is only valid when bit[1] of Auto Clock Gating Enable Control Register 0 （0x01C0) is set.
FAB3_ACG_IDLE_THR	[15:8]	RW	8'h10	[Top] Fabric 3 Auto Clock Gating Idle Threshold. After N cycles (N is defined by this register) of Fabric Idle, Fabric Low Power Controller will start Auto Clock Gating. This field can only be modified when bit[1] of Auto Clock Gating Enable Control Register 0 （0x01C0) is cleared. The function is only valid when bit[1] of Auto Clock Gating Enable Control Register 0 （0x01C0) is set.
FAB0_ACG_IDLE_THR	[7:0]	RW	8'h10	[Top] Fabric 0 Auto Clock Gating Idle Threshold. After N cycles (N is defined by this register) of Fabric Idle, Fabric Low Power Controller will start Auto Clock Gating. This field can only be modified when bit[1] of Auto Clock Gating Enable Control Register 0 （0x01C0) is cleared. The function is only valid when bit[1] of Auto Clock Gating Enable Control Register 0 （0x01C0) is set.

8.4.2.83 Auto Clock Gating Enable Control Register 2 (0x1C8)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:5]	RO	27'h0	Reserved
SW_CLK_EN	[4]	RW	1'h1	SW Clock Enable for AP Clock Gate Cell (1). Set: write 0x091BB283 to this register (2). Clear: write 0x3DFAF882 to this register
AP_IR_SW_CLK_EN	[3]	RW	1'h0	AP Irreversible SW Clock Enable for AP Clock Gate Cell (1). Set: write 0x724E10E0 to this register (2). Clear: N.A

CLSTR0_SW_CLK_EN	[2]	RW	1'h1	Cluster 0 SW Clock Enable for Cluster 0 Clock Gate Cell (1). Set: write 0x6EA5FACC to this register (2). Clear: write 0x2CF11C1D to this register
CLSTR1_SW_CLK_EN	[1]	RW	1'h1	Cluster 1 SW Clock Enable for Cluster 1 Clock Gate Cell (1). Set: write 0xAAF4FDDF to this register (2). Clear: write 0x822F2AA6 to this register
CCI_SW_CLK_EN	[0]	RW	1'h1	CCI SW Clock Enable for CCI Clock Gate Cell (1). Set: write 0xC65A439B to this register (2). Clear: write 0xB588B274 to this register

8.4.2.84 Auto Clock Gating Enable Control Register 3 (0x1CC)

Field Name	Bit	Type	Reset Value	Field Description
HSPERI_FAB_ACG_IDLE_THR	[31:24]	RW	8'h20	[HSPERI] Fabric Auto Clock Gating Idle Threshold. After N cycles (N is defined by this register) of Fabric Idle, Fabric Low Power Controller will start Auto Clock Gating. This field can only be modified when bit[13] of Auto Clock Gating Enable Control Register 0（0x1C0) is cleared. The function is only valid when bit[13] of Auto Clock Gating Enable Control Register 0（0x1C0) is set.
PCIE_FAB_ACG_IDLE_THR	[23:16]	RW	8'h20	[PCIe] Fabric Auto Clock Gating Idle Threshold. After N cycles (N is defined by this register) of Fabric Idle, Fabric Low Power Controller will start Auto Clock Gating. This field can only be modified when bit[5] of Auto Clock Gating Enable Control Register 0（0x1C0) is cleared. The function is only valid when bit[5] of Auto Clock Gating Enable Control Register 0（0x1C0) is set.
TPU_FAB_ACG_IDLE_THR	[15:8]	RW	8'h20	[TPU] Fabric Auto Clock Gating Idle Threshold. After N cycles (N is defined by this register) of Fabric Idle, Fabric Low Power Controller will start Auto Clock Gating. This field can only be modified when bit[4] of Auto Clock Gating Enable Control Register 0（0x1C0) is cleared. The function is only valid when bit[4] of Auto Clock Gating Enable Control Register 0（0x1C0) is set.
AP_FAB_ACG_IDLE_THR	[7:0]	RW	8'h20	[AP] Fabric Auto Clock Gating Idle Threshold. After N cycles (N is defined by this register) of Fabric Idle, Fabric Low Power Controller will start Auto Clock Gating. This field can only be modified when bit[3] of Auto Clock Gating Enable Control Register 0（0x1C0) is cleared. The function is only valid when bit[3] of Auto Clock Gating Enable Control Register 0（0x1C0) is set.

8.4.2.85 Auto Clock Gating Enable Control Register 4 (0x1D0)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:24]	RW	8'h0	Reserved
VDE_FAB_ACG_IDLE_THR	[23:16]	RW	8'h20	[VDE] Video Encoder Subsystem Fabric Auto Clock Gating Idle Threshold. After N cycles (N is defined by this register) of Fabric Idle, Fabric Low Power Controller will start Auto Clock Gating. This field can only be modified when bit[12] of Auto Clock Gating Enable Control Register 0（0x01C0) is cleared. The function is only valid when bit[12] of Auto Clock Gating Enable Control Register 0（0x01C0) is set.

VD1_FAB_ACG_IDLE_THR	[15:8]	RW	8'h20	[VD1] Video Subsystem 1 Fabric Auto Clock Gating Idle Threshold. After N cycles (N is defined by this register) of Fabric Idle, Fabric Low Power Controller will start Auto Clock Gating. This field can only be modified when bit[7] of Auto Clock Gating Enable Control Register 0 （0x01C0) is cleared. The function is only valid when bit[7] of Auto Clock Gating Enable Control Register 0 （0x01C0) is set.
VD0_FAB_ACG_IDLE_THR	[7:0]	RW	8'h20	[VD0] Video Subsystem 0 Fabric Auto Clock Gating Idle Threshold. After N cycles (N is defined by this register) of Fabric Idle, Fabric Low Power Controller will start Auto Clock Gating. This field can only be modified when bit[6] of Auto Clock Gating Enable Control Register 0 （0x01C0) is cleared. The function is only valid when bit[6] of Auto Clock Gating Enable Control Register 0 （0x01C0) is set.

8.4.2.86 Auto Clock Gating Enable Control Register 5 (0x1D4)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:24]	RW	8'h0	Reserved
DDR_FAB_ACG_IDLE_THR_800	[23:16]	RW	8'h40	[DDR] DDR Fabric Auto Clock Gating Idle Threshold for 800M Clock. After N cycles (N is defined by this register) of Fabric Idle, Fabric Low Power Controller will start Auto Clock Gating. This field can only be modified when bit[10] of Auto Clock Gating Enable Control Register 0 （0x01C0) is cleared. The function is only valid when bit[10] of Auto Clock Gating Enable Control Register 0 （0x01C0) is set.
DDR_FAB_ACG_IDLE_THR_1150	[15:8]	RW	8'h40	[DDR] DDR Fabric Auto Clock Gating Idle Threshold for 1150M Clock. After N cycles (N is defined by this register) of Fabric Idle, Fabric Low Power Controller will start Auto Clock Gating. This field can only be modified when bit[9] of Auto Clock Gating Enable Control Register 0 （0x01C0) is cleared. The function is only valid when bit[9] of Auto Clock Gating Enable Control Register 0 （0x01C0) is set.
DDR_FAB_ACG_IDLE_THR_575	[7:0]	RW	8'h40	[DDR] DDR Fabric Auto Clock Gating Idle Threshold for 575M Clock. After N cycles (N is defined by this register) of Fabric Idle, Fabric Low Power Controller will start Auto Clock Gating. This field can only be modified when bit[8] of Auto Clock Gating Enable Control Register 0 （0x1C0) is cleared. The function is only valid when bit[8] of Auto Clock Gating Enable Control Register 0 （0x1C0) is set.

8.4.2.87 General Purpose register16 (0x1E0)

Field Name	Bit	Type	Reset Value	Field Description
GP_REG16	[31:0]	RW	32'h0	General Purpose register for SW usage.

8.4.2.88 General Purpose register17 (0x1E4)

Field Name	Bit	Typ e	Reset Value	Field Description
GP_REG17	[31:0]	RW	32'h0	General Purpose register for SW usage.

8.4.2.89 General Purpose register18 (0x1E8)

Field Name	Bit	Typ e	Reset Value	Field Description
GP_REG18	[31:0]	RW	32'h0	General Purpose register for SW usage.

8.4.2.90 General Purpose register19 (0x1EC)

Field Name	Bit	Typ e	Reset Value	Field Description
GP_REG19	[31:0]	RW	32'h0	General Purpose register for SW usage.

8.4.2.91 General Purpose register20 (0x1F0)

Field Name	Bit	Typ e	Reset Value	Field Description
GP_REG20	[31:0]	RW	32'h0	General Purpose register for SW usage.

8.4.2.92 General Purpose register21 (0x1F4)

Field Name	Bit	Typ e	Reset Value	Field Description
GP_REG21	[31:0]	RW	32'h0	General Purpose register for SW usage.

8.4.2.93 General Purpose register22 (0x1F8)

Field Name	Bit	Typ e	Reset Value	Field Description
GP_REG22	[31:0]	RW	32'h0	General Purpose register for SW usage.

8.4.2.94 General Purpose register23 (0x1FC)

Field Name	Bit	Typ e	Reset Value	Field Description
GP_REG23	[31:0]	RW	32'h0	General Purpose register for SW usage.

8.4.2.95 General Purpose register24 (0x200)

Field Name	Bit	Type	Reset Value	Field Description
GP_REG24	[31:0]	RW	32'h0	General Purpose register for SW usage.

8.4.2.96 General Purpose register25 (0x204)

Field Name	Bit	Type	Reset Value	Field Description
GP_REG25	[31:0]	RW	32'h0	General Purpose register for SW usage.

8.4.2.97 General Purpose register26 (0x208)

Field Name	Bit	Type	Reset Value	Field Description
GP_REG26	[31:0]	RW	32'h0	General Purpose register for SW usage.

8.4.2.98 General Purpose register27 (0x20C)

Field Name	Bit	Type	Reset Value	Field Description
GP_REG27	[31:0]	RW	32'h0	General Purpose register for SW usage.

8.4.2.99 General Purpose register28 (0x210)

Field Name	Bit	Type	Reset Value	Field Description
GP_REG28	[31:0]	RW	32'h0	General Purpose register for SW usage.

8.4.2.100 General Purpose register29 (0x214)

Field Name	Bit	Type	Reset Value	Field Description
GP_REG29	[31:0]	RW	32'h0	General Purpose register for SW usage.

8.4.2.101 General Purpose register30 (0x218)

Field Name	Bit	Type	Reset Value	Field Description
GP_REG30	[31:0]	RW	32'h0	General Purpose register for SW usage.

8.4.2.102 General Purpose register31 (0x21C)

Field Name	Bit	Type	Reset Value	Field Description
GP_REG31	[31:0]	RW	32'h0	General Purpose register for SW usage.

8.5 IO/Pinmux Control and Status Registers

8.5.1 IO/Pinmux Control and Status Registers Summary

Table 15 IO/Pinmux Control and Status Registers Summary

Offset	Register Name	Reg Description
0x00	PM_IO_CTR_0	pin mux and io control register for pciee_clkreq_x and pcier_clkreq_x
0x04	PM_IO_CTR_1	pin mux and io control register for pciee_rst_x and pcier_rst_x
0x08	PM_IO_CTR_2	pin mux and io control register for pciee_wakeup_x_out and pcier_wakeup_x_out
0x0C	PM_IO_CTR_3	pin mux and io control register for spif_clk_sel1 and spif_wp_x
0x10	PM_IO_CTR_4	pin mux and io control register for spif_hold_x and spif_sdi
0x14	PM_IO_CTR_5	pin mux and io control register for spif_cs_x and spif_sck
0x18	PM_IO_CTR_6	pin mux and io control register for spif_sdo and emmc_wp
0x1C	PM_IO_CTR_7	pin mux and io control register for emmc_cd_x and emmc_rst_x
0x20	PM_IO_CTR_8	pin mux and io control register for emmc_pwr_en and sdio_cd_x
0x24	PM_IO_CTR_9	pin mux and io control register for sdio_wp and sdio_rst_x
0x28	PM_IO_CTR_10	pin mux and io control register for sdio_pwr_en and rgmii0_txd0
0x2C	PM_IO_CTR_11	pin mux and io control register for rgmii0_txd1 and rgmii0_txd2
0x30	PM_IO_CTR_12	pin mux and io control register for rgmii0_txd3 and rgmii0_txctrl
0x34	PM_IO_CTR_13	pin mux and io control register for rgmii0_rxd0 and rgmii0_rxd1
0x38	PM_IO_CTR_14	pin mux and io control register for rgmii0_rxd2 and rgmii0_rxd3
0x3C	PM_IO_CTR_15	pin mux and io control register for rgmii0_rxctrl and rgmii0_txc
0x40	PM_IO_CTR_16	pin mux and io control register for rgmii0_rxc and rgmii0_refclko
0x44	PM_IO_CTR_17	pin mux and io control register for rgmii0_irq and rgmii0_mdc
0x48	PM_IO_CTR_18	pin mux and io control register for rgmii0_mdio and rgmii1_txd0
0x4C	PM_IO_CTR_19	pin mux and io control register for rgmii1_txd1 and rgmii1_txd2
0x50	PM_IO_CTR_20	pin mux and io control register for rgmii1_txd3 and rgmii1_txctrl
0x54	PM_IO_CTR_21	pin mux and io control register for rgmii1_rxd0 and rgmii1_rxd1
0x58	PM_IO_CTR_22	pin mux and io control register for rgmii1_rxd2 and rgmii1_rxd3
0x5C	PM_IO_CTR_23	pin mux and io control register for rgmii1_rxctrl and rgmii1_txc
0x60	PM_IO_CTR_24	pin mux and io control register for rgmii1_rxc and rgmii1_refclko
0x64	PM_IO_CTR_25	pin mux and io control register for rgmii1_irq and rgmii1_mdc
0x68	PM_IO_CTR_26	pin mux and io control register for rgmii1_mdio and pwm0
0x6C	PM_IO_CTR_27	pin mux and io control register for pwm1 and fan0
0x70	PM_IO_CTR_28	pin mux and io control register for fan1 and iic0_sda
0x74	PM_IO_CTR_29	pin mux and io control register for iic0_scl and iic1_sda
0x78	PM_IO_CTR_30	pin mux and io control register for iic1_scl and iic2_sda

0x7C	PM_IO_CTR_31	pin mux and io control register for iic2_scl and uart0_tx
0x80	PM_IO_CTR_32	pin mux and io control register for uart0_rx and uart1_tx
0x84	PM_IO_CTR_33	pin mux and io control register for uart1_rx and uart2_tx
0x88	PM_IO_CTR_34	pin mux and io control register for uart2_rx and gpio0
0x8C	PM_IO_CTR_35	pin mux and io control register for gpio1 and gpio2
0x90	PM_IO_CTR_36	pin mux and io control register for gpio3 and gpio4
0x94	PM_IO_CTR_37	pin mux and io control register for gpio5 and gpio6
0x98	PM_IO_CTR_38	pin mux and io control register for gpio7 and gpio8
0x9C	PM_IO_CTR_39	pin mux and io control register for gpio9 and gpio10
0xA0	PM_IO_CTR_40	pin mux and io control register for gpio11 and gpio12
0xA4	PM_IO_CTR_41	pin mux and io control register for gpio13 and gpio14
0xA8	PM_IO_CTR_42	pin mux and io control register for gpio15 and gpio16
0xAC	PM_IO_CTR_43	pin mux and io control register for gpio17 and gpio18
0xB0	PM_IO_CTR_44	pin mux and io control register for gpio19 and gpio20
0xB4	PM_IO_CTR_45	pin mux and io control register for gpio21 and gpio22
0xB8	PM_IO_CTR_46	pin mux and io control register for gpio23 and gpio24
0xBC	PM_IO_CTR_47	pin mux and io control register for gpio25 and gpio26
0xC0	PM_IO_CTR_48	pin mux and io control register for gpio27 and gpio28
0xC4	PM_IO_CTR_49	pin mux and io control register for gpio29 and gpio30
0xC8	PM_IO_CTR_50	pin mux and io control register for gpio31 and boot_sel0
0xCC	PM_IO_CTR_51	pin mux and io control register for boot_sel1 and boot_sel2
0xD0	PM_IO_CTR_52	pin mux and io control register for boot_sel3 and mode_sel0
0xD4	PM_IO_CTR_53	pin mux and io control register for mode_sel1 and mode_sel2
0xD8	PM_IO_CTR_54	pin mux and io control register for mode_sel3 and pll_clk_in
0xDC	PM_IO_CTR_55	pin mux and io control register for pll_clk_in_dpll0 and pll_clk_in_dpll1
0xE0	PM_IO_CTR_56	pin mux and io control register for pll_src_sel and sys_rst_x
0xE4	PM_IO_CTR_57	pin mux and io control register for test_en and bisr_byp
0xE8	PM_IO_CTR_58	pin mux and io control register for p08_pwr_good and osc_ds3
0xEC	PM_IO_CTR_59	pin mux and io control register for ddr_pwr_good

8.5.2 IO/Pinmux Control and Status Registers Description

8.5.2.1 pin mux and io control register for pciee_clkreq_x and pcier_clkreq_x(0x00)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_pcier_clkreq_x	[27]	RW	1'h1	pad oex enable for pcier_clkreq_x. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_pcier_clkreq_x	[26]	RW	1'h1	schmitt trigger enable for pcier_clkreq_x. '1' enables schmitt trigger input function
driving_selector_pcier_clkreq_x	[25:22]	RW	4'h8	driving selector for pcier_clkreq_x
pin_mux_selector_pcier_clkreq_x	[21:20]	RW	2'h0	pin mux selector for pcier_clkreq_x
pull_down_enable_pcier_clkreq_x	[19]	RW	1'h0	pull down enable for pcier_clkreq_x
pull_up_enable_pci	[18]	RW	1'h1	pull up enable for pcier_clkreq_x

er_clkreq_x				
reserved	[17:16]	RW	2'h0	reserved
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_pciee_clkreq_x	[11]	RW	1'h1	pad oex enable for pciee_clkreq_x. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_pciee_clkreq_x	[10]	RW	1'h1	schmitt trigger enable for pciee_clkreq_x. '1' enables schmitt trigger input function
driving_selector_pciee_clkreq_x	[9:6]	RW	4'h8	driving selector for pciee_clkreq_x
pin_mux_selector_pciee_clkreq_x	[5:4]	RW	2'h0	pin mux selector for pciee_clkreq_x
pull_down_enable_pciee_clkreq_x	[3]	RW	1'h0	pull down enable for pciee_clkreq_x
pull_up_enable_pciee_clkreq_x	[2]	RW	1'h1	pull up enable for pciee_clkreq_x
reserved	[1]	RW	2'h0	reserved

8.5.2.2 pin mux and io control register for pciee_rst_x and pcier_rst_x (0x04)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_pcier_rst_x	[27]	RW	1'h1	pad oex enable for pcier_rst_x. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_pcier_rst_x	[26]	RW	1'h1	schmitt trigger enable for pcier_rst_x. '1' enables schmitt trigger input function
driving_selector_pcier_rst_x	[25:22]	RW	4'h8	driving selector for pcier_rst_x
pin_mux_selector_pcier_rst_x	[21:20]	RW	2'h0	pin mux selector for pcier_rst_x
pull_down_enable_pcier_rst_x	[19]	RW	1'h1	pull down enable for pcier_rst_x
pull_up_enable_pcier_rst_x	[18]	RW	1'h0	pull up enable for pcier_rst_x
reserved	[17:16]	RW	2'h0	reserved
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_pciee_rst_x	[11]	RW	1'h1	pad oex enable for pciee_rst_x. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_pciee_rst_x	[10]	RW	1'h1	schmitt trigger enable for pciee_rst_x. '1' enables schmitt trigger input function
driving_selector_pciee_rst_x	[9:6]	RW	4'h8	driving selector for pciee_rst_x
pin_mux_selector_pciee_rst_x	[5:4]	RW	2'h0	pin mux selector for pciee_rst_x
pull_down_enable_pciee_rst_x	[3]	RW	1'h1	pull down enable for pciee_rst_x
pull_up_enable_pciee_rst_x	[2]	RW	1'h0	pull up enable for pciee_rst_x
reserved	[1]	RW	2'h0	reserved

8.5.2.3 pin mux and io control register for pciee_wakeup_x_out and pcier_wakeup_x_out (0x08)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_pcier_wakeup_x_out	[27]	RW	1'h1	pad oex enable for pcier_wakeup_x_out. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_pcier_wakeup_x_out	[26]	RW	1'h1	schmitt trigger enable for pcier_wakeup_x_out. '1' enables schmitt trigger input function
driving_selector_pcier_wakeup_x_out	[25:22]	RW	4'h8	driving selector for pcier_wakeup_x_out
pin_mux_selector_pcier_wakeup_x_out	[21:20]	RW	2'h0	pin mux selector for pcier_wakeup_x_out
pull_down_enable_pcier_wakeup_x_out	[19]	RW	1'h0	pull down enable for pcier_wakeup_x_out
pull_up_enable_pcier_wakeup_x_out	[18]	RW	1'h1	pull up enable for pcier_wakeup_x_out
reserved	[17:16]	RW	2'h0	reserved
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_pciee_wakeup_x_out	[11]	RW	1'h1	pad oex enable for pciee_wakeup_x_out. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_pciee_wakeup_x_out	[10]	RW	1'h1	schmitt trigger enable for pciee_wakeup_x_out. '1' enables schmitt trigger input function
driving_selector_pciee_wakeup_x_out	[9:6]	RW	4'h8	driving selector for pciee_wakeup_x_out
pin_mux_selector_pciee_wakeup_x_out	[5:4]	RW	2'h0	pin mux selector for pciee_wakeup_x_out
pull_down_enable_pciee_wakeup_x_out	[3]	RW	1'h0	pull down enable for pciee_wakeup_x_out
pull_up_enable_pciee_wakeup_x_out	[2]	RW	1'h1	pull up enable for pciee_wakeup_x_out
reserved	[1]	RW	2'h0	reserved

8.5.2.4 pin mux and io control register for spif_clk_sel1 and spif_wp_x (0x0C)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_spif_wp_x	[27]	RW	1'h1	pad oex enable for spif_wp_x. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_spif_wp_x	[26]	RW	1'h0	schmitt trigger enable for spif_wp_x. '1' enables schmitt trigger input function
driving_selector_spif_wp_x	[25:22]	RW	4'h8	driving selector for spif_wp_x
pin_mux_selector_spif_wp_x	[21:20]	RW	2'h0	pin mux selector for spif_wp_x
reserved	[19:18]	RW	2'h0	reserved
pull_selector_spif_wp_x	[17]	RW	1'h0	pull selector for spif_wp_x.
pull_enable_spif_w	[16]	RW	1'h0	pull enable for spif_wp_x.

p_x				
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_spif_clk_sel1	[11]	RW	1'h1	pad oex enable for spif_clk_sel1. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_spif_clk_sel1	[10]	RW	1'h0	schmitt trigger enable for spif_clk_sel1. '1' enables schmitt trigger input function
driving_selector_spif_clk_sel1	[9:6]	RW	4'h8	driving selector for spif_clk_sel1
pin_mux_selector_spif_clk_sel1	[5:4]	RW	2'h0	pin mux selector for spif_clk_sel1
reserved	[3:2]	RW	2'h0	reserved
pull_selector_spif_clk_sel1	[1]	RW	1'h0	pull selector for spif_clk_sel1.
pull_enable_spif_clk_sel1	[0]	RW	1'h0	pull enable for spif_clk_sel1.

8.5.2.5 pin mux and io control register for spif_hold_x and spif_sdi (0x10)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_spif_sdi	[27]	RW	1'h1	pad oex enable for spif_sdi. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_spif_sdi	[26]	RW	1'h0	schmitt trigger enable for spif_sdi. '1' enables schmitt trigger input function
driving_selector_spif_sdi	[25:22]	RW	4'h8	driving selector for spif_sdi
pin_mux_selector_spif_sdi	[21:20]	RW	2'h0	pin mux selector for spif_sdi
reserved	[19:18]	RW	2'h0	reserved
pull_selector_spif_sdi	[17]	RW	1'h0	pull selector for spif_sdi.
pull_enable_spif_sdi	[16]	RW	1'h0	pull enable for spif_sdi.
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_spif_hold_x	[11]	RW	1'h1	pad oex enable for spif_hold_x. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_spif_hold_x	[10]	RW	1'h0	schmitt trigger enable for spif_hold_x. '1' enables schmitt trigger input function
driving_selector_spif_hold_x	[9:6]	RW	4'h8	driving selector for spif_hold_x
pin_mux_selector_spif_hold_x	[5:4]	RW	2'h0	pin mux selector for spif_hold_x
reserved	[3:2]	RW	2'h0	reserved
pull_selector_spif_hold_x	[1]	RW	1'h0	pull selector for spif_hold_x.
pull_enable_spif_hold_x	[0]	RW	1'h0	pull enable for spif_hold_x.

8.5.2.6 pin mux and io control register for spif_cs_x and spif_sck (0x14)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:28]	RW	4'h0	reserved

pad_oex_enable_spif_sck	[27]	RW	1'h1	pad oex enable for spif_sck. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_spif_sck	[26]	RW	1'h0	schmitt trigger enable for spif_sck. '1' enables schmitt trigger input function
driving_selector_spif_sck	[25:22]	RW	4'h8	driving selector for spif_sck
pin_mux_selector_spif_sck	[21:20]	RW	2'h0	pin mux selector for spif_sck
reserved	[19:18]	RW	2'h0	reserved
pull_selector_spif_sck	[17]	RW	1'h0	pull selector for spif_sck.
pull_enable_spif_sck	[16]	RW	1'h0	pull enable for spif_sck.
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_spif_cs_x	[11]	RW	1'h1	pad oex enable for spif_cs_x. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_spif_cs_x	[10]	RW	1'h0	schmitt trigger enable for spif_cs_x. '1' enables schmitt trigger input function
driving_selector_spif_cs_x	[9:6]	RW	4'h8	driving selector for spif_cs_x
pin_mux_selector_spif_cs_x	[5:4]	RW	2'h0	pin mux selector for spif_cs_x
reserved	[3:2]	RW	2'h0	reserved
pull_selector_spif_cs_x	[1]	RW	1'h0	pull selector for spif_cs_x.
pull_enable_spif_cs_x	[0]	RW	1'h0	pull enable for spif_cs_x.

8.5.2.7 pin mux and io control register for spif_sdo and emmc_wp (0x18)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_emmc_wp	[27]	RW	1'h1	pad oex enable for emmc_wp. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_emmc_wp	[26]	RW	1'h1	schmitt trigger enable for emmc_wp. '1' enables schmitt trigger input function
driving_selector_emmc_wp	[25:22]	RW	4'h8	driving selector for emmc_wp
pin_mux_selector_emmc_wp	[21:20]	RW	2'h0	pin mux selector for emmc_wp
reserved	[19:18]	RW	2'h0	reserved
pull_selector_emmc_wp	[17]	RW	1'h0	pull selector for emmc_wp.
pull_enable_emmc_wp	[16]	RW	1'h0	pull enable for emmc_wp.
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_spif_sdo	[11]	RW	1'h1	pad oex enable for spif_sdo. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_spif_sdo	[10]	RW	1'h0	schmitt trigger enable for spif_sdo. '1' enables schmitt trigger input function
driving_selector_spif_sdo	[9:6]	RW	4'h8	driving selector for spif_sdo
pin_mux_selector_spif_sdo	[5:4]	RW	2'h0	pin mux selector for spif_sdo
reserved	[3:2]	RW	2'h0	reserved

pull_selector_spif_sdo	[1]	RW	1'h0	pull selector for spif_sdo.
pull_enable_spif_sdo	[0]	RW	1'h0	pull enable for spif_sdo.

8.5.2.8 pin mux and io control register for emmc_cd_x and emmc_rst_x (0x1C)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_emmc_rst_x	[27]	RW	1'h1	pad oex enable for emmc_rst_x. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_emmc_rst_x	[26]	RW	1'h0	schmitt trigger enable for emmc_rst_x. '1' enables schmitt trigger input function
driving_selector_emmc_rst_x	[25:22]	RW	4'h8	driving selector for emmc_rst_x
pin_mux_selector_emmc_rst_x	[21:20]	RW	2'h0	pin mux selector for emmc_rst_x
reserved	[19:18]	RW	2'h0	reserved
pull_selector_emmc_rst_x	[17]	RW	1'h0	pull selector for emmc_rst_x.
pull_enable_emmc_rst_x	[16]	RW	1'h0	pull enable for emmc_rst_x.
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_emmc_cd_x	[11]	RW	1'h1	pad oex enable for emmc_cd_x. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_emmc_cd_x	[10]	RW	1'h1	schmitt trigger enable for emmc_cd_x. '1' enables schmitt trigger input function
driving_selector_emmc_cd_x	[9:6]	RW	4'h8	driving selector for emmc_cd_x
pin_mux_selector_emmc_cd_x	[5:4]	RW	2'h0	pin mux selector for emmc_cd_x
reserved	[3:2]	RW	2'h0	reserved
pull_selector_emmc_cd_x	[1]	RW	1'h1	pull selector for emmc_cd_x.
pull_enable_emmc_cd_x	[0]	RW	1'h1	pull enable for emmc_cd_x.

8.5.2.9 pin mux and io control register for emmc_pwr_en and sdio_cd_x (0x20)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_sdio_cd_x	[27]	RW	1'h1	pad oex enable for sdio_cd_x. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_sdio_cd_x	[26]	RW	1'h1	schmitt trigger enable for sdio_cd_x. '1' enables schmitt trigger input function
driving_selector_sdio_cd_x	[25:22]	RW	4'h8	driving selector for sdio_cd_x
pin_mux_selector_sdio_cd_x	[21:20]	RW	2'h0	pin mux selector for sdio_cd_x
reserved	[19:18]	RW	2'h0	reserved
pull_selector_sdio_cd_x	[17]	RW	1'h1	pull selector for sdio_cd_x.

pull_enable_sdio_cd_x	[16]	RW	1'h1	pull enable for sdio_cd_x.
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_emmc_pwr_en	[11]	RW	1'h1	pad oex enable for emmc_pwr_en. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_emmc_pwr_en	[10]	RW	1'h0	schmitt trigger enable for emmc_pwr_en. '1' enables schmitt trigger input function
driving_selector_emmc_pwr_en	[9:6]	RW	4'h8	driving selector for emmc_pwr_en
pin_mux_selector_emmc_pwr_en	[5:4]	RW	2'h0	pin mux selector for emmc_pwr_en
reserved	[3:2]	RW	2'h0	reserved
pull_selector_emmc_pwr_en	[1]	RW	1'h0	pull selector for emmc_pwr_en.
pull_enable_emmc_pwr_en	[0]	RW	1'h0	pull enable for emmc_pwr_en.

8.5.2.10 pin mux and io control register for sdio_wp and sdio_rst_x (0x24)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_sdio_rst_x	[27]	RW	1'h1	pad oex enable for sdio_rst_x. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_sdio_rst_x	[26]	RW	1'h0	schmitt trigger enable for sdio_rst_x. '1' enables schmitt trigger input function
driving_selector_sdio_rst_x	[25:22]	RW	4'h8	driving selector for sdio_rst_x
pin_mux_selector_sdio_rst_x	[21:20]	RW	2'h0	pin mux selector for sdio_rst_x
reserved	[19:18]	RW	2'h0	reserved
pull_selector_sdio_rst_x	[17]	RW	1'h0	pull selector for sdio_rst_x.
pull_enable_sdio_rst_x	[16]	RW	1'h0	pull enable for sdio_rst_x.
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_sdio_wp	[11]	RW	1'h1	pad oex enable for sdio_wp. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_sdio_wp	[10]	RW	1'h1	schmitt trigger enable for sdio_wp. '1' enables schmitt trigger input function
driving_selector_sdio_wp	[9:6]	RW	4'h8	driving selector for sdio_wp
pin_mux_selector_sdio_wp	[5:4]	RW	2'h0	pin mux selector for sdio_wp
reserved	[3:2]	RW	2'h0	reserved
pull_selector_sdio_wp	[1]	RW	1'h0	pull selector for sdio_wp.
pull_enable_sdio_wp	[0]	RW	1'h0	pull enable for sdio_wp.

8.5.2.11 pin mux and io control register for sdio_pwr_en and rgmii0_txd0 (0x28)

Field Name	Bit	Type	Reset Value	Field Description

reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_rgmii0_txd0	[27]	RW	1'h1	pad oex enable for rgmii0_txd0. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_rgmii0_txd0	[26]	RW	1'h0	schmitt trigger enable for rgmii0_txd0. '1' enables schmitt trigger input function
driving_selector_rgmii0_txd0	[25:22]	RW	4'h8	driving selector for rgmii0_txd0
pin_mux_selector_rgmii0_txd0	[21:20]	RW	2'h0	pin mux selector for rgmii0_txd0
pull_down_enable_rgmii0_txd0	[19]	RW	1'h0	pull down enable for rgmii0_txd0
pull_up_enable_rgmii0_txd0	[18]	RW	1'h0	pull up enable for rgmii0_txd0
reserved	[17:16]	RW	2'h0	reserved
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_sdio_pwr_en	[11]	RW	1'h1	pad oex enable for sdio_pwr_en. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_sdio_pwr_en	[10]	RW	1'h0	schmitt trigger enable for sdio_pwr_en. '1' enables schmitt trigger input function
driving_selector_sdio_pwr_en	[9:6]	RW	4'h8	driving selector for sdio_pwr_en
pin_mux_selector_sdio_pwr_en	[5:4]	RW	2'h0	pin mux selector for sdio_pwr_en
reserved	[3:2]	RW	2'h0	reserved
pull_selector_sdio_pwr_en	[1]	RW	1'h0	pull selector for sdio_pwr_en.
pull_enable_sdio_pwr_en	[0]	RW	1'h0	pull enable for sdio_pwr_en.

8.5.2.12 pin mux and io control register for rgmii0_txd1 and rgmii0_txd2 (0x2C)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_rgmii0_txd2	[27]	RW	1'h1	pad oex enable for rgmii0_txd2. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_rgmii0_txd2	[26]	RW	1'h0	schmitt trigger enable for rgmii0_txd2. '1' enables schmitt trigger input function
driving_selector_rgmii0_txd2	[25:22]	RW	4'h8	driving selector for rgmii0_txd2
pin_mux_selector_rgmii0_txd2	[21:20]	RW	2'h0	pin mux selector for rgmii0_txd2
pull_down_enable_rgmii0_txd2	[19]	RW	1'h0	pull down enable for rgmii0_txd2
pull_up_enable_rgmii0_txd2	[18]	RW	1'h0	pull up enable for rgmii0_txd2
reserved	[17:16]	RW	2'h0	reserved
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_rgmii0_txd1	[11]	RW	1'h1	pad oex enable for rgmii0_txd1. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_rgmii0_txd1	[10]	RW	1'h0	schmitt trigger enable for rgmii0_txd1. '1' enables schmitt trigger input function
driving_selector_rgmii0_txd1	[9:6]	RW	4'h8	driving selector for rgmii0_txd1
pin_mux_selector_rgmii0_txd1	[5:4]	RW	2'h0	pin mux selector for rgmii0_txd1

pull_down_enable_rgmii0_txd1	[3]	RW	1'h0	pull down enable for rgmii0_txd1
pull_up_enable_rgmii0_txd1	[2]	RW	1'h0	pull up enable for rgmii0_txd1
reserved	[1:0]	RW	2'h0	reserved

8.5.2.13 pin mux and io control register for rgmii0_txd3 and rgmii0_txctrl (0x30)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_rgmii0_txctrl	[27]	RW	1'h1	pad oex enable for rgmii0_txctrl. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_rgmii0_txctrl	[26]	RW	1'h0	schmitt trigger enable for rgmii0_txctrl. '1' enables schmitt trigger input function
driving_selector_rgmii0_txctrl	[25:22]	RW	4'h8	driving selector for rgmii0_txctrl
pin_mux_selector_rgmii0_txctrl	[21:20]	RW	2'h0	pin mux selector for rgmii0_txctrl
pull_down_enable_rgmii0_txctrl	[19]	RW	1'h0	pull down enable for rgmii0_txctrl
pull_up_enable_rgmii0_txctrl	[18]	RW	1'h0	pull up enable for rgmii0_txctrl
reserved	[17:16]	RW	2'h0	reserved
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_rgmii0_txd3	[11]	RW	1'h1	pad oex enable for rgmii0_txd3. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_rgmii0_txd3	[10]	RW	1'h0	schmitt trigger enable for rgmii0_txd3. '1' enables schmitt trigger input function
driving_selector_rgmii0_txd3	[9:6]	RW	4'h8	driving selector for rgmii0_txd3
pin_mux_selector_rgmii0_txd3	[5:4]	RW	2'h0	pin mux selector for rgmii0_txd3
pull_down_enable_rgmii0_txd3	[3]	RW	1'h0	pull down enable for rgmii0_txd3
pull_up_enable_rgmii0_txd3	[2]	RW	1'h0	pull up enable for rgmii0_txd3
reserved	[1:0]	RW	2'h0	reserved

8.5.2.14 pin mux and io control register for rgmii0_rxd0 and rgmii0_rxd1 (0x34)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_rgmii0_rxd1	[27]	RW	1'h1	pad oex enable for rgmii0_rxd1. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_rgmii0_rxd1	[26]	RW	1'h0	schmitt trigger enable for rgmii0_rxd1. '1' enables schmitt trigger input function
driving_selector_rgmii0_rxd1	[25:22]	RW	4'h8	driving selector for rgmii0_rxd1
pin_mux_selector_r	[21:20]	RW	2'h0	pin mux selector for rgmii0_rxd1

gmii0_rxd1				
pull_down_enable_rgmii0_rxd1	[19]	RW	1'h0	pull down enable for rgmii0_rxd1
pull_up_enable_rgmii0_rxd1	[18]	RW	1'h0	pull up enable for rgmii0_rxd1
reserved	[17:16]	RW	2'h0	reserved
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_rgmii0_rxd0	[11]	RW	1'h1	pad oex enable for rgmii0_rxd0. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_rgmii0_rxd0	[10]	RW	1'h0	schmitt trigger enable for rgmii0_rxd0. '1' enables schmitt trigger input function
driving_selector_rgmii0_rxd0	[9:6]	RW	4'h8	driving selector for rgmii0_rxd0
pin_mux_selector_rgmii0_rxd0	[5:4]	RW	2'h0	pin mux selector for rgmii0_rxd0
pull_down_enable_rgmii0_rxd0	[3]	RW	1'h0	pull down enable for rgmii0_rxd0
pull_up_enable_rgmii0_rxd0	[2]	RW	1'h0	pull up enable for rgmii0_rxd0
reserved	[1:0]	RW	2'h0	reserved

8.5.2.15 pin mux and io control register for rgmii0_rxd2 and rgmii0_rxd3 (0x38)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_rgmii0_rxd3	[27]	RW	1'h1	pad oex enable for rgmii0_rxd3. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_rgmii0_rxd3	[26]	RW	1'h0	schmitt trigger enable for rgmii0_rxd3. '1' enables schmitt trigger input function
driving_selector_rgmii0_rxd3	[25:22]	RW	4'h8	driving selector for rgmii0_rxd3
pin_mux_selector_rgmii0_rxd3	[21:20]	RW	2'h0	pin mux selector for rgmii0_rxd3
pull_down_enable_rgmii0_rxd3	[19]	RW	1'h0	pull down enable for rgmii0_rxd3
pull_up_enable_rgmii0_rxd3	[18]	RW	1'h0	pull up enable for rgmii0_rxd3
reserved	[17:16]	RW	2'h0	reserved
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_rgmii0_rxd2	[11]	RW	1'h1	pad oex enable for rgmii0_rxd2. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_rgmii0_rxd2	[10]	RW	1'h0	schmitt trigger enable for rgmii0_rxd2. '1' enables schmitt trigger input function
driving_selector_rgmii0_rxd2	[9:6]	RW	4'h8	driving selector for rgmii0_rxd2
pin_mux_selector_rgmii0_rxd2	[5:4]	RW	2'h0	pin mux selector for rgmii0_rxd2
pull_down_enable_rgmii0_rxd2	[3]	RW	1'h0	pull down enable for rgmii0_rxd2
pull_up_enable_rgmii0_rxd2	[2]	RW	1'h0	pull up enable for rgmii0_rxd2
reserved	[1:0]	RW	2'h0	reserved

8.5.2.16 pin mux and io control register for rgmii0_rxctrl and rgmii0_txc (0x3C)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_rgmii0_txc	[27]	RW	1'h1	pad oex enable for rgmii0_txc. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_rgmii0_txc	[26]	RW	1'h0	schmitt trigger enable for rgmii0_txc. '1' enables schmitt trigger input function
driving_selector_rgmii0_txc	[25:22]	RW	4'h8	driving selector for rgmii0_txc
pin_mux_selector_rgmii0_txc	[21:20]	RW	2'h0	pin mux selector for rgmii0_txc
pull_down_enable_rgmii0_txc	[19]	RW	1'h0	pull down enable for rgmii0_txc
pull_up_enable_rgmii0_txc	[18]	RW	1'h0	pull up enable for rgmii0_txc
reserved	[17:16]	RW	2'h0	reserved
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_rgmii0_rxctrl	[11]	RW	1'h1	pad oex enable for rgmii0_rxctrl. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_rgmii0_rxctrl	[10]	RW	1'h0	schmitt trigger enable for rgmii0_rxctrl. '1' enables schmitt trigger input function
driving_selector_rgmii0_rxctrl	[9:6]	RW	4'h8	driving selector for rgmii0_rxctrl
pin_mux_selector_rgmii0_rxctrl	[5:4]	RW	2'h0	pin mux selector for rgmii0_rxctrl
pull_down_enable_rgmii0_rxctrl	[3]	RW	1'h0	pull down enable for rgmii0_rxctrl
pull_up_enable_rgmii0_rxctrl	[2]	RW	1'h0	pull up enable for rgmii0_rxctrl
reserved	[1:0]	RW	2'h0	reserved

8.5.2.17 pin mux and io control register for rgmii0_rxc and rgmii0_refclko (0x40)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_rgmii0_refclko	[27]	RW	1'h1	pad oex enable for rgmii0_refclko. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_rgmii0_refclko	[26]	RW	1'h0	schmitt trigger enable for rgmii0_refclko. '1' enables schmitt trigger input function
driving_selector_rgmii0_refclko	[25:22]	RW	4'h8	driving selector for rgmii0_refclko
pin_mux_selector_rgmii0_refclko	[21:20]	RW	2'h0	pin mux selector for rgmii0_refclko
pull_down_enable_rgmii0_refclko	[19]	RW	1'h0	pull down enable for rgmii0_refclko
pull_up_enable_rgmii0_refclko	[18]	RW	1'h0	pull up enable for rgmii0_refclko
reserved	[17:16]	RW	2'h0	reserved
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_rgmii0_rxc	[11]	RW	1'h1	pad oex enable for rgmii0_rxc. '1' enables pad output mode under ip's drive
schmitt_trigger_ena	[10]	RW	1'h0	schmitt trigger enable for rgmii0_rxc.

ble_rgmii0_rxc				'1' enables schmitt trigger input function
driving_selector_rgmii0_rxc	[9:6]	RW	4'h8	driving selector for rgmii0_rxc
pin_mux_selector_rgmii0_rxc	[5:4]	RW	2'h0	pin mux selector for rgmii0_rxc
pull_down_enable_rgmii0_rxc	[3]	RW	1'h0	pull down enable for rgmii0_rxc
pull_up_enable_rgmii0_rxc	[2]	RW	1'h0	pull up enable for rgmii0_rxc
reserved	[1:0]	RW	2'h0	reserved

8.5.2.18 pin mux and io control register for rgmii0_irq and rgmii0_mdc (0x44)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_rgmii0_mdc	[27]	RW	1'h1	pad oex enable for rgmii0_mdc. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_rgmii0_mdc	[26]	RW	1'h0	schmitt trigger enable for rgmii0_mdc. '1' enables schmitt trigger input function
driving_selector_rgmii0_mdc	[25:22]	RW	4'h8	driving selector for rgmii0_mdc
pin_mux_selector_rgmii0_mdc	[21:20]	RW	2'h0	pin mux selector for rgmii0_mdc
pull_down_enable_rgmii0_mdc	[19]	RW	1'h0	pull down enable for rgmii0_mdc
pull_up_enable_rgmii0_mdc	[18]	RW	1'h0	pull up enable for rgmii0_mdc
reserved	[17:16]	RW	2'h0	reserved
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_rgmii0_irq	[11]	RW	1'h1	pad oex enable for rgmii0_irq. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_rgmii0_irq	[10]	RW	1'h0	schmitt trigger enable for rgmii0_irq. '1' enables schmitt trigger input function
driving_selector_rgmii0_irq	[9:6]	RW	4'h8	driving selector for rgmii0_irq
pin_mux_selector_rgmii0_irq	[5:4]	RW	2'h0	pin mux selector for rgmii0_irq
pull_down_enable_rgmii0_irq	[3]	RW	1'h0	pull down enable for rgmii0_irq
pull_up_enable_rgmii0_irq	[2]	RW	1'h0	pull up enable for rgmii0_irq
reserved	[1:0]	RW	2'h0	reserved

8.5.2.19 pin mux and io control register for rgmii0_mdio and rgmii1_txd0 (0x48)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_rgmii1_txd0	[27]	RW	1'h1	pad oex enable for rgmii1_txd0. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_rgmii1_txd0	[26]	RW	1'h0	schmitt trigger enable for rgmii1_txd0. '1' enables schmitt trigger input function
driving_selector_rg	[25:22]	RW	4'h8	driving selector for rgmii1_txd0

mii1_txd0				
pin_mux_selector_rgmii1_txd0	[21:20]	RW	2'h0	pin mux selector for rgmii1_txd0
pull_down_enable_rgmii1_txd0	[19]	RW	1'h0	pull down enable for rgmii1_txd0
pull_up_enable_rgmii1_txd0	[18]	RW	1'h0	pull up enable for rgmii1_txd0
reserved	[17:16]	RW	2'h0	reserved
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_rgmii0_mdio	[11]	RW	1'h1	pad oex enable for rgmii0_mdio. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_rgmii0_mdio	[10]	RW	1'h0	schmitt trigger enable for rgmii0_mdio. '1' enables schmitt trigger input function
driving_selector_rgmii0_mdio	[9:6]	RW	4'h8	driving selector for rgmii0_mdio
pin_mux_selector_rgmii0_mdio	[5:4]	RW	2'h0	pin mux selector for rgmii0_mdio
pull_down_enable_rgmii0_mdio	[3]	RW	1'h0	pull down enable for rgmii0_mdio
pull_up_enable_rgmii0_mdio	[2]	RW	1'h0	pull up enable for rgmii0_mdio
reserved	[1:0]	RW	2'h0	reserved

8.5.2.20 pin mux and io control register for rgmii1_txd1 and rgmii1_txd2 (0x4C)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_rgmii1_txd2	[27]	RW	1'h1	pad oex enable for rgmii1_txd2. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_rgmii1_txd2	[26]	RW	1'h0	schmitt trigger enable for rgmii1_txd2. '1' enables schmitt trigger input function
driving_selector_rgmii1_txd2	[25:22]	RW	4'h8	driving selector for rgmii1_txd2
pin_mux_selector_rgmii1_txd2	[21:20]	RW	2'h0	pin mux selector for rgmii1_txd2
pull_down_enable_rgmii1_txd2	[19]	RW	1'h0	pull down enable for rgmii1_txd2
pull_up_enable_rgmii1_txd2	[18]	RW	1'h0	pull up enable for rgmii1_txd2
reserved	[17:16]	RW	2'h0	reserved
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_rgmii1_txd1	[11]	RW	1'h1	pad oex enable for rgmii1_txd1. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_rgmii1_txd1	[10]	RW	1'h0	schmitt trigger enable for rgmii1_txd1. '1' enables schmitt trigger input function
driving_selector_rgmii1_txd1	[9:6]	RW	4'h8	driving selector for rgmii1_txd1
pin_mux_selector_rgmii1_txd1	[5:4]	RW	2'h0	pin mux selector for rgmii1_txd1
pull_down_enable_rgmii1_txd1	[3]	RW	1'h0	pull down enable for rgmii1_txd1
pull_up_enable_rgmii1_txd1	[2]	RW	1'h0	pull up enable for rgmii1_txd1
reserved	[1:0]	RW	2'h0	reserved

8.5.2.21 pin mux and io control register for rgmii1_txd3 and rgmii1_txctrl (0x50)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_rgmii1_txctrl	[27]	RW	1'h1	pad oex enable for rgmii1_txctrl. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_rgmii1_txctrl	[26]	RW	1'h0	schmitt trigger enable for rgmii1_txctrl. '1' enables schmitt trigger input function
driving_selector_rgmii1_txctrl	[25:22]	RW	4'h8	driving selector for rgmii1_txctrl
pin_mux_selector_rgmii1_txctrl	[21:20]	RW	2'h0	pin mux selector for rgmii1_txctrl
pull_down_enable_rgmii1_txctrl	[19]	RW	1'h0	pull down enable for rgmii1_txctrl
pull_up_enable_rgmii1_txctrl	[18]	RW	1'h0	pull up enable for rgmii1_txctrl
reserved	[17:16]	RW	2'h0	reserved
reserved	[15:12}	RW	4'h0	reserved
pad_oex_enable_rgmii1_txd3	[11]	RW	1'h1	pad oex enable for rgmii1_txd3. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_rgmii1_txd3	[10]	RW	1'h0	schmitt trigger enable for rgmii1_txd3. '1' enables schmitt trigger input function
driving_selector_rgmii1_txd3	[9:6]	RW	4'h8	driving selector for rgmii1_txd3
pin_mux_selector_rgmii1_txd3	[5:4]	RW	2'h0	pin mux selector for rgmii1_txd3
pull_down_enable_rgmii1_txd3	[3]	RW	1'h0	pull down enable for rgmii1_txd3
pull_up_enable_rgmii1_txd3	[2]	RW	1'h0	pull up enable for rgmii1_txd3
reserved	[1:0]	RW	2'h0	reserved

8.5.2.22 pin mux and io control register for rgmii1_rxd0 and rgmii1_rxd1 (0x54)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_rgmii1_rxd1	[27]	RW	1'h1	pad oex enable for rgmii1_rxd1. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_rgmii1_rxd1	[26]	RW	1'h0	schmitt trigger enable for rgmii1_rxd1. '1' enables schmitt trigger input function
driving_selector_rgmii1_rxd1	[25:22]	RW	4'h8	driving selector for rgmii1_rxd1
pin_mux_selector_rgmii1_rxd1	[21:20]	RW	2'h0	pin mux selector for rgmii1_rxd1
pull_down_enable_rgmii1_rxd1	[19]	RW	1'h0	pull down enable for rgmii1_rxd1
pull_up_enable_rgmii1_rxd1	[18]	RW	1'h0	pull up enable for rgmii1_rxd1
reserved	[17:16]	RW	2'h0	reserved
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_rgmii1_rxd0	[11]	RW	1'h1	pad oex enable for rgmii1_rxd0. '1' enables pad output mode under ip's drive
schmitt_trigger_ena	[10]	RW	1'h0	schmitt trigger enable for rgmii1_rxd0.

ble_rgmii1_rxd0				'1' enables schmitt trigger input function
driving_selector_rgmii1_rxd0	[9:6]	RW	4'h8	driving selector for rgmii1_rxd0
pin_mux_selector_rgmii1_rxd0	[5:4]	RW	2'h0	pin mux selector for rgmii1_rxd0
pull_down_enable_rgmii1_rxd0	[3]	RW	1'h0	pull down enable for rgmii1_rxd0
pull_up_enable_rgmii1_rxd0	[2]	RW	1'h0	pull up enable for rgmii1_rxd0
reserved	[1:0]	RW	2'h0	reserved

8.5.2.23 pin mux and io control register for rgmii1_rxd2 and rgmii1_rxd3 (0x58)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_rgmii1_rxd3	[27]	RW	1'h1	pad oex enable for rgmii1_rxd3. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_rgmii1_rxd3	[26]	RW	1'h0	schmitt trigger enable for rgmii1_rxd3. '1' enables schmitt trigger input function
driving_selector_rgmii1_rxd3	[25:22]	RW	4'h8	driving selector for rgmii1_rxd3
pin_mux_selector_rgmii1_rxd3	[21:20]	RW	2'h0	pin mux selector for rgmii1_rxd3
pull_down_enable_rgmii1_rxd3	[19]	RW	1'h0	pull down enable for rgmii1_rxd3
pull_up_enable_rgmii1_rxd3	[18]	RW	1'h0	pull up enable for rgmii1_rxd3
reserved	[17:16]	RW	2'h0	reserved
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_rgmii1_rxd2	[11]	RW	1'h1	pad oex enable for rgmii1_rxd2. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_rgmii1_rxd2	[10]	RW	1'h0	schmitt trigger enable for rgmii1_rxd2. '1' enables schmitt trigger input function
driving_selector_rgmii1_rxd2	[9:6]	RW	4'h8	driving selector for rgmii1_rxd2
pin_mux_selector_rgmii1_rxd2	[5:4]	RW	2'h0	pin mux selector for rgmii1_rxd2
pull_down_enable_rgmii1_rxd2	[3]	RW	1'h0	pull down enable for rgmii1_rxd2
pull_up_enable_rgmii1_rxd2	[2]	RW	1'h0	pull up enable for rgmii1_rxd2
reserved	[1:0]	RW	2'h0	reserved

8.5.2.24 pin mux and io control register for rgmii1_rxctrl and rgmii1_txc (0x5C)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_rgmii1_txc	[27]	RW	1'h1	pad oex enable for rgmii1_txc. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_rgmii1_txc	[26]	RW	1'h0	schmitt trigger enable for rgmii1_txc. '1' enables schmitt trigger input function
driving_selector_rg	[25:22]	RW	4'h8	driving selector for rgmii1_txc

mii1_txc				
pin_mux_selector_rgmii1_txc	[21:20]	RW	2'h0	pin mux selector for rgmii1_txc
pull_down_enable_rgmii1_txc	[19]	RW	1'h0	pull down enable for rgmii1_txc
pull_up_enable_rgmii1_txc	[18]	RW	1'h0	pull up enable for rgmii1_txc
reserved	[17:16]	RW	2'h0	reserved
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_rgmii1_rxctrl	[11]	RW	1'h1	pad oex enable for rgmii1_rxctrl. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_rgmii1_rxctrl	[10]	RW	1'h0	schmitt trigger enable for rgmii1_rxctrl. '1' enables schmitt trigger input function
driving_selector_rgmii1_rxctrl	[9:6]	RW	4'h8	driving selector for rgmii1_rxctrl
pin_mux_selector_rgmii1_rxctrl	[5:4]	RW	2'h0	pin mux selector for rgmii1_rxctrl
pull_down_enable_rgmii1_rxctrl	[3]	RW	1'h0	pull down enable for rgmii1_rxctrl
pull_up_enable_rgmii1_rxctrl	[2]	RW	1'h0	pull up enable for rgmii1_rxctrl
reserved	[1:0]	RW	2'h0	reserved

8.5.2.25 pin mux and io control register for rgmii1_rxc and rgmii1_refclko (0x60)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_rgmii1_refclko	[27]	RW	1'h1	pad oex enable for rgmii1_refclko. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_rgmii1_refclko	[26]	RW	1'h0	schmitt trigger enable for rgmii1_refclko. '1' enables schmitt trigger input function
driving_selector_rgmii1_refclko	[25:22]	RW	4'h8	driving selector for rgmii1_refclko
pin_mux_selector_rgmii1_refclko	[21:20]	RW	2'h0	pin mux selector for rgmii1_refclko
pull_down_enable_rgmii1_refclko	[19]	RW	1'h0	pull down enable for rgmii1_refclko
pull_up_enable_rgmii1_refclko	[18]	RW	1'h0	pull up enable for rgmii1_refclko
reserved	[17:16]	RW	2'h0	reserved
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_rgmii1_rxc	[11]	RW	1'h1	pad oex enable for rgmii1_rxc. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_rgmii1_rxc	[10]	RW	1'h0	schmitt trigger enable for rgmii1_rxc. '1' enables schmitt trigger input function
driving_selector_rgmii1_rxc	[9:6]	RW	4'h8	driving selector for rgmii1_rxc
pin_mux_selector_rgmii1_rxc	[5:4]	RW	2'h0	pin mux selector for rgmii1_rxc
pull_down_enable_rgmii1_rxc	[3]	RW	1'h0	pull down enable for rgmii1_rxc
pull_up_enable_rgmii1_rxc	[2]	RW	1'h0	pull up enable for rgmii1_rxc
reserved	[1:0]	RW	2'h0	reserved

8.5.2.26 pin mux and io control register for rgmii1_irq and rgmii1_mdc (0x64)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_rgmii1_mdc	[27]	RW	1'h1	pad oex enable for rgmii1_mdc. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_rgmii1_mdc	[26]	RW	1'h0	schmitt trigger enable for rgmii1_mdc. '1' enables schmitt trigger input function
driving_selector_rgmii1_mdc	[25:22]	RW	4'h8	driving selector for rgmii1_mdc
pin_mux_selector_rgmii1_mdc	[21:20]	RW	2'h0	pin mux selector for rgmii1_mdc
pull_down_enable_rgmii1_mdc	[19]	RW	1'h0	pull down enable for rgmii1_mdc
pull_up_enable_rgmii1_mdc	[18]	RW	1'h0	pull up enable for rgmii1_mdc
reserved	[17:16]	RW	2'h0	reserved
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_rgmii1_irq	[11]	RW	1'h1	pad oex enable for rgmii1_irq. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_rgmii1_irq	[10]	RW	1'h0	schmitt trigger enable for rgmii1_irq. '1' enables schmitt trigger input function
driving_selector_rgmii1_irq	[9:6]	RW	4'h8	driving selector for rgmii1_irq
pin_mux_selector_rgmii1_irq	[5:4]	RW	2'h0	pin mux selector for rgmii1_irq
pull_down_enable_rgmii1_irq	[3]	RW	1'h0	pull down enable for rgmii1_irq
pull_up_enable_rgmii1_irq	[2]	RW	1'h0	pull up enable for rgmii1_irq
reserved	[1:0]	RW	2'h0	reserved

8.5.2.27 pin mux and io control register for rgmii1_mdio and pwm0 (0x68)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_pwm0	[27]	RW	1'h1	pad oex enable for pwm0. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_pwm0	[26]	RW	1'h0	schmitt trigger enable for pwm0. '1' enables schmitt trigger input function
driving_selector_pwm0	[25:22]	RW	4'h8	driving selector for pwm0
pin_mux_selector_pwm0	[21:20]	RW	2'h0	pin mux selector for pwm0
reserved	[19:18]	RW	2'h0	reserved
pull_selector_pwm0	[17]	RW	1'h0	pull selector for pwm0.
pull_enable_pwm0	[16]	RW	1'h0	pull enable for pwm0.
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_rgmii1_mdio	[11]	RW	1'h1	pad oex enable for rgmii1_mdio. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_rgmii1_mdio	[10]	RW	1'h0	schmitt trigger enable for rgmii1_mdio. '1' enables schmitt trigger input function
driving_selector_rg	[9:6]	RW	4'h8	driving selector for rgmii1_mdio

mii1_mdio				
pin_mux_selector_rgmii1_mdio	[5:4]	RW	2'h0	pin mux selector for rgmii1_mdio
pull_down_enable_rgmii1_mdio	[3]	RW	1'h0	pull down enable for rgmii1_mdio
pull_up_enable_rgmii1_mdio	[2]	RW	1'h0	pull up enable for rgmii1_mdio
reserved	[1:0]	RW	2'h0	reserved

8.5.2.28 pin mux and io control register for pwm1 and fan0 (0x6C)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_fan0	[27]	RW	1'h1	pad oex enable for fan0. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_fan0	[26]	RW	1'h1	schmitt trigger enable for fan0. '1' enables schmitt trigger input function
driving_selector_fan0	[25:22]	RW	4'h8	driving selector for fan0
pin_mux_selector_fan0	[21:20]	RW	2'h0	pin mux selector for fan0
reserved	[19:18]	RW	2'h0	reserved
pull_selector_fan0	[17]	RW	1'h0	pull selector for fan0.
pull_enable_fan0	[16]	RW	1'h0	pull enable for fan0.
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_pwm1	[11]	RW	1'h1	pad oex enable for pwm1. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_pwm1	[10]	RW	1'h0	schmitt trigger enable for pwm1. '1' enables schmitt trigger input function
driving_selector_pwm1	[9:6]	RW	4'h8	driving selector for pwm1
pin_mux_selector_pwm1	[5:4]	RW	2'h0	pin mux selector for pwm1
reserved	[3:2]	RW	2'h0	reserved
pull_selector_pwm1	[1]	RW	1'h0	pull selector for pwm1.
pull_enable_pwm1	[0]	RW	1'h0	pull enable for pwm1.

8.5.2.29 pin mux and io control register for fan1 and iic0_sda (0x70)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_iic0_sda	[27]	RW	1'h1	pad oex enable for iic0_sda. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_iic0_sda	[26]	RW	1'h1	schmitt trigger enable for iic0_sda. '1' enables schmitt trigger input function
driving_selector_iic0_sda	[25:22]	RW	4'h8	driving selector for iic0_sda
pin_mux_selector_iic0_sda	[21:20]	RW	2'h0	pin mux selector for iic0_sda
pull_down_enable_iic0_sda	[19]	RW	1'h0	pull down enable for iic0_sda

pull_up_enable_iic0_sda	[18]	RW	1'h0	pull up enable for iic0_sda
reserved	[17:16]	RW	2'h0	reserved
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_fan1	[11]	RW	1'h1	pad oex enable for fan1. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_fan1	[10]	RW	1'h1	schmitt trigger enable for fan1. '1' enables schmitt trigger input function
driving_selector_fan1	[9:6]	RW	4'h8	driving selector for fan1
pin_mux_selector_fan1	[5:4]	RW	2'h0	pin mux selector for fan1
reserved	[3:2]	RW	2'h0	reserved
pull_selector_fan1	[1]	RW	1'h0	pull selector for fan1.
pull_enable_fan1	[0]	RW	1'h0	pull enable for fan1.

8.5.2.30 pin mux and io control register for iic0_scl and iic1_sda (0x74)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_iic1_sda	[27]	RW	1'h1	pad oex enable for iic1_sda. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_iic1_sda	[26]	RW	1'h1	schmitt trigger enable for iic1_sda. '1' enables schmitt trigger input function
driving_selector_iic1_sda	[25:22]	RW	4'h8	driving selector for iic1_sda
pin_mux_selector_iic1_sda	[21:20]	RW	2'h0	pin mux selector for iic1_sda
pull_down_enable_iic1_sda	[19]	RW	1'h0	pull down enable for iic1_sda
pull_up_enable_iic1_sda	[18]	RW	1'h0	pull up enable for iic1_sda
reserved	[17:16]	RW	2'h0	reserved
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_iic0_scl	[11]	RW	1'h1	pad oex enable for iic0_scl. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_iic0_scl	[10]	RW	1'h1	schmitt trigger enable for iic0_scl. '1' enables schmitt trigger input function
driving_selector_iic0_scl	[9:6]	RW	4'h8	driving selector for iic0_scl
pin_mux_selector_iic0_scl	[5:4]	RW	2'h0	pin mux selector for iic0_scl
pull_down_enable_iic0_scl	[3]	RW	1'h0	pull down enable for iic0_scl
pull_up_enable_iic0_scl	[2]	RW	1'h0	pull up enable for iic0_scl
reserved	[1:0]	RW	2'h0	reserved

8.5.2.31 pin mux and io control register for iic1_scl and iic2_sda (0x78)

Field Name	Bit	Type	Reset Value	Field Description

reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_iic2_sda	[27]	RW	1'h1	pad oex enable for iic2_sda. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_iic2_sda	[26]	RW	1'h1	schmitt trigger enable for iic2_sda. '1' enables schmitt trigger input function
driving_selector_iic2_sda	[25:22]	RW	4'h8	driving selector for iic2_sda
pin_mux_selector_iic2_sda	[21:20]	RW	2'h0	pin mux selector for iic2_sda
pull_down_enable_iic2_sda	[19]	RW	1'h0	pull down enable for iic2_sda
pull_up_enable_iic2_sda	[18]	RW	1'h0	pull up enable for iic2_sda
reserved	[17:16]	RW	2'h0	reserved
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_iic1_scl	[11]	RW	1'h1	pad oex enable for iic1_scl. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_iic1_scl	[10]	RW	1'h1	schmitt trigger enable for iic1_scl. '1' enables schmitt trigger input function
driving_selector_iic1_scl	[9:6]	RW	4'h8	driving selector for iic1_scl
pin_mux_selector_iic1_scl	[5:4]	RW	2'h0	pin mux selector for iic1_scl
pull_down_enable_iic1_scl	[3]	RW	1'h0	pull down enable for iic1_scl
pull_up_enable_iic1_scl	[2]	RW	1'h0	pull up enable for iic1_scl
reserved	[1:0]	RW	2'h0	reserved

8.5.2.32 pin mux and io control register for iic2_scl and uart0_tx (0x7C)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_uart0_tx	[27]	RW	1'h1	pad oex enable for uart0_tx. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_uart0_tx	[26]	RW	1'h1	schmitt trigger enable for uart0_tx. '1' enables schmitt trigger input function
driving_selector_uart0_tx	[25:22]	RW	4'h8	driving selector for uart0_tx
pin_mux_selector_uart0_tx	[21:20]	RW	2'h0	pin mux selector for uart0_tx
reserved	[19:18]	RW	2'h0	reserved
pull_selector_uart0_tx	[17]	RW	1'h1	pull selector for uart0_tx.
pull_enable_uart0_tx	[16]	RW	1'h1	pull enable for uart0_tx.
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_iic2_scl	[11]	RW	1'h1	pad oex enable for iic2_scl. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_iic2_scl	[10]	RW	1'h1	schmitt trigger enable for iic2_scl. '1' enables schmitt trigger input function
driving_selector_iic2_scl	[9:6]	RW	4'h8	driving selector for iic2_scl
pin_mux_selector_iic2_scl	[5:4]	RW	2'h0	pin mux selector for iic2_scl

pull_down_enable_iic2_scl	[3]	RW	1'h0	pull down enable for iic2_scl
pull_up_enable_iic2_scl	[2]	RW	1'h0	pull up enable for iic2_scl
reserved	[1:0]	RW	2'h0	reserved

8.5.2.33 pin mux and io control register for uart0_rx and uart1_tx (0x80)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_uart1_tx	[27]	RW	1'h1	pad oex enable for uart1_tx. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_uart1_tx	[26]	RW	1'h1	schmitt trigger enable for uart1_tx. '1' enables schmitt trigger input function
driving_selector_uart1_tx	[25:22]	RW	4'h8	driving selector for uart1_tx
pin_mux_selector_uart1_tx	[21:20]	RW	2'h0	pin mux selector for uart1_tx
reserved	[19:18]	RW	2'h0	reserved
pull_selector_uart1_tx	[17]	RW	1'h1	pull selector for uart1_tx.
pull_enable_uart1_tx	[16]	RW	1'h1	pull enable for uart1_tx.
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_uart0_rx	[11]	RW	1'h1	pad oex enable for uart0_rx. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_uart0_rx	[10]	RW	1'h1	schmitt trigger enable for uart0_rx. '1' enables schmitt trigger input function
driving_selector_uart0_rx	[9:6]	RW	4'h8	driving selector for uart0_rx
pin_mux_selector_uart0_rx	[5:4]	RW	2'h0	pin mux selector for uart0_rx
reserved	[3:2]	RW	2'h0	reserved
pull_selector_uart0_rx	[1]	RW	1'h1	pull selector for uart0_rx.
pull_enable_uart0_rx	[0]	RW	1'h1	pull enable for uart0_rx.

8.5.2.34 pin mux and io control register for uart1_rx and uart2_tx (0x84)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_uart2_tx	[27]	RW	1'h1	pad oex enable for uart2_tx. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_uart2_tx	[26]	RW	1'h1	schmitt trigger enable for uart2_tx. '1' enables schmitt trigger input function
driving_selector_uart2_tx	[25:22]	RW	4'h8	driving selector for uart2_tx
pin_mux_selector_uart2_tx	[21:20]	RW	2'h0	pin mux selector for uart2_tx
reserved	[19:18]	RW	2'h0	reserved

pull_selector_uart2_tx	[17]	RW	1'h1	pull selector for uart2_tx.
pull_enable_uart2_tx	[16]	RW	1'h1	pull enable for uart2_tx.
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_uart1_rx	[11]	RW	1'h1	pad oex enable for uart1_rx. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_uart1_rx	[10]	RW	1'h1	schmitt trigger enable for uart1_rx. '1' enables schmitt trigger input function
driving_selector_uart1_rx	[9:6]	RW	4'h8	driving selector for uart1_rx
pin_mux_selector_uart1_rx	[5:4]	RW	2'h0	pin mux selector for uart1_rx
reserved	[3:2]	RW	2'h0	reserved
pull_selector_uart1_rx	[1]	RW	1'h1	pull selector for uart1_rx.
pull_enable_uart1_rx	[0]	RW	1'h1	pull enable for uart1_rx.

8.5.2.35 pin mux and io control register for uart2_rx and gpio0 (0x88)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_gpio0	[27]	RW	1'h1	pad oex enable for gpio0. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_gpio0	[26]	RW	1'h1	schmitt trigger enable for gpio0. '1' enables schmitt trigger input function
driving_selector_gpio0	[25:22]	RW	4'h8	driving selector for gpio0
pin_mux_selector_gpio0	[21:20]	RW	2'h0	pin mux selector for gpio0
reserved	[19:18]	RW	2'h0	reserved
pull_selector_gpio0	[17]	RW	1'h0	pull selector for gpio0.
pull_enable_gpio0	[16]	RW	1'h1	pull enable for gpio0.
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_uart2_rx	[11]	RW	1'h1	pad oex enable for uart2_rx. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_uart2_rx	[10]	RW	1'h1	schmitt trigger enable for uart2_rx. '1' enables schmitt trigger input function
driving_selector_uart2_rx	[9:6]	RW	4'h8	driving selector for uart2_rx
pin_mux_selector_uart2_rx	[5:4]	RW	2'h0	pin mux selector for uart2_rx
reserved	[3:2]	RW	2'h0	reserved
pull_selector_uart2_rx	[1]	RW	1'h1	pull selector for uart2_rx.
pull_enable_uart2_rx	[0]	RW	1'h1	pull enable for uart2_rx.

8.5.2.36 pin mux and io control register for gpio1 and gpio2 (0x8C)

Field Name	Bit	Type	Reset	Field Description

			Value	
reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_gpio2	[27]	RW	1'h1	pad oex enable for gpio2. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_gpio2	[26]	RW	1'h1	schmitt trigger enable for gpio2. '1' enables schmitt trigger input function
driving_selector_gpio2	[25:22]	RW	4'h8	driving selector for gpio2
pin_mux_selector_gpio2	[21:20]	RW	2'h0	pin mux selector for gpio2
reserved	[19:18]	RW	2'h0	reserved
pull_selector_gpio2	[17]	RW	1'h0	pull selector for gpio2.
pull_enable_gpio2	[16]	RW	1'h1	pull enable for gpio2.
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_gpio1	[11]	RW	1'h1	pad oex enable for gpio1. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_gpio1	[10]	RW	1'h1	schmitt trigger enable for gpio1. '1' enables schmitt trigger input function
driving_selector_gpio1	[9:6]	RW	4'h8	driving selector for gpio1
pin_mux_selector_gpio1	[5:4]	RW	2'h0	pin mux selector for gpio1
reserved	[3:2]	RW	2'h0	reserved
pull_selector_gpio1	[1]	RW	1'h0	pull selector for gpio1.
pull_enable_gpio1	[0]	RW	1'h1	pull enable for gpio1.

8.5.2.37 pin mux and io control register for gpio3 and gpio4 (0x90)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_gpio4	[27]	RW	1'h1	pad oex enable for gpio4. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_gpio4	[26]	RW	1'h1	schmitt trigger enable for gpio4. '1' enables schmitt trigger input function
driving_selector_gpio4	[25:22]	RW	4'h8	driving selector for gpio4
pin_mux_selector_gpio4	[21:20]	RW	2'h0	pin mux selector for gpio4
reserved	[19:18]	RW	2'h0	reserved
pull_selector_gpio4	[17]	RW	1'h0	pull selector for gpio4.
pull_enable_gpio4	[16]	RW	1'h1	pull enable for gpio4.
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_gpio3	[11]	RW	1'h1	pad oex enable for gpio3. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_gpio3	[10]	RW	1'h1	schmitt trigger enable for gpio3. '1' enables schmitt trigger input function
driving_selector_gpio3	[9:6]	RW	4'h8	driving selector for gpio3
pin_mux_selector_gpio3	[5:4]	RW	2'h0	pin mux selector for gpio3
reserved	[3:2]	RW	2'h0	reserved
pull_selector_gpio3	[1]	RW	1'h0	pull selector for gpio3.
pull_enable_gpio3	[0]	RW	1'h1	pull enable for gpio3.

8.5.2.38 pin mux and io control register for gpio5 and gpio6 (0x94)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_gpio6	[27]	RW	1'h1	pad oex enable for gpio6. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_gpio6	[26]	RW	1'h1	schmitt trigger enable for gpio6. '1' enables schmitt trigger input function
driving_selector_gpio6	[25:22]	RW	4'h8	driving selector for gpio6
pin_mux_selector_gpio6	[21:20]	RW	2'h0	pin mux selector for gpio6
reserved	[19:18]	RW	2'h0	reserved
pull_selector_gpio6	[17]	RW	1'h0	pull selector for gpio6.
pull_enable_gpio6	[16]	RW	1'h1	pull enable for gpio6.
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_gpio5	[11]	RW	1'h1	pad oex enable for gpio5. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_gpio5	[10]	RW	1'h1	schmitt trigger enable for gpio5. '1' enables schmitt trigger input function
driving_selector_gpio5	[9:6]	RW	4'h8	driving selector for gpio5
pin_mux_selector_gpio5	[5:4]	RW	2'h0	pin mux selector for gpio5
reserved	[3:2]	RW	2'h0	reserved
pull_selector_gpio5	[1]	RW	1'h0	pull selector for gpio5.
pull_enable_gpio5	[0]	RW	1'h1	pull enable for gpio5.

8.5.2.39 pin mux and io control register for gpio7 and gpio8 (0x98)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_gpio8	[27]	RW	1'h1	pad oex enable for gpio8. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_gpio8	[26]	RW	1'h1	schmitt trigger enable for gpio8. '1' enables schmitt trigger input function
driving_selector_gpio8	[25:22]	RW	4'h8	driving selector for gpio8
pin_mux_selector_gpio8	[21:20]	RW	2'h0	pin mux selector for gpio8
reserved	[19:18]	RW	2'h0	reserved
pull_selector_gpio8	[17]	RW	1'h0	pull selector for gpio8.
pull_enable_gpio8	[16]	RW	1'h1	pull enable for gpio8.
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_gpio7	[11]	RW	1'h1	pad oex enable for gpio7. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_gpio7	[10]	RW	1'h1	schmitt trigger enable for gpio7. '1' enables schmitt trigger input function
driving_selector_gpio7	[9:6]	RW	4'h8	driving selector for gpio7

pin_mux_selector_gpio7	[5:4]	RW	2'h0	pin mux selector for gpio7
reserved	[3:2]	RW	2'h0	reserved
pull_selector_gpio7	[1]	RW	1'h0	pull selector for gpio7.
pull_enable_gpio7	[0]	RW	1'h1	pull enable for gpio7.

8.5.2.40 pin mux and io control register for gpio9 and gpio10 (0x9C)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_gpio10	[27]	RW	1'h1	pad oex enable for gpio10. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_gpio10	[26]	RW	1'h1	schmitt trigger enable for gpio10. '1' enables schmitt trigger input function
driving_selector_gpio10	[25:22]	RW	4'h8	driving selector for gpio10
pin_mux_selector_gpio10	[21:20]	RW	2'h0	pin mux selector for gpio10
reserved	[19:18]	RW	2'h0	reserved
pull_selector_gpio10	[17]	RW	1'h0	pull selector for gpio10.
pull_enable_gpio10	[16]	RW	1'h1	pull enable for gpio10.
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_gpio9	[11]	RW	1'h1	pad oex enable for gpio9. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_gpio9	[10]	RW	1'h1	schmitt trigger enable for gpio9. '1' enables schmitt trigger input function
driving_selector_gpio9	[9:6]	RW	4'h8	driving selector for gpio9
pin_mux_selector_gpio9	[5:4]	RW	2'h0	pin mux selector for gpio9
reserved	[3:2]	RW	2'h0	reserved
pull_selector_gpio9	[1]	RW	1'h0	pull selector for gpio9.
pull_enable_gpio9	[0]	RW	1'h1	pull enable for gpio9.

8.5.2.41 pin mux and io control register for gpio11 and gpio12 (0xA0)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_gpio12	[27]	RW	1'h1	pad oex enable for gpio12. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_gpio12	[26]	RW	1'h1	schmitt trigger enable for gpio12. '1' enables schmitt trigger input function
driving_selector_gpio12	[25:22]	RW	4'h8	driving selector for gpio12
pin_mux_selector_gpio12	[21:20]	RW	2'h0	pin mux selector for gpio12
reserved	[19:18]	RW	2'h0	reserved
pull_selector_gpio12	[17]	RW	1'h0	pull selector for gpio12.
pull_enable_gpio12	[16]	RW	1'h1	pull enable for gpio12.

reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_gpio11	[11]	RW	1'h1	pad oex enable for gpio11. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_gpio11	[10]	RW	1'h1	schmitt trigger enable for gpio11. '1' enables schmitt trigger input function
driving_selector_gpio11	[9:6]	RW	4'h8	driving selector for gpio11
pin_mux_selector_gpio11	[5:4]	RW	2'h0	pin mux selector for gpio11
reserved	[3:2]	RW	2'h0	reserved
pull_selector_gpio11	[1]	RW	1'h0	pull selector for gpio11.
pull_enable_gpio11	[0]	RW	1'h1	pull enable for gpio11.

8.5.2.42 pin mux and io control register for gpio13 and gpio14 (0xA4)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_gpio14	[27]	RW	1'h1	pad oex enable for gpio14. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_gpio14	[26]	RW	1'h1	schmitt trigger enable for gpio14. '1' enables schmitt trigger input function
driving_selector_gpio14	[25:22]	RW	4'h8	driving selector for gpio14
pin_mux_selector_gpio14	[21:20]	RW	2'h0	pin mux selector for gpio14
reserved	[19:18]	RW	2'h0	reserved
pull_selector_gpio14	[17]	RW	1'h0	pull selector for gpio14.
pull_enable_gpio14	[16]	RW	1'h1	pull enable for gpio14.
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_gpio13	[11]	RW	1'h1	pad oex enable for gpio13. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_gpio13	[10]	RW	1'h1	schmitt trigger enable for gpio13. '1' enables schmitt trigger input function
driving_selector_gpio13	[9:6]	RW	4'h8	driving selector for gpio13
pin_mux_selector_gpio13	[5:4]	RW	2'h0	pin mux selector for gpio13
reserved	[3:2]	RW	2'h0	reserved
pull_selector_gpio13	[1]	RW	1'h0	pull selector for gpio13.
pull_enable_gpio13	[0]	RW	1'h1	pull enable for gpio13.

8.5.2.43 pin mux and io control register for gpio15 and gpio16 (0xA8)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_gpio16	[27]	RW	1'h1	pad oex enable for gpio16. '1' enables pad output mode under ip's drive
schmitt_trigger_ena	[26]	RW	1'h1	schmitt trigger enable for gpio16.

ble_gpio16				'1' enables schmitt trigger input function
driving_selector_gpio16	[25:22]	RW	4'h8	driving selector for gpio16
pin_mux_selector_gpio16	[21:20]	RW	2'h0	pin mux selector for gpio16
reserved	[19:18]	RW	2'h0	reserved
pull_selector_gpio16	[17]	RW	1'h0	pull selector for gpio16.
pull_enable_gpio16	[16]	RW	1'h1	pull enable for gpio16.
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_gpio15	[11]	RW	1'h1	pad oex enable for gpio15. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_gpio15	[10]	RW	1'h1	schmitt trigger enable for gpio15. '1' enables schmitt trigger input function
driving_selector_gpio15	[9:6]	RW	4'h8	driving selector for gpio15
pin_mux_selector_gpio15	[5:4]	RW	2'h0	pin mux selector for gpio15
reserved	[3:2]	RW	2'h0	reserved
pull_selector_gpio15	[1]	RW	1'h0	pull selector for gpio15.
pull_enable_gpio15	[0]	RW	1'h1	pull enable for gpio15.

8.5.2.44 pin mux and io control register for gpio17 and gpio18 (0xAC)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_gpio18	[27]	RW	1'h1	pad oex enable for gpio18. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_gpio18	[26]	RW	1'h1	schmitt trigger enable for gpio18. '1' enables schmitt trigger input function
driving_selector_gpio18	[25:22]	RW	4'h8	driving selector for gpio18
pin_mux_selector_gpio18	[21:20]	RW	2'h0	pin mux selector for gpio18
reserved	[19:18]	RW	2'h0	reserved
pull_selector_gpio18	[17]	RW	1'h0	pull selector for gpio18.
pull_enable_gpio18	[16]	RW	1'h1	pull enable for gpio18.
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_gpio17	[11]	RW	1'h1	pad oex enable for gpio17. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_gpio17	[10]	RW	1'h1	schmitt trigger enable for gpio17. '1' enables schmitt trigger input function
driving_selector_gpio17	[9:6]	RW	4'h8	driving selector for gpio17
pin_mux_selector_gpio17	[5:4]	RW	2'h0	pin mux selector for gpio17
reserved	[3:2]	RW	2'h0	reserved
pull_selector_gpio17	[1]	RW	1'h0	pull selector for gpio17.
pull_enable_gpio17	[0]	RW	1'h1	pull enable for gpio17.

8.5.2.45 pin mux and io control register for gpio19 and gpio20 (0xB0)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_gpio20	[27]	RW	1'h1	pad oex enable for gpio20. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_gpio20	[26]	RW	1'h1	schmitt trigger enable for gpio20. '1' enables schmitt trigger input function
driving_selector_gpio20	[25:22]	RW	4'h8	driving selector for gpio20
pin_mux_selector_gpio20	[21:20]	RW	2'h0	pin mux selector for gpio20
reserved	[19:18]	RW	2'h0	reserved
pull_selector_gpio20	[17]	RW	1'h0	pull selector for gpio20.
pull_enable_gpio20	[16]	RW	1'h1	pull enable for gpio20.
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_gpio19	[11]	RW	1'h1	pad oex enable for gpio19. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_gpio19	[10]	RW	1'h0	schmitt trigger enable for gpio19. '1' enables schmitt trigger input function
driving_selector_gpio19	[9:6]	RW	4'h8	driving selector for gpio19
pin_mux_selector_gpio19	[5:4]	RW	2'h0	pin mux selector for gpio19
reserved	[3:2]	RW	2'h0	reserved
pull_selector_gpio19	[1]	RW	1'h0	pull selector for gpio19.
pull_enable_gpio19	[0]	RW	1'h1	pull enable for gpio19.

8.5.2.46 pin mux and io control register for gpio21 and gpio22 (0xB4)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_gpio22	[27]	RW	1'h1	pad oex enable for gpio22. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_gpio22	[26]	RW	1'h1	schmitt trigger enable for gpio22. '1' enables schmitt trigger input function
driving_selector_gpio22	[25:22]	RW	4'h8	driving selector for gpio22
pin_mux_selector_gpio22	[21:20]	RW	2'h0	pin mux selector for gpio22
reserved	[19:18]	RW	2'h0	reserved
pull_selector_gpio22	[17]	RW	1'h1	pull selector for gpio22.
pull_enable_gpio22	[16]	RW	1'h1	pull enable for gpio22.
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_gpio21	[11]	RW	1'h1	pad oex enable for gpio21. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_gpio21	[10]	RW	1'h1	schmitt trigger enable for gpio21. '1' enables schmitt trigger input function
driving_selector_gpio21	[9:6]	RW	4'h8	driving selector for gpio21

pin_mux_selector_gpio21	[5:4]	RW	2'h0	pin mux selector for gpio21
reserved	[3:2]	RW	2'h0	reserved
pull_selector_gpio21	[1]	RW	1'h1	pull selector for gpio21.
pull_enable_gpio21	[0]	RW	1'h1	pull enable for gpio21.

8.5.2.47 pin mux and io control register for gpio23 and gpio24 (0xB8)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_gpio24	[27]	RW	1'h1	pad oex enable for gpio24. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_gpio24	[26]	RW	1'h1	schmitt trigger enable for gpio24. '1' enables schmitt trigger input function
driving_selector_gpio24	[25:22]	RW	4'h8	driving selector for gpio24
pin_mux_selector_gpio24	[21:20]	RW	2'h0	pin mux selector for gpio24
reserved	[19:18]	RW	2'h0	reserved
pull_selector_gpio24	[17]	RW	1'h0	pull selector for gpio24.
pull_enable_gpio24	[16]	RW	1'h1	pull enable for gpio24.
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_gpio23	[11]	RW	1'h1	pad oex enable for gpio23. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_gpio23	[10]	RW	1'h1	schmitt trigger enable for gpio23. '1' enables schmitt trigger input function
driving_selector_gpio23	[9:6]	RW	4'h8	driving selector for gpio23
pin_mux_selector_gpio23	[5:4]	RW	2'h0	pin mux selector for gpio23
reserved	[3:2]	RW	2'h0	reserved
pull_selector_gpio23	[1]	RW	1'h0	pull selector for gpio23.
pull_enable_gpio23	[0]	RW	1'h1	pull enable for gpio23.

8.5.2.48 pin mux and io control register for gpio25 and gpio26 (0xBC)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_gpio26	[27]	RW	1'h1	pad oex enable for gpio26. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_gpio26	[26]	RW	1'h1	schmitt trigger enable for gpio26. '1' enables schmitt trigger input function
driving_selector_gpio26	[25:22]	RW	4'h8	driving selector for gpio26
pin_mux_selector_gpio26	[21:20]	RW	2'h0	pin mux selector for gpio26
reserved	[19:18]	RW	2'h0	reserved
pull_selector_gpio2	[17]	RW	1'h0	pull selector for gpio26.

6				
pull_enable_gpio26	[16]	RW	1'h1	pull enable for gpio26.
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_gpio25	[11]	RW	1'h1	pad oex enable for gpio25. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_gpio25	[10]	RW	1'h0	schmitt trigger enable for gpio25. '1' enables schmitt trigger input function
driving_selector_gpio25	[9:6]	RW	4'h8	driving selector for gpio25
pin_mux_selector_gpio25	[5:4]	RW	2'h0	pin mux selector for gpio25
reserved	[3:2]	RW	2'h0	reserved
pull_selector_gpio25	[1]	RW	1'h0	pull selector for gpio25.
pull_enable_gpio25	[0]	RW	1'h1	pull enable for gpio25.

8.5.2.49 pin mux and io control register for gpio27 and gpio28 (0xC0)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_gpio28	[27]	RW	1'h1	pad oex enable for gpio28. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_gpio28	[26]	RW	1'h1	schmitt trigger enable for gpio28. '1' enables schmitt trigger input function
driving_selector_gpio28	[25:22]	RW	4'h8	driving selector for gpio28
pin_mux_selector_gpio28	[21:20]	RW	2'h0	pin mux selector for gpio28
reserved	[19:18]	RW	2'h0	reserved
pull_selector_gpio28	[17]	RW	1'h1	pull selector for gpio28.
pull_enable_gpio28	[16]	RW	1'h1	pull enable for gpio28.
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_gpio27	[11]	RW	1'h1	pad oex enable for gpio27. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_gpio27	[10]	RW	1'h1	schmitt trigger enable for gpio27. '1' enables schmitt trigger input function
driving_selector_gpio27	[9:6]	RW	4'h8	driving selector for gpio27
pin_mux_selector_gpio27	[5:4]	RW	2'h0	pin mux selector for gpio27
reserved	[3:2]	RW	2'h0	reserved
pull_selector_gpio27	[1]	RW	1'h1	pull selector for gpio27.
pull_enable_gpio27	[0]	RW	1'h1	pull enable for gpio27.

8.5.2.50 pin mux and io control register for gpio29 and gpio30 (0xC4)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:28]	RW	4'h0	reserved
pad_oex_enable_g	[27]	RW	1'h1	pad oex enable for gpio30.

pio30				'1' enables pad output mode under ip's drive
schmitt_trigger_enable_gpio30	[26]	RW	1'h1	schmitt trigger enable for gpio30. '1' enables schmitt trigger input function
driving_selector_gpio30	[25:22]	RW	4'h8	driving selector for gpio30
pin_mux_selector_gpio30	[21:20]	RW	2'h0	pin mux selector for gpio30
reserved	[19:18]	RW	2'h0	reserved
pull_selector_gpio30	[17]	RW	1'h0	pull selector for gpio30.
pull_enable_gpio30	[16]	RW	1'h0	pull enable for gpio30.
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_gpio29	[11]	RW	1'h1	pad oex enable for gpio29. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_gpio29	[10]	RW	1'h1	schmitt trigger enable for gpio29. '1' enables schmitt trigger input function
driving_selector_gpio29	[9:6]	RW	4'h8	driving selector for gpio29
pin_mux_selector_gpio29	[5:4]	RW	2'h0	pin mux selector for gpio29
reserved	[3:2]	RW	2'h0	reserved
pull_selector_gpio29	[1]	RW	1'h0	pull selector for gpio29.
pull_enable_gpio29	[0]	RW	1'h0	pull enable for gpio29.

8.5.2.51 pin mux and io control register for gpio31 and boot_sel0 (0xC8)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:27]	RW	5'h0	reserved
schmitt_trigger_enable_boot_sel0	[26]	RW	1'h1	schmitt trigger enable for boot_sel0. '1' enables schmitt trigger input function
reserved	[25:18]	RW	8'h0	reserved
pull_selector_boot_sel0	[17]	RW	1'h0	pull selector for boot_sel0.
pull_enable_boot_sel0	[16]	RW	1'h1	pull enable for boot_sel0.
reserved	[15:12]	RW	4'h0	reserved
pad_oex_enable_gpio31	[11]	RW	1'h1	pad oex enable for gpio31. '1' enables pad output mode under ip's drive
schmitt_trigger_enable_gpio31	[10]	RW	1'h1	schmitt trigger enable for gpio31. '1' enables schmitt trigger input function
driving_selector_gpio31	[9:6]	RW	4'h8	driving selector for gpio31
pin_mux_selector_gpio31	[5:4]	RW	2'h0	pin mux selector for gpio31
reserved	[3:2]	RW	2'h0	reserved
pull_selector_gpio31	[1]	RW	1'h0	pull selector for gpio31.
pull_enable_gpio31	[0]	RW	1'h1	pull enable for gpio31.

8.5.2.52 pin mux and io control register for boot_sel1 and boot_sel2 (0xCC)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:27]	RW	5'h0	reserved
schmitt_trigger_enable_boot_sel2	[26]	RW	1'h1	schmitt trigger enable for boot_sel2. '1' enables schmitt trigger input function
reserved	[25:18]	RW	8'h0	reserved
pull_selector_boot_sel2	[17]	RW	1'h0	pull selector for boot_sel2.
pull_enable_boot_sel2	[16]	RW	1'h1	pull enable for boot_sel2.
reserved	[15:11]	RW	5'h0	reserved
schmitt_trigger_enable_boot_sel1	[10]	RW	1'h1	schmitt trigger enable for boot_sel1. '1' enables schmitt trigger input function
reserved	[9:2]	RW	8'h0	reserved
pull_selector_boot_sel1	[1]	RW	1'h0	pull selector for boot_sel1.
pull_enable_boot_sel1	[0]	RW	1'h1	pull enable for boot_sel1.

8.5.2.53 pin mux and io control register for boot_sel3 and mode_sel0 (0xD0)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:27]	RW	5'h0	reserved
schmitt_trigger_enable_mode_sel0	[26]	RW	1'h1	schmitt trigger enable for mode_sel0. '1' enables schmitt trigger input function
reserved	[25:18]	RW	8'h0	reserved
pull_selector_mode_sel0	[17]	RW	1'h0	pull selector for mode_sel0.
pull_enable_mode_sel0	[16]	RW	1'h1	pull enable for mode_sel0.
reserved	[15:11]	RW	5'h0	reserved
schmitt_trigger_enable_boot_sel3	[10]	RW	1'h1	schmitt trigger enable for boot_sel3. '1' enables schmitt trigger input function
reserved	[9:2]	RW	8'h0	reserved
pull_selector_boot_sel3	[1]	RW	1'h0	pull selector for boot_sel3.
pull_enable_boot_sel3	[0]	RW	1'h1	pull enable for boot_sel3.

8.5.2.54 pin mux and io control register for mode_sel1 and mode_sel2 (0xD4)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:27]	RW	5'h0	reserved
schmitt_trigger_enable_mode_sel2	[26]	RW	1'h1	schmitt trigger enable for mode_sel2. '1' enables schmitt trigger input function
reserved	[25:18]	RW	8'h0	reserved
pull_selector_mode_sel2	[17]	RW	1'h0	pull selector for mode_sel2.

pull_enable_mode_sel2	[16]	RW	1'h1	pull enable for mode_sel2.
reserved	[15:11]	RW	5'h0	reserved
schmitt_trigger_enable_mode_sel1	[10]	RW	1'h1	schmitt trigger enable for mode_sel1. '1' enables schmitt trigger input function
reserved	[9:2]	RW	8'h0	reserved
pull_selector_mode_sel1	[1]	RW	1'h0	pull selector for mode_sel1.
pull_enable_mode_sel1	[0]	RW	1'h1	pull enable for mode_sel1.

8.5.2.55 pin mux and io control register for mode_sel3 and pll_clk_in (0xD8)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:27]	RW	5'h0	reserved
schmitt_trigger_enable_pll_clk_in	[26]	RW	1'h0	schmitt trigger enable for pll_clk_in. '1' enables schmitt trigger input function
reserved	[25:18]	RW	8'h0	reserved
pull_selector_pll_clk_in	[17]	RW	1'h0	pull selector for pll_clk_in.
pull_enable_pll_clk_in	[16]	RW	1'h0	pull enable for pll_clk_in.
reserved	[15:11]	RW	5'h0	reserved
schmitt_trigger_enable_mode_sel3	[10]	RW	1'h1	schmitt trigger enable for mode_sel3. '1' enables schmitt trigger input function
reserved	[9:2]	RW	8'h0	reserved
pull_selector_mode_sel3	[1]	RW	1'h0	pull selector for mode_sel3.
pull_enable_mode_sel3	[0]	RW	1'h1	pull enable for mode_sel3.

8.5.2.56 pin mux and io control register for pll_clk_in_dpll0 and pll_clk_in_dpll1 (0xDC)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:27]	RW	5'h0	reserved
schmitt_trigger_enable_pll_clk_in_dpll1	[26]	RW	1'h0	schmitt trigger enable for pll_clk_in_dpll1. '1' enables schmitt trigger input function
reserved	[25:18]	RW	8'h0	reserved
pull_selector_pll_clk_in_dpll1	[17]	RW	1'h0	pull selector for pll_clk_in_dpll1.
pull_enable_pll_clk_in_dpll1	[16]	RW	1'h0	pull enable for pll_clk_in_dpll1.
reserved	[15:11]	RW	5'h0	reserved
schmitt_trigger_enable_pll_clk_in_dpll0	[10]	RW	1'h0	schmitt trigger enable for pll_clk_in_dpll0. '1' enables schmitt trigger input function
reserved	[9:2]	RW	8'h0	reserved
pull_selector_pll_clk_in_dpll0	[1]	RW	1'h0	pull selector for pll_clk_in_dpll0.
pull_enable_pll_clk_in_dpll0	[0]	RW	1'h0	pull enable for pll_clk_in_dpll0.

8.5.2.57 pin mux and io control register for pll_src_sel and sys_rst_x (0xE0)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:27]	RW	5'h0	reserved
schmitt_trigger_enable_sys_rst_x	[26]	RW	1'h1	schmitt trigger enable for sys_rst_x. '1' enables schmitt trigger input function
reserved	[25:18]	RW	8'h0	reserved
pull_selector_sys_rst_x	[17]	RW	1'h1	pull selector for sys_rst_x.
pull_enable_sys_rst_x	[16]	RW	1'h1	pull enable for sys_rst_x.
reserved	[15:11]	RW	5'h0	reserved
schmitt_trigger_enable_pll_src_sel	[10]	RW	1'h1	schmitt trigger enable for pll_src_sel. '1' enables schmitt trigger input function
reserved	[9:2]	RW	8'h0	reserved
pull_selector_pll_src_sel	[1]	RW	1'h0	pull selector for pll_src_sel.
pull_enable_pll_src_sel	[0]	RW	1'h1	pull enable for pll_src_sel.

8.5.2.58 pin mux and io control register for test_en and bisr_byp (0xE4)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:27]	RW	5'h0	reserved
schmitt_trigger_enable_bisr_byp	[26]	RW	1'h1	schmitt trigger enable for bisr_byp. '1' enables schmitt trigger input function
reserved	[25:18]	RW	8'h0	reserved
pull_selector_bisr_byp	[17]	RW	1'h0	pull selector for bisr_byp.
pull_enable_bisr_byp	[16]	RW	1'h1	pull enable for bisr_byp.
reserved	[15:11]	RW	5'h0	reserved
reserved	[10]	RW	1'h1	reserved
reserved	[9:2]	RW	8'h0	reserved
reserved	[1]	RW	1'h0	reserved
reserved	[0]	RW	1'h1	reserved

8.5.2.59 pin mux and io control register for p08_pwr_good and osc_ds3 (0xE8)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:27]	RW	5'h0	reserved
schmitt_trigger_enable_osc_ds3	[26]	RW	1'h1	schmitt trigger enable for osc_ds3. '1' enables schmitt trigger input function
reserved	[25:18]	RW	8'h0	reserved
pull_selector_osc_ds3	[17]	RW	1'h0	pull selector for osc_ds3.
pull_enable_osc_ds3	[16]	RW	1'h1	pull enable for osc_ds3.
reserved	[15:11]	RW	5'h0	reserved

schmitt_trigger_enable_p08_pwr_good	[10]	RW	1'h1	schmitt trigger enable for p08_pwr_good. '1' enables schmitt trigger input function
reserved	[9:2]	RW	8'h0	reserved
pull_selector_p08_pwr_good	[1]	RW	1'h0	pull selector for p08_pwr_good.
pull_enable_p08_pwr_good	[0]	RW	1'h1	pull enable for p08_pwr_good.

8.5.2.60 pin mux and io control register for ddr_pwr_good (0xEC)

Field Name	Bit	Type	Reset Value	Field Description
reserved	[31:16]	RO	16'h0	reserved
reserved	[15:11]	RW	5'h0	reserved
schmitt_trigger_enable_ddr_pwr_good	[10]	RW	1'h1	schmitt trigger enable for ddr_pwr_good. '1' enables schmitt trigger input function
reserved	[9:2]	RW	8'h0	reserved
pull_selector_ddr_pwr_good	[1]	RW	1'h0	pull selector for ddr_pwr_good.
pull_enable_ddr_pwr_good	[0]	RW	1'h1	pull enable for ddr_pwr_good.

8.6 Clock Generation Control and Status Registers

8.6.1 Clock Generation Control and Status Registers Summary

Table 16 Clock Generation Control and Status Registers Summary

Offset	Register Name	Reg Description
0x00	clk_gen_reg0	clock enable register 0
0x04	clk_gen_reg1	clock enable register 1
0x08	clk_gen_reg2	clock enable register 2
0x20	clk_gen_reg3	clock select register 0
0x40	clk_gen_reg4	clock divider control register of divider 0 for clk_a53
0x44	clk_gen_reg5	clock divider control register of divider 1 for clk_a53
0x48	clk_gen_reg6	clock divider control register of divider for clk_50m_a53
0x4C	clk_gen_reg7	clock divider control register of divider for clk_emmc
0x50	clk_gen_reg8	clock divider control register of divider for clk_100k_emmc
0x54	clk_gen_reg9	clock divider control register of divider for clk_sd
0x58	clk_gen_reg10	clock divider control register of divider for clk_100k_sd
0x5C	clk_gen_reg11	clock divider control register of divider for clk_tx_eth0
0x60	clk_gen_reg12	clock divider control register of divider for clk_ptp_ref_i_eth0
0x64	clk_gen_reg13	clock divider control register of divider for clk_ref_eth0
0x68	clk_gen_reg14	clock divider control register of divider for clk_tx_eth1
0x6C	clk_gen_reg15	clock divider control register of divider for clk_ptp_ref_i_eth1
0x70	clk_gen_reg16	clock divider control register of divider for clk_ref_eth1
0x74	clk_gen_reg17	clock divider control register of divider for clk_uart_500m

0x78	clk_gen_reg18	clock divider control register of divider for clk_timer_1
0x7C	clk_gen_reg19	clock divider control register of divider for clk_timer_2
0x80	clk_gen_reg20	clock divider control register of divider for clk_timer_3
0x84	clk_gen_reg21	clock divider control register of divider for clk_timer_4
0x88	clk_gen_reg22	clock divider control register of divider for clk_timer_5
0x8C	clk_gen_reg23	clock divider control register of divider for clk_timer_6
0x90	clk_gen_reg24	clock divider control register of divider for clk_timer_7
0x94	clk_gen_reg25	clock divider control register of divider for clk_timer_8
0x98	clk_gen_reg26	clock divider control register of divider for clk_efuse
0x9C	clk_gen_reg27	clock divider control register of divider for clk_gpio_db
0xA0	clk_gen_reg28	clock divider control register of divider 0 for clk_tpu
0xA4	clk_gen_reg29	clock divider control register of divider 1 for clk_tpu
0xA8	clk_gen_reg30	clock divider control register of divider for clk_fixed_tpu_clk
0xAC	clk_gen_reg31	clock divider control register of divider 0 for clk_axi_vde_wave
0xB0	clk_gen_reg32	clock divider control register of divider 1 for clk_axi_vde_wave
0xB4	clk_gen_reg33	clock divider control register of divider for clk_axi3
0xB8	clk_gen_reg34	clock divider control register of divider for clk_axi6
0xBC	clk_gen_reg35	clock divider control register of divider for clk_axi8
0xC0	clk_gen_reg36	clock divider control register of divider 0 for clk_axi10
0xC4	clk_gen_reg37	clock divider control register of divider 1 for clk_axi10
0xC8	clk_gen_reg38	clock divider control register of divider 0 for clk_axi_ddr
0xCC	clk_gen_reg39	clock divider control register of divider 1 for clk_axi_ddr
0xD0	clk_gen_reg40	clock divider control register of divider 0 for clk_ddr0
0xD4	clk_gen_reg41	clock divider control register of divider 1 for clk_ddr0
0xD8	clk_gen_reg42	clock divider control register of divider 0 for clk_ddr12
0xDC	clk_gen_reg43	clock divider control register of divider 1 for clk_ddr12

8.6.2 Clock Generation Control and Status Registers Description

8.6.2.1 clock enable register 0 (0x00)

Field Name	Bit	Type	Reset Value	Field Description
clk_en_apb_gpio_gen_reg0	[31]	RW	1'h1	clock enable for clk_apb_gpio (1: enable; 0: gate)
clk_en_ahb_rom_gen_reg0	[30]	RW	1'h1	clock enable for clk_ahb_rom (1: enable; 0: gate)
clk_en_apb_efuse_gen_reg0	[29]	RW	1'h1	clock enable for clk_apb_efuse (1: enable; 0: gate)
clk_en_efuse_gen_reg0	[28]	RW	1'h1	clock enable for clk_efuse (1: enable; 0: gate)
clk_en_timer_8_gen_reg0	[27]	RW	1'h1	clock enable for clk_timer_8 (1: enable; 0: gate)
clk_en_timer_7_gen_reg0	[26]	RW	1'h1	clock enable for clk_timer_7 (1: enable; 0: gate)
clk_en_timer_6_gen_reg0	[25]	RW	1'h1	clock enable for clk_timer_6 (1: enable; 0: gate)
clk_en_timer_5_gen_reg0	[24]	RW	1'h1	clock enable for clk_timer_5 (1: enable; 0: gate)

clk_en_timer_4_gen_reg0	[23]	RW	1'h1	clock enable for clk_timer_4 (1: enable; 0: gate)
clk_en_timer_3_gen_reg0	[22]	RW	1'h1	clock enable for clk_timer_3 (1: enable; 0: gate)
clk_en_timer_2_gen_reg0	[21]	RW	1'h1	clock enable for clk_timer_2 (1: enable; 0: gate)
clk_en_timer_1_gen_reg0	[20]	RW	1'h1	clock enable for clk_timer_1 (1: enable; 0: gate)
clk_en_apb_timer_gen_reg0	[19]	RW	1'h1	clock enable for clk_apb_timer (1: enable; 0: gate)
clk_en_apb_intc_gen_reg0	[18]	RW	1'h1	clock enable for clk_apb_intc (1: enable; 0: gate)
clk_en_uart_500m_gen_reg0	[17]	RW	1'h1	clock enable for clk_uart_500m (1: enable; 0: gate)
clk_en_sdma_axi_gen_reg0	[16]	RW	1'h1	clock enable for clk_sdma_axi (1: enable; 0: gate)
clk_en_ref_eth1_gen_reg0	[15]	RW	1'h1	clock enable for clk_ref_eth1 (1: enable; 0: gate)
clk_en_ptp_ref_i_eth1_gen_reg0	[14]	RW	1'h1	clock enable for clk_ptp_ref_i_eth1 (1: enable; 0: gate)
clk_en_axi6_eth1_gen_reg0	[13]	RW	1'h1	clock enable for clk_axi6_eth1 (1: enable; 0: gate)
clk_en_tx_eth1_gen_reg0	[12]	RW	1'h1	clock enable for clk_tx_eth1 (1: enable; 0: gate)
clk_en_ref_eth0_gen_reg0	[11]	RW	1'h1	clock enable for clk_ref_eth0 (1: enable; 0: gate)
clk_en_ptp_ref_i_eth0_gen_reg0	[10]	RW	1'h1	clock enable for clk_ptp_ref_i_eth0 (1: enable; 0: gate)
clk_en_axi6_eth0_gen_reg0	[9]	RW	1'h1	clock enable for clk_axi6_eth0 (1: enable; 0: gate)
clk_en_tx_eth0_gen_reg0	[8]	RW	1'h1	clock enable for clk_tx_eth0 (1: enable; 0: gate)
clk_en_100k_sd_gen_reg0	[7]	RW	1'h1	clock enable for clk_100k_sd (1: enable; 0: gate)
clk_en_sd_gen_reg0	[6]	RW	1'h1	clock enable for clk_sd (1: enable; 0: gate)
clk_en_axi_sd_gen_reg0	[5]	RW	1'h1	clock enable for clk_axi_sd (1: enable; 0: gate)
clk_en_100k_emmc_gen_reg0	[4]	RW	1'h1	clock enable for clk_100k_emmc (1: enable; 0: gate)
clk_en_emmc_gen_reg0	[3]	RW	1'h1	clock enable for clk_emmc (1: enable; 0: gate)
clk_en_axi_emmc_gen_reg0	[2]	RW	1'h1	clock enable for clk_axi_emmc (1: enable; 0: gate)
clk_en_50m_a53_gen_reg0	[1]	RW	1'h1	clock enable for clk_50m_a53 (1: enable; 0: gate)
clk_en_a53_gen_reg0	[0]	RW	1'h1	clock enable for clk_a53 (1: enable; 0: gate)

8.6.2.2 clock enable register 1 (0x04)

Field Name	**Bit**	**Type**	**Reset Value**	**Field Description**
clk_en_axi_vd0_vpp_gen_reg1	[31]	RW	1'h1	clock enable for clk_axi_vd0_vpp (1: enable; 0: gate)
clk_en_apb_vd0_vpp_gen_reg1	[30]	RW	1'h1	clock enable for clk_apb_vd0_vpp (1: enable; 0: gate)

clk_en_axi_vd0_jpeg1_gen_reg1	[29]	RW	1'h1	clock enable for clk_axi_vd0_jpeg1 (1: enable; 0: gate)
clk_en_apb_vd0_jpeg1_gen_reg1	[28]	RW	1'h1	clock enable for clk_apb_vd0_jpeg1 (1: enable; 0: gate)
clk_en_axi_vd0_jpeg0_gen_reg1	[27]	RW	1'h1	clock enable for clk_axi_vd0_jpeg0 (1: enable; 0: gate)
clk_en_apb_vd0_jpeg0_gen_reg1	[26]	RW	1'h1	clock enable for clk_apb_vd0_jpeg0 (1: enable; 0: gate)
clk_en_axi_vd0_wave1_gen_reg1	[25]	RW	1'h1	clock enable for clk_axi_vd0_wave1 (1: enable; 0: gate)
clk_en_apb_vd0_wave1_gen_reg1	[24]	RW	1'h1	clock enable for clk_apb_vd0_wave1 (1: enable; 0: gate)
clk_en_axi_vd0_wave0_gen_reg1	[23]	RW	1'h1	clock enable for clk_axi_vd0_wave0 (1: enable; 0: gate)
clk_en_apb_vd0_wave0_gen_reg1	[22]	RW	1'h1	clock enable for clk_apb_vd0_wave0 (1: enable; 0: gate)
clk_en_axi_vde_axi_bridge_gen_reg1	[21]	RW	1'h1	clock enable for clk_axi_vde_axi_bridge (1: enable; 0: gate)
clk_en_apb_vde_wave_gen_reg1	[20]	RW	1'h1	clock enable for clk_apb_vde_wave (1: enable; 0: gate)
clk_en_axi_vde_wave_gen_reg1	[19]	RW	1'h1	clock enable for clk_axi_vde_wave (1: enable; 0: gate)
clk_en_riscv_gen_reg1	[18]	RW	1'h1	clock enable for clk_riscv (1: enable; 0: gate)
clk_en_axi_spacc_gen_reg1	[17]	RW	1'h1	clock enable for clk_axi_spacc (1: enable; 0: gate)
clk_en_axi_pka_gen_reg1	[16]	RW	1'h1	clock enable for clk_axi_pka (1: enable; 0: gate)
clk_en_axi_dbg_i2c_gen_reg1	[15]	RW	1'h1	clock enable for clk_axi_dbg_i2c (1: enable; 0: gate)
clk_en_fixed_tpu_clk_gen_reg1	[14]	RW	1'h1	clock enable for clk_fixed_tpu_clk (1: enable; 0: gate)
clk_en_tpu_gen_reg1	[13]	RW	1'h1	clock enable for clk_tpu (1: enable; 0: gate)
clk_en_gdma_gen_reg1	[12]	RW	1'h1	clock enable for clk_gdma (1: enable; 0: gate)
clk_en_axisram_gen_reg1	[11]	RW	1'h1	clock enable for clk_axisram (1: enable; 0: gate)
clk_en_arm_gen_reg1	[10]	RW	1'h1	clock enable for clk_arm (1: enable; 0: gate)
clk_en_axi8_cdma_gen_reg1	[9]	RW	1'h1	clock enable for clk_axi8_cdma (1: enable; 0: gate)
clk_en_axi8_pcie_gen_reg1	[8]	RW	1'h1	clock enable for clk_axi8_pcie (1: enable; 0: gate)
clk_en_axi8_mmu_gen_reg1	[7]	RW	1'h1	clock enable for clk_axi8_mmu (1: enable; 0: gate)
clk_en_apb_trng_gen_reg1	[6]	RW	1'h1	clock enable for clk_apb_trng (1: enable; 0: gate)
clk_en_apb_pw	[5]	RW	1'h1	clock enable for clk_apb_pwm (1: enable; 0: gate)

m_gen_reg1				
clk_en_apb_wdt_gen_reg1	[4]	RW	1'h1	clock enable for clk_apb_wdt (1: enable; 0: gate)
clk_en_apb_i2c_gen_reg1	[3]	RW	1'h1	clock enable for clk_apb_i2c (1: enable; 0: gate)
clk_en_ahb_sf_gen_reg1	[2]	RW	1'h1	clock enable for clk_ahb_sf (1: enable; 0: gate)
clk_en_gpio_db_gen_reg1	[1]	RW	1'h1	clock enable for clk_gpio_db (1: enable; 0: gate)
clk_en_apb_gpio_intr_gen_reg1	[0]	RW	1'h1	clock enable for clk_apb_gpio_intr (1: enable; 0: gate)

8.6.2.3 clock enable register 2 (0x08)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:20]	RW	12'h0	reserved
clk_en_tpu_for_tpu_only_gen_reg2	[19]	RW	1'h1	clock enable for clk_tpu_for_tpu_only (1: enable; 0: gate)
clk_en_tsdma_gen_reg2	[18]	RW	1'h1	clock enable for clk_tsdma (1: enable; 0: gate)
clk_en_hau_ngs_gen_reg2	[17]	RW	1'h1	clock enable for clk_hau_ngs (1: enable; 0: gate)
clk_en_ddr12_gen_reg2	[16]	RW	1'h1	clock enable for clk_ddr12 (1: enable; 0: gate)
clk_en_ddr0_gen_reg2	[15]	RW	1'h1	clock enable for clk_ddr0 (1: enable; 0: gate)
clk_en_axi_ddr_gen_reg2	[14]	RW	1'h1	clock enable for clk_axi_ddr (1: enable; 0: gate)
clk_en_axi10_gen_reg2	[13]	RW	1'h1	clock enable for clk_axi10 (1: enable; 0: gate)
clk_en_axi8_gen_reg2	[12]	RW	1'h1	clock enable for clk_axi8 (1: enable; 0: gate)
clk_en_axi6_gen_reg2	[11]	RW	1'h1	clock enable for clk_axi6 (1: enable; 0: gate)
clk_en_axi3_gen_reg2	[10]	RW	1'h1	clock enable for clk_axi3 (1: enable; 0: gate)
clk_en_axi_vd1_vpp_gen_reg2	[9]	RW	1'h1	clock enable for clk_axi_vd1_vpp (1: enable; 0: gate)
clk_en_apb_vd1_vpp_gen_reg2	[8]	RW	1'h1	clock enable for clk_apb_vd1_vpp (1: enable; 0: gate)
clk_en_axi_vd1_jpeg1_gen_reg2	[7]	RW	1'h1	clock enable for clk_axi_vd1_jpeg1 (1: enable; 0: gate)
clk_en_apb_vd1_jpeg1_gen_reg2	[6]	RW	1'h1	clock enable for clk_apb_vd1_jpeg1 (1: enable; 0: gate)
clk_en_axi_vd1_jpeg0_gen_reg2	[5]	RW	1'h1	clock enable for clk_axi_vd1_jpeg0 (1: enable; 0: gate)
clk_en_apb_vd1_jpeg0_gen_reg2	[4]	RW	1'h1	clock enable for clk_apb_vd1_jpeg0 (1: enable; 0: gate)
clk_en_axi_vd1_wave1_gen_reg2	[3]	RW	1'h1	clock enable for clk_axi_vd1_wave1 (1: enable; 0: gate)

clk_en_apb_vd1_wave1_gen_reg2	[2]	RW	1'h1	clock enable for clk_apb_vd1_wave1 (1: enable; 0: gate)
clk_en_axi_vd1_wave0_gen_reg2	[1]	RW	1'h1	clock enable for clk_axi_vd1_wave0 (1: enable; 0: gate)
clk_en_apb_vd1_wave0_gen_reg2	[0]	RW	1'h1	clock enable for clk_apb_vd1_wave0 (1: enable; 0: gate)

8.6.2.4 *clock select register 0 (0x20)*

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:30]	RW	2'h0	reserved
clk_en_ddr12_div1_gen_reg3	[29]	RW	1'h1	clock enable for clk_ddr12_div1 (1: enable; 0: gate)
clk_en_ddr12_div0_gen_reg3	[28]	RW	1'h1	clock enable for clk_ddr12_div0 (1: enable; 0: gate)
clk_en_ddr0_div1_gen_reg3	[27]	RW	1'h1	clock enable for clk_ddr0_div1 (1: enable; 0: gate)
clk_en_ddr0_div0_gen_reg3	[26]	RW	1'h1	clock enable for clk_ddr0_div0 (1: enable; 0: gate)
clk_en_axi_ddr_div1_gen_reg3	[25]	RW	1'h1	clock enable for clk_axi_ddr_div1 (1: enable; 0: gate)
clk_en_axi_ddr_div0_gen_reg3	[24]	RW	1'h1	clock enable for clk_axi_ddr_div0 (1: enable; 0: gate)
clk_en_axi10_div1_gen_reg3	[23]	RW	1'h1	clock enable for clk_axi10_div1 (1: enable; 0: gate)
clk_en_axi10_div0_gen_reg3	[22]	RW	1'h1	clock enable for clk_axi10_div0 (1: enable; 0: gate)
clk_en_axi_vde_wave_div1_gen_reg3	[21]	RW	1'h1	clock enable for clk_axi_vde_wave_div1 (1: enable; 0: gate)
clk_en_axi_vde_wave_div0_gen_reg3	[20]	RW	1'h1	clock enable for clk_axi_vde_wave_div0 (1: enable; 0: gate)
clk_en_tpu_div1_gen_reg3	[19]	RW	1'h1	clock enable for clk_tpu_div1 (1: enable; 0: gate)
clk_en_tpu_div0_gen_reg3	[18]	RW	1'h1	clock enable for clk_tpu_div0 (1: enable; 0: gate)
clk_en_a53_div1_gen_reg3	[17]	RW	1'h1	clock enable for clk_a53_div1 (1: enable; 0: gate)
clk_en_a53_div0_gen_reg3	[16]	RW	1'h1	clock enable for clk_a53_div0 (1: enable; 0: gate)
Reserved	[15:7]	RW	9'h0	reserved
clk_sel_ddr12s_clock_core_ddrc_core_clk_gen_reg3	[6]	RW	1'h1	clock select for ddr12's clock core_ddrc_core_clk (aka clk_ddr12) 1: select in_dpll1_clk as clock source 0: select in_fpll_clk as clock source
clk_sel_ddr0s_clock_core_ddrc_core_clk_gen_reg3	[5]	RW	1'h1	clock select for ddr0's clock core_ddrc_core_clk (aka clk_ddr0) 1: select in_dpll0_clk as clock source 0: select in_fpll_clk as clock source
clk_sel_fabric_axi_ddrs_clock_aclk_gen_reg3	[4]	RW	1'h1	clock select for fabric_axi_ddr's clock aclk (aka clk_axi_ddr) 1: select in_mpll_clk as clock source 0: select in_fpll_clk as clock source

clk_sel_fabric_axi10s_clock_aclk_gen_reg3	[3]	RW	1'h1	clock select for fabric_axi10's clock aclk (aka clk_axi10) 1: select in_vpll_clk as clock source 0: select in_fpll_clk as clock source
clk_sel_vde_wave521s_clock_clk_wave_axi_gen_reg3	[2]	RW	1'h1	clock select for vde_wave521's clock clk_wave_axi (aka clk_axi_vde_wave) 1: select in_vpll_clk as clock source 0: select in_fpll_clk as clock source
clk_sel_tpus_clock_clk_gen_reg3	[1]	RW	1'h1	clock select for tpu's clock clk (aka clk_tpu) 1: select in_tpll_clk as clock source 0: select in_fpll_clk as clock source
clk_sel_a53s_clock_cpu_clk_gen_reg3	[0]	RW	1'h1	clock select for a53's clock cpu_clk (aka clk_a53) 1: select in_mpll_clk as clock source 0: select in_fpll_clk as clock source

8.6.2.5 clock divider control register of divider 0 for clk_a53 （0x40）

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:21]	RW	11'h0	reserved
clock_divider_factor_gen_reg4	[20:16]	RW	5'h0	clock divider factor
Reserved	[15:4]	RW	12'h0	reserved
select_divide_factor_from_register_gen_reg4	[3]	RW	1'h0	select divide factor from register 0: select initial value 1: select divide factor from this register
select_high_wide_control_from_register_gen_reg4	[2]	RW	1'h0	select high wide control from register 0: select initial value 1: select high wide from this register
high_wide_control_gen_reg4	[1]	RW	1'h0	high wide control (when divider factor is odd) 0: low level of the clock is wider 1: high level of the clock is wider
divider_reset_control_gen_reg4	[0]	RW	1'h1	divider reset control 0: assert reset 1: de-assert reset

8.6.2.6 clock divider control register of divider 1 for clk_a53 (0x44)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:21]	RW	11'h0	reserved
clock_divider_factor_gen_reg5	[20:16]	RW	5'h0	clock divider factor
Reserved	[15:4]	RW	12'h0	reserved
select_divide_factor_from_register_gen_reg5	[3]	RW	1'h0	select divide factor from register 0: select initial value 1: select divide factor from this register
select_high_wide_control_from_register_gen_reg5	[2]	RW	1'h0	select high wide control from register 0: select initial value 1: select high wide from this register
high_wide_control_gen_reg5	[1]	RW	1'h0	high wide control (when divider factor is odd) 0: low level of the clock is wider 1: high level of the clock is wider

divider_reset_control_gen_reg5	[0]	RW	1'h1	divider reset control 0: assert reset 1: de-assert reset

8.6.2.7 clock divider control register of divider for clk_50m_a53 (0x48)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:24]	RW	8'h0	reserved
clock_divider_factor_gen_reg6	[23:16]	RW	8'h0	clock divider factor
Reserved	[15:4]	RW	12'h0	reserved
select_divide_factor_from_register_gen_reg6	[3]	RW	1'h0	select divide factor from register 0: select initial value 1: select divide factor from this register
select_high_wide_control_from_register_gen_reg6	[2]	RW	1'h0	select high wide control from register 0: select initial value 1: select high wide from this register
high_wide_control_gen_reg6	[1]	RW	1'h0	high wide control (when divider factor is odd) 0: low level of the clock is wider 1: high level of the clock is wider
divider_reset_control_gen_reg6	[0]	RW	1'h1	divider reset control 0: assert reset 1: de-assert reset

8.6.2.8 clock divider control register of divider for clk_emmc (0x4C)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:21]	RW	11'h0	reserved
clock_divider_factor_gen_reg7	[20:16]	RW	5'h0	clock divider factor
Reserved	[15:4]	RW	12'h0	reserved
select_divide_factor_from_register_gen_reg7	[3]	RW	1'h0	select divide factor from register 0: select initial value 1: select divide factor from this register
select_high_wide_control_from_register_gen_reg7	[2]	RW	1'h0	select high wide control from register 0: select initial value 1: select high wide from this register
high_wide_control_gen_reg7	[1]	RW	1'h0	high wide control (when divider factor is odd) 0: low level of the clock is wider 1: high level of the clock is wider
divider_reset_control_gen_reg7	[0]	RW	1'h1	divider reset control 0: assert reset 1: de-assert reset

8.6.2.9 clock divider control register of divider for clk_100k_emmc (0x50)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:24]	RW	8'h0	reserved

clock_divider_factor_gen_reg8	[23:16]	RW	8'h0	clock divider factor
Reserved	[15:4]	RW	12'h0	reserved
select_divide_factor_from_register_gen_reg8	[3]	RW	1'h0	select divide factor from register 0: select initial value 1: select divide factor from this register
select_high_wide_control_from_register_gen_reg8	[2]	RW	1'h0	select high wide control from register 0: select initial value 1: select high wide from this register
high_wide_control_gen_reg8	[1]	RW	1'h0	high wide control (when divider factor is odd) 0: low level of the clock is wider 1: high level of the clock is wider
divider_reset_control_gen_reg8	[0]	RW	1'h1	divider reset control 0: assert reset 1: de-assert reset

8.6.2.10 clock divider control register of divider for clk_sd (0x54)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:21]	RW	11'h0	reserved
clock_divider_factor_gen_reg9	[20:16]	RW	5'h0	clock divider factor
Reserved	[15:4]	RW	12'h0	reserved
select_divide_factor_from_register_gen_reg9	[3]	RW	1'h0	select divide factor from register 0: select initial value 1: select divide factor from this register
select_high_wide_control_from_register_gen_reg9	[2]	RW	1'h0	select high wide control from register 0: select initial value 1: select high wide from this register
high_wide_control_gen_reg9	[1]	RW	1'h0	high wide control (when divider factor is odd) 0: low level of the clock is wider 1: high level of the clock is wider
divider_reset_control_gen_reg9	[0]	RW	1'h1	divider reset control 0: assert reset 1: de-assert reset

8.6.2.11 clock divider control register of divider for clk_100k_sd (0x58)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:24]	RW	8'h0	reserved
clock_divider_factor_gen_reg10	[23:16]	RW	8'h0	clock divider factor
Reserved	[15:4]	RW	12'h0	reserved
select_divide_factor_from_register_gen_reg10	[3]	RW	1'h0	select divide factor from register 0: select initial value 1: select divide factor from this register
select_high_wide_control_from_register_gen_reg10	[2]	RW	1'h0	select high wide control from register 0: select initial value 1: select high wide from this register
high_wide_contr	[1]	RW	1'h0	high wide control (when divider factor is odd)

ol_gen_reg10				0: low level of the clock is wider 1: high level of the clock is wider
divider_reset_control_gen_reg10	[0]	RW	1'h1	divider reset control 0: assert reset 1: de-assert reset

8.6.2.12 clock divider control register of divider for clk_tx_eth0 (0x5C)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:27]	RW	5'h0	reserved
clock_divider_factor_gen_reg11	[26:16]	RW	11'h0	clock divider factor
Reserved	[15:4]	RW	12'h0	reserved
select_divide_factor_from_register_gen_reg11	[3]	RW	1'h0	select divide factor from register 0: select initial value 1: select divide factor from this register
select_high_wide_control_from_register_gen_reg11	[2]	RW	1'h0	select high wide control from register 0: select initial value 1: select high wide from this register
high_wide_control_gen_reg11	[1]	RW	1'h0	high wide control (when divider factor is odd) 0: low level of the clock is wider 1: high level of the clock is wider
divider_reset_control_gen_reg11	[0]	RW	1'h1	divider reset control 0: assert reset 1: de-assert reset

8.6.2.13 clock divider control register of divider for clk_ptp_ref_i_eth0 (0x60)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:24]	RW	8'h0	reserved
clock_divider_factor_gen_reg12	[23:16]	RW	8'h0	clock divider factor
Reserved	[15:4]	RW	12'h0	reserved
select_divide_factor_from_register_gen_reg12	[3]	RW	1'h0	select divide factor from register 0: select initial value 1: select divide factor from this register
select_high_wide_control_from_register_gen_reg12	[2]	RW	1'h0	select high wide control from register 0: select initial value 1: select high wide from this register
high_wide_control_gen_reg12	[1]	RW	1'h0	high wide control (when divider factor is odd) 0: low level of the clock is wider 1: high level of the clock is wider
divider_reset_control_gen_reg12	[0]	RW	1'h1	divider reset control 0: assert reset 1: de-assert reset

8.6.2.14 clock divider control register of divider for clk_ref_eth0 (0x64)

Field Name	Bit	Typ	Reset	Field Description

		e	Value	
Reserved	[31:24]	RW	8'h0	reserved
clock_divider_factor_gen_reg13	[23:16]	RW	8'h0	clock divider factor
Reserved	[15:4]	RW	12'h0	reserved
select_divide_factor_from_register_gen_reg13	[3]	RW	1'h0	select divide factor from register 0: select initial value 1: select divide factor from this register
select_high_wide_control_from_register_gen_reg13	[2]	RW	1'h0	select high wide control from register 0: select initial value 1: select high wide from this register
high_wide_control_gen_reg13	[1]	RW	1'h0	high wide control (when divider factor is odd) 0: low level of the clock is wider 1: high level of the clock is wider
divider_reset_control_gen_reg13	[0]	RW	1'h1	divider reset control 0: assert reset 1: de-assert reset

8.6.2.15 clock divider control register of divider for clk_tx_eth1 (0x68)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:27]	RW	5'h0	reserved
clock_divider_factor_gen_reg14	[26:16]	RW	11'h0	clock divider factor
Reserved	[15:4]	RW	12'h0	reserved
select_divide_factor_from_register_gen_reg14	[3]	RW	1'h0	select divide factor from register 0: select initial value 1: select divide factor from this register
select_high_wide_control_from_register_gen_reg14	[2]	RW	1'h0	select high wide control from register 0: select initial value 1: select high wide from this register
high_wide_control_gen_reg14	[1]	RW	1'h0	high wide control (when divider factor is odd) 0: low level of the clock is wider 1: high level of the clock is wider
divider_reset_control_gen_reg14	[0]	RW	1'h1	divider reset control 0: assert reset 1: de-assert reset

8.6.2.16 clock divider control register of divider for clk_ptp_ref_i_eth1 (0x6C)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:24]	RW	8'h0	reserved
clock_divider_factor_gen_reg15	[23:16]	RW	8'h0	clock divider factor
Reserved	[15:4]	RW	12'h0	reserved
select_divide_factor_from_register_gen_reg15	[3]	RW	1'h0	select divide factor from register 0: select initial value 1: select divide factor from this register
select_high_wide_control_from_register_gen_re	[2]	RW	1'h0	select high wide control from register 0: select initial value 1: select high wide from this register

g15				
high_wide_control_gen_reg15	[1]	RW	1'h0	high wide control (when divider factor is odd) 0: low level of the clock is wider 1: high level of the clock is wider
divider_reset_control_gen_reg15	[0]	RW	1'h1	divider reset control 0: assert reset 1: de-assert reset

8.6.2.17 clock divider control register of divider for clk_ref_eth1 (0x70)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:24]	RW	8'h0	reserved
clock_divider_factor_gen_reg16	[23:16]	RW	8'h0	clock divider factor
Reserved	[15:4]	RW	12'h0	reserved
select_divide_factor_from_register_gen_reg16	[3]	RW	1'h0	select divide factor from register 0: select initial value 1: select divide factor from this register
select_high_wide_control_from_register_gen_reg16	[2]	RW	1'h0	select high wide control from register 0: select initial value 1: select high wide from this register
high_wide_control_gen_reg16	[1]	RW	1'h0	high wide control (when divider factor is odd) 0: low level of the clock is wider 1: high level of the clock is wider
divider_reset_control_gen_reg16	[0]	RW	1'h1	divider reset control 0: assert reset 1: de-assert reset

8.6.2.18 clock divider control register of divider for clk_uart_500m (0x74)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:23]	RW	9'h0	reserved
clock_divider_factor_gen_reg17	[22:16]	RW	7'h0	clock divider factor
Reserved	[15:4]	RW	12'h0	reserved
select_divide_factor_from_register_gen_reg17	[3]	RW	1'h0	select divide factor from register 0: select initial value 1: select divide factor from this register
select_high_wide_control_from_register_gen_reg17	[2]	RW	1'h0	select high wide control from register 0: select initial value 1: select high wide from this register
high_wide_control_gen_reg17	[1]	RW	1'h0	high wide control (when divider factor is odd) 0: low level of the clock is wider 1: high level of the clock is wider
divider_reset_control_gen_reg17	[0]	RW	1'h1	divider reset control 0: assert reset 1: de-assert reset

8.6.2.19 clock divider control register of divider for clk_timer_1 (0x78)

Field Name	Bit	Type	Reset Value	Field Description
clock_divider_factor_gen_reg18	[31:16]	RW	16'h0	clock divider factor
Reserved	[15:4]	RW	12'h0	reserved
select_divide_factor_from_register_gen_reg18	[3]	RW	1'h0	select divide factor from register 0: select initial value 1: select divide factor from this register
select_high_wide_control_from_register_gen_reg18	[2]	RW	1'h0	select high wide control from register 0: select initial value 1: select high wide from this register
high_wide_control_gen_reg18	[1]	RW	1'h0	high wide control (when divider factor is odd) 0: low level of the clock is wider 1: high level of the clock is wider
divider_reset_control_gen_reg18	[0]	RW	1'h1	divider reset control 0: assert reset 1: de-assert reset

8.6.2.20 clock divider control register of divider for clk_timer_2 (0x7C)

Field Name	Bit	Type	Reset Value	Field Description
clock_divider_factor_gen_reg19	[31:16]	RW	16'h0	clock divider factor
Reserved	[15:4]	RW	12'h0	reserved
select_divide_factor_from_register_gen_reg19	[3]	RW	1'h0	select divide factor from register 0: select initial value 1: select divide factor from this register
select_high_wide_control_from_register_gen_reg19	[2]	RW	1'h0	select high wide control from register 0: select initial value 1: select high wide from this register
high_wide_control_gen_reg19	[1]	RW	1'h0	high wide control (when divider factor is odd) 0: low level of the clock is wider 1: high level of the clock is wider
divider_reset_control_gen_reg19	[0]	RW	1'h1	divider reset control 0: assert reset 1: de-assert reset

8.6.2.21 clock divider control register of divider for clk_timer_3(0x80)

Field Name	Bit	Type	Reset Value	Field Description
clock_divider_factor_gen_reg20	[31:16]	RW	16'h0	clock divider factor
Reserved	[15:4]	RW	12'h0	reserved
select_divide_factor_from_register_gen_reg20	[3]	RW	1'h0	select divide factor from register 0: select initial value 1: select divide factor from this register
select_high_wide_control_from_	[2]	RW	1'h0	select high wide control from register 0: select initial value

register_gen_reg20				1: select high wide from this register
high_wide_control_gen_reg20	[1]	RW	1'h0	high wide control (when divider factor is odd) 0: low level of the clock is wider 1: high level of the clock is wider
divider_reset_control_gen_reg20	[0]	RW	1'h1	divider reset control 0: assert reset 1: de-assert reset

8.6.2.22 clock divider control register of divider for clk_timer_4 (0x84)

Field Name	Bit	Type	Reset Value	Field Description
clock_divider_factor_gen_reg21	[31:16]	RW	16'h0	clock divider factor
Reserved	[15:4]	RW	12'h0	reserved
select_divide_factor_from_register_gen_reg21	[3]	RW	1'h0	select divide factor from register 0: select initial value 1: select divide factor from this register
select_high_wide_control_from_register_gen_reg21	[2]	RW	1'h0	select high wide control from register 0: select initial value 1: select high wide from this register
high_wide_control_gen_reg21	[1]	RW	1'h0	high wide control (when divider factor is odd) 0: low level of the clock is wider 1: high level of the clock is wider
divider_reset_control_gen_reg21	[0]	RW	1'h1	divider reset control 0: assert reset 1: de-assert reset

8.6.2.23 clock divider control register of divider for clk_timer_5 (0x88)

Field Name	Bit	Type	Reset Value	Field Description
clock_divider_factor_gen_reg22	[31:16]	RW	16'h0	clock divider factor
Reserved	[15:4]	RW	12'h0	reserved
select_divide_factor_from_register_gen_reg22	[3]	RW	1'h0	select divide factor from register 0: select initial value 1: select divide factor from this register
select_high_wide_control_from_register_gen_reg22	[2]	RW	1'h0	select high wide control from register 0: select initial value 1: select high wide from this register
high_wide_control_gen_reg22	[1]	RW	1'h0	high wide control (when divider factor is odd) 0: low level of the clock is wider 1: high level of the clock is wider
divider_reset_control_gen_reg22	[0]	RW	1'h1	divider reset control 0: assert reset 1: de-assert reset

8.6.2.24 clock divider control register of divider for clk_timer_6 (0x8C)

Field Name	Bit	Type	Reset Value	Field Description
clock_divider_factor_gen_reg23	[31:16]	RW	16'h0	clock divider factor
Reserved	[15:4]	RW	12'h0	reserved
select_divide_factor_from_register_gen_reg23	[3]	RW	1'h0	select divide factor from register 0: select initial value 1: select divide factor from this register
select_high_wide_control_from_register_gen_reg23	[2]	RW	1'h0	select high wide control from register 0: select initial value 1: select high wide from this register
high_wide_control_gen_reg23	[1]	RW	1'h0	high wide control (when divider factor is odd) 0: low level of the clock is wider 1: high level of the clock is wider
divider_reset_control_gen_reg23	[0]	RW	1'h1	divider reset control 0: assert reset 1: de-assert reset

8.6.2.25 clock divider control register of divider for clk_timer_7 (0x90)

Field Name	Bit	Type	Reset Value	Field Description
clock_divider_factor_gen_reg24	[31:16]	RW	16'h0	clock divider factor
Reserved	[15:4]	RW	12'h0	reserved
select_divide_factor_from_register_gen_reg24	[3]	RW	1'h0	select divide factor from register 0: select initial value 1: select divide factor from this register
select_high_wide_control_from_register_gen_reg24	[2]	RW	1'h0	select high wide control from register 0: select initial value 1: select high wide from this register
high_wide_control_gen_reg24	[1]	RW	1'h0	high wide control (when divider factor is odd) 0: low level of the clock is wider 1: high level of the clock is wider
divider_reset_control_gen_reg24	[0]	RW	1'h1	divider reset control 0: assert reset 1: de-assert reset

8.6.2.26 clock divider control register of divider for clk_timer_8 (0x94)

Field Name	Bit	Type	Reset Value	Field Description
clock_divider_factor_gen_reg25	[31:16]	RW	16'h0	clock divider factor
Reserved	[15:4]	RW	12'h0	reserved
select_divide_factor_from_register_gen_reg25	[3]	RW	1'h0	select divide factor from register 0: select initial value 1: select divide factor from this register
select_high_wide_control_from_	[2]	RW	1'h0	select high wide control from register 0: select initial value

register_gen_reg25				1: select high wide from this register
high_wide_control_gen_reg25	[1]	RW	1'h0	high wide control (when divider factor is odd) 0: low level of the clock is wider 1: high level of the clock is wider
divider_reset_control_gen_reg25	[0]	RW	1'h1	divider reset control 0: assert reset 1: de-assert reset

8.6.2.27 clock divider control register of divider for clk_efuse (0x98)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:23]	RW	9'h0	reserved
clock_divider_factor_gen_reg26	[22:16]	RW	7'h0	clock divider factor
Reserved	[15:4]	RW	12'h0	reserved
select_divide_factor_from_register_gen_reg26	[3]	RW	1'h0	select divide factor from register 0: select initial value 1: select divide factor from this register
select_high_wide_control_from_register_gen_reg26	[2]	RW	1'h0	select high wide control from register 0: select initial value 1: select high wide from this register
high_wide_control_gen_reg26	[1]	RW	1'h0	high wide control (when divider factor is odd) 0: low level of the clock is wider 1: high level of the clock is wider
divider_reset_control_gen_reg26	[0]	RW	1'h1	divider reset control 0: assert reset 1: de-assert reset

8.6.2.28 clock divider control register of divider for clk_gpio_db (0x9C)

Field Name	Bit	Type	Reset Value	Field Description
clock_divider_factor_gen_reg27	[31:16]	RW	16'h0	clock divider factor
Reserved	[15:4]	RW	12'h0	reserved
select_divide_factor_from_register_gen_reg27	[3]	RW	1'h0	select divide factor from register 0: select initial value 1: select divide factor from this register
select_high_wide_control_from_register_gen_reg27	[2]	RW	1'h0	select high wide control from register 0: select initial value 1: select high wide from this register
high_wide_control_gen_reg27	[1]	RW	1'h0	high wide control (when divider factor is odd) 0: low level of the clock is wider 1: high level of the clock is wider
divider_reset_control_gen_reg27	[0]	RW	1'h1	divider reset control 0: assert reset 1: de-assert reset

8.6.2.29 clock divider control register of divider 0 for clk_tpu (0xA0)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:21]	RW	11'h0	reserved
clock_divider_factor_gen_reg28	[20:16]	RW	5'h0	clock divider factor
Reserved	[15:4]	RW	12'h0	reserved
select_divide_factor_from_register_gen_reg28	[3]	RW	1'h0	select divide factor from register 0: select initial value 1: select divide factor from this register
select_high_wide_control_from_register_gen_reg28	[2]	RW	1'h0	select high wide control from register 0: select initial value 1: select high wide from this register
high_wide_control_gen_reg28	[1]	RW	1'h0	high wide control (when divider factor is odd) 0: low level of the clock is wider 1: high level of the clock is wider
divider_reset_control_gen_reg28	[0]	RW	1'h1	divider reset control 0: assert reset 1: de-assert reset

8.6.2.30 clock divider control register of divider 1 for clk_tpu (0xA4)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:21]	RW	11'h0	reserved
clock_divider_factor_gen_reg29	[20:16]	RW	5'h0	clock divider factor
Reserved	[15:4]	RW	12'h0	reserved
select_divide_factor_from_register_gen_reg29	[3]	RW	1'h0	select divide factor from register 0: select initial value 1: select divide factor from this register
select_high_wide_control_from_register_gen_reg29	[2]	RW	1'h0	select high wide control from register 0: select initial value 1: select high wide from this register
high_wide_control_gen_reg29	[1]	RW	1'h0	high wide control (when divider factor is odd) 0: low level of the clock is wider 1: high level of the clock is wider
divider_reset_control_gen_reg29	[0]	RW	1'h1	divider reset control 0: assert reset 1: de-assert reset

8.6.2.31 clock divider control register of divider for clk_fixed_tpu_clk (0xA8)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:21]	RW	11'h0	reserved
clock_divider_factor_gen_reg30	[20:16]	RW	5'h0	clock divider factor
Reserved	[15:4]	RW	12'h0	reserved
select_divide_factor_from_regist	[3]	RW	1'h0	select divide factor from register 0: select initial value

er_gen_reg30				1: select divide factor from this register
select_high_wide_control_from_register_gen_reg30	[2]	RW	1'h0	select high wide control from register 0: select initial value 1: select high wide from this register
high_wide_control_gen_reg30	[1]	RW	1'h0	high wide control (when divider factor is odd) 0: low level of the clock is wider 1: high level of the clock is wider
divider_reset_control_gen_reg30	[0]	RW	1'h1	divider reset control 0: assert reset 1: de-assert reset

8.6.2.32 clock divider control register of divider 0 for clk_axi_vde_wave (0xAC)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:21]	RW	11'h0	reserved
clock_divider_factor_gen_reg31	[20:16]	RW	5'h0	clock divider factor
Reserved	[15:4]	RW	12'h0	reserved
select_divide_factor_from_register_gen_reg31	[3]	RW	1'h0	select divide factor from register 0: select initial value 1: select divide factor from this register
select_high_wide_control_from_register_gen_reg31	[2]	RW	1'h0	select high wide control from register 0: select initial value 1: select high wide from this register
high_wide_control_gen_reg31	[1]	RW	1'h0	high wide control (when divider factor is odd) 0: low level of the clock is wider 1: high level of the clock is wider
divider_reset_control_gen_reg31	[0]	RW	1'h1	divider reset control 0: assert reset 1: de-assert reset

8.6.2.33 clock divider control register of divider 1 for clk_axi_vde_wave (0xB0)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:21]	RW	11'h0	reserved
clock_divider_factor_gen_reg32	[20:16]	RW	5'h0	clock divider factor
Reserved	[15:4]	RW	12'h0	reserved
select_divide_factor_from_register_gen_reg32	[3]	RW	1'h0	select divide factor from register 0: select initial value 1: select divide factor from this register
select_high_wide_control_from_register_gen_reg32	[2]	RW	1'h0	select high wide control from register 0: select initial value 1: select high wide from this register
high_wide_control_gen_reg32	[1]	RW	1'h0	high wide control (when divider factor is odd) 0: low level of the clock is wider 1: high level of the clock is wider
divider_reset_control_gen_reg32	[0]	RW	1'h1	divider reset control 0: assert reset 1: de-assert reset

8.6.2.34 clock divider control register of divider for clk_axi3 (0xB4)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:21]	RW	11'h0	reserved
clock_divider_factor_gen_reg33	[20:16]	RW	5'h0	clock divider factor
Reserved	[15:4]	RW	12'h0	reserved
select_divide_factor_from_register_gen_reg33	[3]	RO	1'h0	select divide factor from register this bit is reserved for this divider.
select_high_wide_control_from_register_gen_reg33	[2]	RO	1'h0	select high wide control from register this bit is reserved for this divider.
high_wide_control_gen_reg33	[1]	RW	1'h0	high wide control (when divider factor is odd) 0: low level of the clock is wider 1: high level of the clock is wider
divider_reset_control_gen_reg33	[0]	RO	1'h1	divider reset control this bit is reserved for this divider.

8.6.2.35 clock divider control register of divider for clk_axi6 (0xB8)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:21]	RW	11'h0	reserved
clock_divider_factor_gen_reg34	[20:16]	RW	5'h0	clock divider factor
Reserved	[15:4]	RW	12'h0	reserved
select_divide_factor_from_register_gen_reg34	[3]	RW	1'h0	select divide factor from register 0: select initial value 1: select divide factor from this register
select_high_wide_control_from_register_gen_reg34	[2]	RW	1'h0	select high wide control from register 0: select initial value 1: select high wide from this register
high_wide_control_gen_reg34	[1]	RW	1'h0	high wide control (when divider factor is odd) 0: low level of the clock is wider 1: high level of the clock is wider
divider_reset_control_gen_reg34	[0]	RW	1'h1	divider reset control 0: assert reset 1: de-assert reset

8.6.2.36 clock divider control register of divider for clk_axi8 (0xBC)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:21]	RW	11'h0	reserved
clock_divider_factor_gen_reg35	[20:16]	RW	5'h0	clock divider factor
Reserved	[15:4]	RW	12'h0	reserved

select_divide_factor_from_register_gen_reg35	[3]	RW	1'h0	select divide factor from register 0: select initial value 1: select divide factor from this register
select_high_wide_control_from_register_gen_reg35	[2]	RW	1'h0	select high wide control from register 0: select initial value 1: select high wide from this register
high_wide_control_gen_reg35	[1]	RW	1'h0	high wide control (when divider factor is odd) 0: low level of the clock is wider 1: high level of the clock is wider
divider_reset_control_gen_reg35	[0]	RW	1'h1	divider reset control 0: assert reset 1: de-assert reset

8.6.2.37 clock divider control register of divider 0 for clk_axi10 (0xC0)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:21]	RW	11'h0	reserved
clock_divider_factor_gen_reg36	[20:16]	RW	5'h0	clock divider factor
Reserved	[15:4]	RW	12'h0	reserved
select_divide_factor_from_register_gen_reg36	[3]	RW	1'h0	select divide factor from register 0: select initial value 1: select divide factor from this register
select_high_wide_control_from_register_gen_reg36	[2]	RW	1'h0	select high wide control from register 0: select initial value 1: select high wide from this register
high_wide_control_gen_reg36	[1]	RW	1'h0	high wide control (when divider factor is odd) 0: low level of the clock is wider 1: high level of the clock is wider
divider_reset_control_gen_reg36	[0]	RW	1'h1	divider reset control 0: assert reset 1: de-assert reset

8.6.2.38 clock divider control register of divider 1 for clk_axi10 (0xC4)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:21]	RW	11'h0	reserved
clock_divider_factor_gen_reg37	[20:16]	RW	5'h0	clock divider factor
Reserved	[15:4]	RW	12'h0	reserved
select_divide_factor_from_register_gen_reg37	[3]	RW	1'h0	select divide factor from register 0: select initial value 1: select divide factor from this register
select_high_wide_control_from_register_gen_reg37	[2]	RW	1'h0	select high wide control from register 0: select initial value 1: select high wide from this register
high_wide_control_gen_reg37	[1]	RW	1'h0	high wide control (when divider factor is odd) 0: low level of the clock is wider 1: high level of the clock is wider
divider_reset_co	[0]	RW	1'h1	divider reset control

ntrol_gen_reg37				0: assert reset 1: de-assert reset

8.6.2.39 clock divider control register of divider 0 for clk_axi_ddr (0xC8)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:21]	RW	11'h0	reserved
clock_divider_factor_gen_reg38	[20:16]	RW	5'h0	clock divider factor
Reserved	[15:4]	RW	12'h0	reserved
select_divide_factor_from_register_gen_reg38	[3]	RW	1'h0	select divide factor from register 0: select initial value 1: select divide factor from this register
select_high_wide_control_from_register_gen_reg38	[2]	RW	1'h0	select high wide control from register 0: select initial value 1: select high wide from this register
high_wide_control_gen_reg38	[1]	RW	1'h0	high wide control (when divider factor is odd) 0: low level of the clock is wider 1: high level of the clock is wider
divider_reset_control_gen_reg38	[0]	RW	1'h1	divider reset control 0: assert reset 1: de-assert reset

8.6.2.40 clock divider control register of divider 1 for clk_axi_ddr (0xCC)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:21]	RW	11'h0	reserved
clock_divider_factor_gen_reg39	[20:16]	RW	5'h0	clock divider factor
Reserved	[15:4]	RW	12'h0	reserved
select_divide_factor_from_register_gen_reg39	[3]	RW	1'h0	select divide factor from register 0: select initial value 1: select divide factor from this register
select_high_wide_control_from_register_gen_reg39	[2]	RW	1'h0	select high wide control from register 0: select initial value 1: select high wide from this register
high_wide_control_gen_reg39	[1]	RW	1'h0	high wide control (when divider factor is odd) 0: low level of the clock is wider 1: high level of the clock is wider
divider_reset_control_gen_reg39	[0]	RW	1'h1	divider reset control 0: assert reset 1: de-assert reset

8.6.2.41 clock divider control register of divider 0 for clk_ddr0 (0xD0)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:21]	RW	11'h0	reserved

clock_divider_factor_gen_reg40	[20:16]	RW	5'h0	clock divider factor
Reserved	[15:4]	RW	12'h0	reserved
select_divide_factor_from_register_gen_reg40	[3]	RW	1'h0	select divide factor from register 0: select initial value 1: select divide factor from this register
select_high_wide_control_from_register_gen_reg40	[2]	RW	1'h0	select high wide control from register 0: select initial value 1: select high wide from this register
high_wide_control_gen_reg40	[1]	RW	1'h0	high wide control (when divider factor is odd) 0: low level of the clock is wider 1: high level of the clock is wider
divider_reset_control_gen_reg40	[0]	RW	1'h1	divider reset control 0: assert reset 1: de-assert reset

8.6.2.42 clock divider control register of divider 1 for clk_ddr0 (0xD4)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:21]	RW	11'h0	reserved
clock_divider_factor_gen_reg41	[20:16]	RW	5'h0	clock divider factor
Reserved	[15:4]	RW	12'h0	reserved
select_divide_factor_from_register_gen_reg41	[3]	RW	1'h0	select divide factor from register 0: select initial value 1: select divide factor from this register
select_high_wide_control_from_register_gen_reg41	[2]	RW	1'h0	select high wide control from register 0: select initial value 1: select high wide from this register
high_wide_control_gen_reg41	[1]	RW	1'h0	high wide control (when divider factor is odd) 0: low level of the clock is wider 1: high level of the clock is wider
divider_reset_control_gen_reg41	[0]	RW	1'h1	divider reset control 0: assert reset 1: de-assert reset

8.6.2.43 clock divider control register of divider 0 for clk_ddr12 (0xD8)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:21]	RW	11'h0	reserved
clock_divider_factor_gen_reg42	[20:16]	RW	5'h0	clock divider factor
Reserved	[15:4]	RW	12'h0	reserved
select_divide_factor_from_register_gen_reg42	[3]	RW	1'h0	select divide factor from register 0: select initial value 1: select divide factor from this register
select_high_wide_control_from_register_gen_reg42	[2]	RW	1'h0	select high wide control from register 0: select initial value 1: select high wide from this register
high_wide_contr	[1]	RW	1'h0	high wide control (when divider factor is odd)

ol_gen_reg42				0: low level of the clock is wider 1: high level of the clock is wider
divider_reset_control_gen_reg42	[0]	RW	1'h1	divider reset control 0: assert reset 1: de-assert reset

8.6.2.44 clock divider control register of divider 1 for clk_ddr12 (0xDC)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:21]	RW	11'h0	reserved
clock_divider_factor_gen_reg43	[20:16]	RW	5'h0	clock divider factor
Reserved	[15:4]	RW	12'h0	reserved
select_divide_factor_from_register_gen_reg43	[3]	RW	1'h0	select divide factor from register 0: select initial value 1: select divide factor from this register
select_high_wide_control_from_register_gen_reg43	[2]	RW	1'h0	select high wide control from register 0: select initial value 1: select high wide from this register
high_wide_control_gen_reg43	[1]	RW	1'h0	high wide control (when divider factor is odd) 0: low level of the clock is wider 1: high level of the clock is wider
divider_reset_control_gen_reg43	[0]	RW	1'h1	divider reset control 0: assert reset 1: de-assert reset

8.7 Reset Generation Control and Status Registers

8.7.1 Reset Generation Control and Status Registers Summary

Table 17 Reset Generation Control and Status Registers Summary

Offset	Register Name	Reg Description
0x0	sw_reg0	software reset register 0
0x4	sw_reg1	software reset register 1
0x8	sw_reg2	software reset register 2

8.7.2 Reset Generation Control and Status Registers Description

8.7.2.1 software reset register 0 （0x0）

Field Name	Bit	Type	Reset Value	Field Description
sw_rst_gpio0	[31]	RW	1'h1	software reset for gpio0 (active low)
sw_rst_spi	[30]	RW	1'h1	software reset for spi (active low) (reserved in bm1684.)

sw_rst_pwm	[29]	RW	1'h1	software reset for pwm (active low)
sw_rst_i2c2	[28]	RW	1'h1	software reset for i2c2 (active low)
sw_rst_i2c1	[27]	RW	1'h1	software reset for i2c1 (active low)
sw_rst_i2c0	[26]	RW	1'h1	software reset for i2c0 (active low)
sw_rst_uart2	[25]	RW	1'h1	software reset for uart2 (active low)
sw_rst_uart1	[24]	RW	1'h1	software reset for uart1 (active low)
sw_rst_uart0	[23]	RW	1'h1	software reset for uart0 (active low)
sw_rst_sdma	[22]	RW	1'h1	software reset for sdma (active low)
sw_rst_sd	[21]	RW	1'h1	software reset for sd (active low)
sw_rst_emmc	[20]	RW	1'h1	software reset for emmc (active low)
sw_rst_eth1	[19]	RW	1'h1	software reset for eth1 (active low)
sw_rst_eth0	[18]	RW	1'h1	software reset for eth0 (active low)
sw_rst_tpu	[17]	RW	1'h1	software reset for tpu (active low)
sw_rst_axi_sram	[16]	RW	1'h1	software reset for axi_sram (active low)
sw_rst_gdma	[15]	RW	1'h1	software reset for gdma (active low)
sw_rst_ddr2_power_ok	[14]	RW	1'h0	software reset for ddr2 power ok (active high)
sw_rst_ddr1_power_ok	[13]	RW	1'h0	software reset for ddr1 power ok (active high)
sw_rst_ddr0_b_power_ok	[12]	RW	1'h0	software reset for ddr0_b power ok (active high)
sw_rst_ddr0_a_power_ok	[11]	RW	1'h0	software reset for ddr0_a power ok (active high)
sw_rst_ddr2_apb	[10]	RW	1'h0	software reset for ddr2 apb (active low)
sw_rst_ddr1_apb	[9]	RW	1'h0	software reset for ddr1 apb (active low)
sw_rst_ddr0_b_apb	[8]	RW	1'h0	software reset for ddr0_b apb (active low)
sw_rst_ddr0_a_apb	[7]	RW	1'h0	software reset for ddr0_a apb (active low)
sw_rst_ddr2_axi	[6]	RW	1'h0	software reset for ddr2 axi (active low)
sw_rst_ddr1_axi	[5]	RW	1'h0	software reset for ddr1 axi (active low)
sw_rst_ddr0_b_axi	[4]	RW	1'h0	software reset for ddr0_b axi (active low)
sw_rst_ddr0_a_axi	[3]	RW	1'h0	software reset for ddr0_a axi (active low)
sw_rst_riscv	[2]	RW	1'h0	software reset for riscv (active low)
sw_rst_second_ap	[1]	RW	1'h0	software reset for second_ap (active low) (ap lite reset)
sw_rst_main_ap	[0]	RW	1'h1	software reset for main_ap (active low)

8.7.2.2 software reset register 1 (0x4)

Field Name	Bit	Type	Reset Value	Field Description
sw_rst_video_system_1_jpeg1	[31]	RW	1'h1	software reset for video system 1 jpeg1 (active low)
sw_rst_video_system_1_jpeg0	[30]	RW	1'h1	software reset for video system 1 jpeg0 (active low)
sw_rst_video_system_1_wave1	[29]	RW	1'h1	software reset for video system 1 wave1 (active low)
sw_rst_video_system_1_wave0	[28]	RW	1'h1	software reset for video system 1 wave0 (active low)
sw_rst_video_system_0_vmmu_dma	[27]	RW	1'h1	software reset for video system 0 vmmu_dma (active low)

sw_rst_video_system_0_wmmu1	[26]	RW	1'h1	software reset for video system 0 wmmu1 (active low)
sw_rst_video_system_0_wmmu0	[25]	RW	1'h1	software reset for video system 0 wmmu0 (active low)
sw_rst_video_system_0_jmmu	[24]	RW	1'h1	software reset for video system 0 jmmu (active low)
sw_rst_video_system_0_vmmu	[23]	RW	1'h1	software reset for video system 0 vmmu (active low)
sw_rst_video_system_0_vpp	[22]	RW	1'h1	software reset for video system 0 vpp (active low)
sw_rst_video_system_0_jpeg1	[21]	RW	1'h1	software reset for video system 0 jpeg1 (active low)
sw_rst_video_system_0_jpeg0	[20]	RW	1'h1	software reset for video system 0 jpeg0 (active low)
sw_rst_video_system_0_wave1	[19]	RW	1'h1	software reset for video system 0 wave1 (active low)
sw_rst_video_system_0_wave0	[18]	RW	1'h1	software reset for video system 0 wave0 (active low)
sw_rst_trng	[17]	RW	1'h1	software reset for trng (active low)
sw_rst_spacc	[16]	RW	1'h1	software reset for spacc (active low)
sw_rst_pka	[15]	RW	1'h1	software reset for pka (active low)
sw_rst_dbg_i2c	[14]	RW	1'h1	software reset for dbg_i2c (active low)
sw_rst_pcie	[13]	RW	1'h1	software reset for pcie (active low)
sw_rst_mmu	[12]	RW	1'h1	software reset for mmu (active low)
sw_rst_cdma	[11]	RW	1'h1	software reset for cdma (active low)
sw_rst_intc3	[10]	RW	1'h1	software reset for intc3 (active low)
sw_rst_intc2	[9]	RW	1'h1	software reset for intc2 (active low)
sw_rst_intc1	[8]	RW	1'h1	software reset for intc1 (active low)
sw_rst_intc0	[7]	RW	1'h1	software reset for intc0 (active low)
sw_rst_timer	[6]	RW	1'h1	software reset for timer (active low)
sw_rst_spic	[5]	RW	1'h1	software reset for spic (active low)
sw_rst_ahb_rom	[4]	RW	1'h1	software reset for ahb_rom (active low)
sw_rst_wdt	[3]	RW	1'h1	software reset for wdt (active low)
sw_rst_efuse	[2]	RW	1'h1	software reset for efuse (active low)
sw_rst_gpio2	[1]	RW	1'h1	software reset for gpio2 (active low)
sw_rst_gpio1	[0]	RW	1'h1	software reset for gpio1 (active low)

8.7.2.3 software reset register 2 (0x8)

Field Name	Bit	Type	Reset Value	Field Description
Reserved	[31:17]	RW	15'h0	reserved
sw_rst_tsdma	[16]	RW	1'h1	software reset for tsdma (active low)
sw_rst_hau_ngs	[15]	RW	1'h1	software reset for hau ngs (active low)
sw_rst_video_system_1_debug_reset	[14]	RW	1'h1	software reset for video system 1 debug reset (active low) (assert this reset only when all vd1 ips are idle.)
sw_rst_video_system_0_debug_reset	[13]	RW	1'h1	software reset for video system 0 debug reset (active low) (assert this reset only when all vd0 ips are idle.)
sw_rst_video_encoder_subsys_debug_reset	[12]	RW	1'h1	software reset for video encoder subsys debug reset (active low) (assert this reset only when all vde ips are idle.)
sw_rst_video_encoder_subsys_wm	[11]	RW	1'h1	software reset for video encoder subsys wmmu (active low)

mu				
sw_rst_video_encoder_subsys_wave521	[10]	RW	1'h1	software reset for video encoder subsys wave521 (active low)
sw_rst_ddr2_dfi_clock_div	[9]	RW	1'h1	software reset for ddr2 dfi clock div (active low)
sw_rst_ddr1_dfi_clock_div	[8]	RW	1'h1	software reset for ddr1 dfi clock div (active low)
sw_rst_ddr0_b_dfi_clock_div	[7]	RW	1'h1	software reset for ddr0_b dfi clock div (active low)
sw_rst_ddr0_a_dfi_clock_div	[6]	RW	1'h1	software reset for ddr0_a dfi clock div (active low)
sw_rst_video_system_1_vmmu_dma	[5]	RW	1'h1	software reset for video system 1 vmmu_dma (active low)
sw_rst_video_system_1_wmmu1	[4]	RW	1'h1	software reset for video system 1 wmmu1 (active low)
sw_rst_video_system_1_wmmu0	[3]	RW	1'h1	software reset for video system 1 wmmu0 (active low)
sw_rst_video_system_1_jmmu	[2]	RW	1'h1	software reset for video system 1 jmmu (active low)
sw_rst_video_system_1_vmmu	[1]	RW	1'h1	software reset for video system 1 vmmu (active low)
sw_rst_video_system_1_vpp	[0]	RW	1'h1	software reset for video system 1 vpp (active low)