SG2002

Technical Reference Manual

Version : 1.0

Release Date : 2024-06-17

Contents

Revision History

Table 1: Revision History

Revision	Date	Description
1.0-alpha	2023/11/23	Initial Version
1.0-beta	2024/02/26	Converted to reStructuredText
1.0	2024/06/17	Fixed cumulative issues and officially released v1.0

CHAPTER

ONE

DISCLAIMER

Contact Us

Official website
https://www.sophgo.com/

Document publishing website
https://github.com/sophgo/sophgo-doc

CHAPTER

TWO

SYSTEM OVERVIEW

2.1 Introduction

SG2002 is a high-performance, low-power chip launched for the AIot field. It has multiple built-in powerful cores: 2 x C906, 1 x Arm Cortex A53, 1 x 8051. Users can quickly switch cores according to needs. At the same time, it integrates hardware codecs: H.264 video compression codec, H.265 video compression encoder and ISP, supporting the configuration of professional-level video image ISP: HDR wide dynamic range, 3D noise reduction, defogging, lens distortion correction and other image enhancement and correction algorithms.

The chip also integrates a self-developed TPU, which can provide 1.0TOPS IN8 computing power. The specially designed TPU scheduling engine efficiently delivers extremely high-bandwidth data streams to all tensor processor cores.

At the same time, it supports a variety of peripheral interfaces: 5 x I2C, 2 x SDIO3.0, 2 x I2S, 15 x PWM, 1 x USB 2.0, etc., to meet the needs of various scenarios.

2.2 System Framework

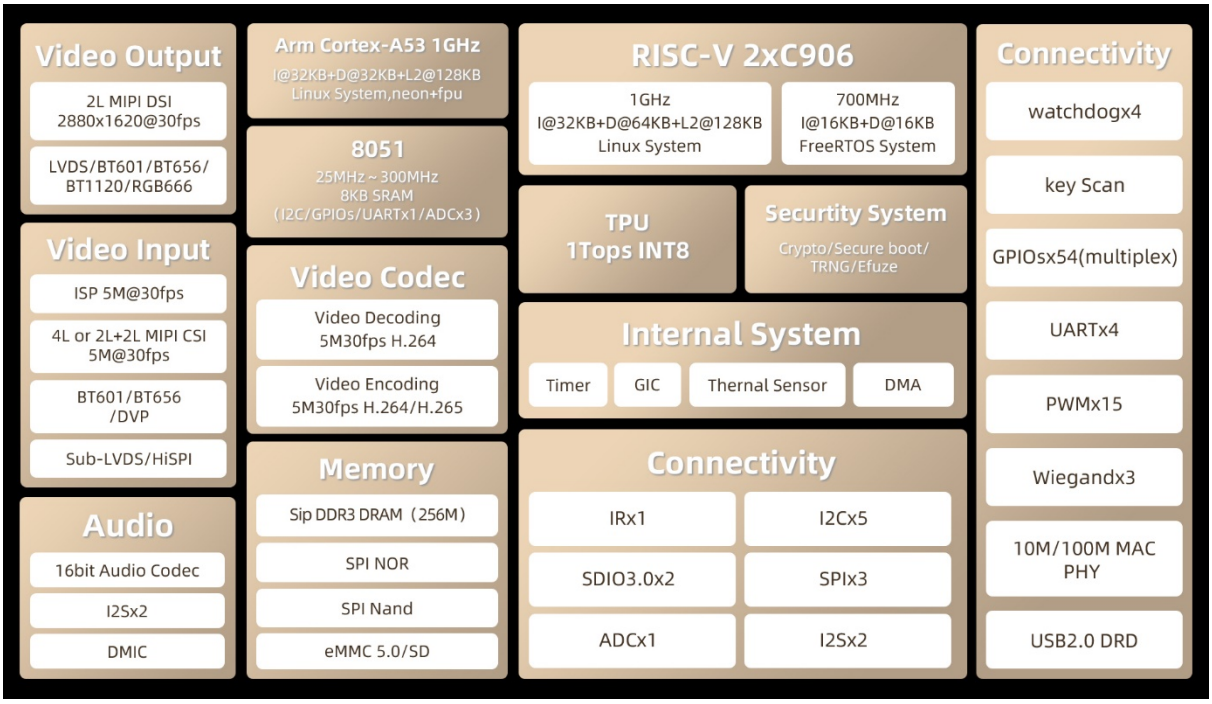


Diagram 2.1: System Framework

2.3 Features

2.3.1 Processor Cores

- Main processor RISCV C906 @ 1.0Ghz
  - 32KB I-cache, 64KB D-Cache
  - Integrated Vector and Floating Point Unit (FPU)
- Main processor ARM Cortex-A53 @ 1.0Ghz
  - 32KB I-cache, 32KB D-Cache
  - 128KB L2 cache
  - Support Neon and Floating Point Unit (FPU)
- Coprocessor RISCV C906 @ 700Mhz
  - Integrated Floating Point Unit (FPU)

2.3.2 TPU

- Built-in TPU, computing power reaches ~1.0TOPS INT8.
- Supports mainstream neural network architectures: Caffe, Pytorch, TensorFlow(Lite), ONNX and MXNet.
- Support Pedestrian Detection, Face Detection, Face recognition, Face anti-spoofing and other video structured applications.

2.3.3 Video codec

- H.264 Baseline/Main/High profile
- H.265 Main profile
- H.264/H.265 both support I frames and P frames
- MJPEG/JPEG baseline
- H.264 codec maximum resolution: 2880x1620 (5M)
- H.265 encoding maximum resolution: 2880x1620 (5M)
- H.264 codec performance
  - 2880x1620@30fps + 720x576@30fps
  - 1920x1080@30fps encoding + 1920x1080@30fps decoding
- H.265 encoding performance
  - 2880x1620@30fps + 720x576@30fps
- JPEG maximum codec performance
  - 2880x1620@30fps
- Supports multiple rate control modes such as CBR/VBR/FIXQP.
- Supports Region of Interest (ROI) encoding

2.3.4 Video Interface

- Input
  - Supports two simultaneous video inputs (mipi 2L + 1L)
  - Supports MIPI, Sub-LVDS, HiSPI and other serial interfaces
  - Support 8/10/12 bit RGB Bayer video input
  - Support BT.656
  - Support AHD multi-channel mixed BT format
  - Supports SONY, OnSemi, OmniVision and other high-definition CMOS sensors
  - Provide programmable frequency output for sensor as reference clock
  - Supports a maximum width of 2880 and a maximum resolution of 5M (2688x1944, 2880x1620)
- Output
  - Supports multiple serial and parallel screen display specifications
  - Support serial interfaces such as MIPI

  - Supports BT656, BT601 (8bit), BT1120, 8080 and other parallel interfaces
  - Support SPI output interface

2.3.5 ISP and Image processing

- 90 degree, 180 degree, 270 degree rotation for image and video
- Mirror and Flip functions for image and video
- Video 2-layer OSD overlay
- Video 1/32 ~ 32x zoom function
- 3A (AE/AWB/AF) algorithm
- Fixed mode noise elimination, dead pixel correction
- Lens shading correction, lens distortion correction, purple fringing correction
- Direction adaptive demosaic
- Gamma correction, (regional/global) dynamic contrast enhancement, color management and enhancement
- Area adaptive defogging
- Bayer noise reduction, 3D noise reduction, detail enhancement and sharpening enhancement
- Local Tone mapping
- Sensor self-bandwidth dynamic and 2 frame wide dynamic
- Two-axis digital image stabilization
- Lens distortion correction
- Provide PC-side ISP tuning tools

2.3.6 Hardware acceleration engine

- Mixed hardware and software mode supports part of OpenCV libraries
- Mixed hardware and software mode supports part of IVE libraries

2.3.7 Audio Codec

- Integrated Audio CODEC, supports 16-bit audio source/voice input and output
- Integrated mono microphone input
- Integrated mono output. (An external amplifier is required to drive the speaker)
- Internally integrated another microphone is directly connected to the output channel to facilitate AEC implementation
- Software audio codec protocols (G.711, G.726, ADPCM)
- Software supports audio 3A (AEC, ANR, AGC) function

2.3.8 Network Interface

- The Ethernet module provides one Ethernet MAC to receive and send network data.
- Ethernet MAC with built-in 10/100Mbps Fast Ethernet Transceiver, can work in 10/100Mbps full-duplex or half-duplex mode.

2.3.9 Security System Module

- Hardware implements AES/DES/SM4 multiple encryption and decryption algorithms
- Hardware implementation of HASH (SHA1/SHA256) hash algorithm
- Hardware implemented random number generator
- Internally integrated 2Kbit eFuse logical space

2.3.10 Intelligent and safe operating environment

- Support the establishment of trust chain: Provide the foundation of a secure environment and the foundation of a trusted environment, such as hardware security settings and root of trust.
- Supports secure boot and provides security hardware and software protection functions.
- Support data encryption security: data encryption program, computing core encryption.
- Support software and firmware signature verification process: verify software credibility and integrity, including booting and loading signature verification procedures.
- Support secure storage and transmission: protect external data storage and exchange.
- Support security updates.

2.3.11 Peripheral interface

- Integrated POR, Power sequence.
- 4 single-ended ADCs (3 No-die domain).
- 6 I2C (1 No-die domain).
- 3 SPIs.
- 5 sets of UART (1 No-die domain).
- 4 sets (15 channels) PWM.
- 2 SDIO interfaces:
  - One supports connected with SD 3.0 Card under 3V (supports maximum capacity SDXC 2TB, supported speed is UHS-I).
  - One supports connected with other SDIO 3.0 devices under 1.8V/3.0V (supported speed is UHS-I).
- 66 GPIO interfaces (14 no die domain).
- Integrate keyscan and Wiegand.
- Integrated MAC PHY supports 10/100Mbps full-duplex or half-duplex mode.
- One USB Host/device interface.

2.3.12 External memory interface

- Built-in DRAM.
  - DDR3 16bit x 1, maximum speed up to 1866Mbps, capacity 2Gbit (256MB).
- SPI NOR flash interface (1.8V / 3.0V).
  - Supports 1, 2, 4 wire modes.
  - Maximum support is 256MByte.
- SPI Nand flash interface (1.8V / 3.0V).
  - Supports 1KB/2KB/4KB page (corresponding maximum capacity 16GB/32GB/64GB).
  - Use the ECC module built into the device itself.
- eMMC 4.5 interface (1.8V/3.0V) SD0 EMMC has a common power supply. Because the SD card defaults to 3V, it is not suitable to connect to 1.8V eMMC when there is an SD card.
  - 4 bit interface.
  - Support HS200.
  - Maximum supported capacity 2TB.

2.3.13 Chip physical specifications

- Power consumption
  - 1080P + Video encode + AI : ~ 500mW
- Operating Voltage
  - Core voltage is 0.9V
  - IO voltage is 1.8V and 3.0V
  - DDR voltage is as shown in the table below
    - 1.35V
- Encapsulation
  - Using QFN package, the package size is 9mmx9mmx0.9mm. The pin pitch is 0.35mm. The total number of pins is 88.

CHAPTER

THREE

SYSTEM ARCHITECTURE

3.1 Interrupt Subsystem

Table 3.1: Interrupt number and Interrupt source mapping for ARM Cortex-A53

INT #	INT Source	INT	INT Source	INT #	INT Source
32	TEMPSENS	70	SPI1	108	TRNG
33	RTC Alarm	71	SPI2	109	ddr_axi_mon
34	RTC Longpress	72	SPI3	110	ddr_pi_phy
35	VBAT DET	73	SPI4	111	SPI_NOR
36	JPEG	74	WatchDog0	112	EPHY
37	H264	75	KEYSCAN	113	IVE
38	H265	76	GPIO0	114	Reserved
39	VC SBM	77	GPIO1	115	Reserved
40	ISP	78	GPIO2	116	SARADC
41	SC_TOP	79	GPIO3	117	mbox
42	CSI_MAC0	80	Wiegand0	151	npmuirq[0]
43	CSI_MAC1	81	Wiegand1	152	ctiirq[0]
44	LDC	82	Wiegand2	159	nexterrirq
45	System DMA	83	RTC MBOX	116	SARADC
46	USB	84	N/A		
47	Ethnet0	85	RTC IRRX		
48	Ethnet0	86	RTC GPIO		
49	EMMC Wakup	87	RTC UART		
50	EMMC	88	RTC SPI_NOR		
51	SD0 Wakup	89	RTC I2C		
52	SD0	90	RTC WDG		
53	SD1 Wakup	91	TPU		
54	SD1	92	TDMA		

continues on next page

Table 3.1 – continued from previous page

INT #	INT Source	INT	INT Source	INT #	INT Source
55	SPI_NAND	93	Reserved		
56	I2S0	94	Reserved		
57	I2S1	95	Timer0		
58	I2S2	96	Timer1		
59	I2S3	97	Timer2		
60	UART0	98	Timer3		
61	UART1	99	Timer4		
62	UART2	100	Timer5		
63	UART3	101	Timer6		
64	UART4	102	Timer7		
65	I2C0	103	peri_firewall		
66	I2C1	104	hsperi_firewall		
67	I2C2	105	ddr_fw		
68	I2C3	106	rom_firewall		
69	I2C4	107	SPACC		

Table 3.2: Interrupt number and Interrupt source mapping for Master RISCV C906 @ 1.0Ghz

INT #	INT Source	INT #	INT Source	INT #	INT Source
16	TEMPSENS	52	I2C3	88	hsperi_firewall
17	RTC Alarm	53	I2C4	89	ddr_fw
18	RTC Longpress	54	SPI1	90	rom_firewall
19	VBAT DET	55	SPI2	91	SPACC
20	JPEG	56	SPI3	92	TRNG
21	H264	57	SPI4	93	ddr_axi_mon
22	H265	58	WatchDog1	94	ddr_pi_phy
23	VC SBM	59	KEYSCAN	95	SPI_NOR
24	ISP	60	GPIO0	96	EPHY
25	SC_TOP	61	GPIO1	97	IVE
26	CSI_MAC0	62	GPIO2	98	Reserved
27	CSI_MAC1	63	GPIO3	99	Reserved
28	LDC	64	Wiegand0	100	SARADC
29	System DMA	65	Wiegand1	101	mbox

continues on next page

Table 3.2 – continued from previous page

INT #	INT Source	INT #	INT Source	INT #	INT Source
30	USB	66	Wiegand2		
31	Ethnet0	67	RTC MBOX		
32	Ethnet0	68	N/A		
33	EMMC Wakup	69	RTC IRRX		
34	EMMC	70	RTC GPIO		
35	SD0 Wakup	71	RTC UART		
36	SD0	72	RTC SPI_NOR		
37	SD1 Wakup	73	RTC I2C		
38	SD1	74	RTC WDG		
39	SPI_NAND	75	TPU		
40	I2S0	76	TDMA		
41	I2S1	77	Reserved		
42	I2S2	78	Reserved		
43	I2S3	79	Timer0		
44	UART0	80	Timer1		
45	UART1	81	Timer2		
46	UART2	82	Timer3		
47	UART3	83	Timer4		
48	UART4	84	Timer5		
49	I2C0	85	Timer6		
50	I2C1	86	Timer7		
51	I2C2	87	peri_firewall		

Table 3.3: Interrupt number and Interrupt source mapping for Slave RISCV C906 @ 700Mhz

INT #	INT Source	INT #	INT Source	INT #	INT Source
16	JPEG	32	I2C0	48	RTC GPIO
17	H264	33	I2C1	49	RTC UART
18	H265	34	I2C2	50	RTC I2C
19	VC SBM	35	I2C3	51	RTC WDG
20	ISP	36	I2C4	52	TDMA
21	SC_TOP	37	SPI1	53	Reserved
22	CSI_MAC0	38	SPI2	54	Reserved
23	CSI_MAC1	39	WatchDog2	55	Timer4
24	LDC	40	KEYSCAN	56	Timer5
25	System DMA	41	GPIO0	57	Timer6
26	I2S0	42	GPIO1	58	Timer7
27	I2S1	43	GPIO2	59	SPACC
28	I2S2	44	GPIO3	60	IVE
29	I2S3	45	Wiegand0	61	mbox
30	UART0	46	RTC MBOX		
31	UART1	47	RTC IRRX		

3.2 Address-Space Mapping

Table 3.4: Memory mapping

Start Address [31:0]	End Address [31:0]	Function Description	Size (Byte)
0x01000000	0x017FFFFF	Reserved	8M
0x01800000	0x018FFFFF	Reserved	
0x01900000	0x01900FFF	ap_mailbox	4K
0x01901000	0x01901FFF	ap_system_ctrl	4K
0x01902000	0x019EFFFF	Reserved	
0x01F00000	0x01F0FFFF	Reserved	64K
0x01F10000	0x01FFFFFF	Reserved	
0x02000000	0x02FFFFFF	Reserved	64K
0x03000000	0x03000FFF	TOP_MISC	4K
0x03001000	0x03001FFF	PINMUX	4K
0x03002000	0x03002FFF	CLKGEN/PLL	4K
0x03003000	0x03003FFF	RSTGEN	4K
0x03004000	0x03005FFF	Reserved	
0x03006000	0x03006FFF	Reserved	4K
0x03007000	0x03008FFF	Reserved	
0x03009000	0x03009FFF	Reserved	4K
0x0300A000	0x0300AFFF	Reserved	4K
0x0300B000	0x0300FFFF	Reserved	
0x03010000	0x03010FFF	WATCH DOG0	4K
0x03011000	0x03011FFF	WATCH DOG1	4K
0x03012000	0x03012FFF	WATCH DOG2	4K
0x03020000	0x03020FFF	GPIO0	4K
0x03021000	0x03021FFF	GPIO1	4K
0x03022000	0x03022FFF	GPIO2	4K
0x03023000	0x03023FFF	GPIO3	4K
0x03024000	0x0302FFFF	Reserved	
0x03030000	0x03030FFF	WGN0	4K
0x03031000	0x03031FFF	WGN1	4K
0x03032000	0x03032FFF	WGN2	4K
0x03033000	0x0303FFFF	Reserved	
0x03040000	0x0304FFFF	KEYSCAN	64K
0x03050000	0x0305FFFF	EFUSE	64K
0x03060000	0x03060FFF	PWM0	4K
0x03061000	0x03061FFF	PWM1	4K
0x03062000	0x03062FFF	PWM2	4K
0x03063000	0x03063FFF	PWM3	4K
0x03064000	0x0309FFFF	Reserved	
0x030A0000	0x030AFFFF	TIMER	64K
0x030C0000	0x030CFFFF	Reserved	
0x030D0000	0x030D0FFF	Reserved	4K
0x030D1000	0x030D1FFF	Reserved	4K
0x030D2000	0x030D2FFF	Reserved	4K
0x030D3000	0x030DFFFF	Reserved	
0x030E0000	0x030EFFFF	TEMPSEN	64K
0x030F0000	0x030FFFFF	SARADC	64K
0x04000000	0x0400FFFF	I2C0	64K

continues on next page

Table 3.4 – continued from previous page

Start Address [31:0]	End Address [31:0]	Function Description	Size (Byte)
0x04010000	0x0401FFFF	I2C1	64K
0x04020000	0x0402FFFF	I2C2	64K
0x04030000	0x0403FFFF	I2C3	64K
0x04040000	0x0404FFFF	I2C4	64K
0x04050000	0x0405FFFF	Reserved	
0x04060000	0x0406FFFF	SPI_NAND	64K
0x04070000	0x0407FFFF	ETH0	
0x04080000	0x040FFFFF	Reserved	
0x04100000	0x04107FFF	I2S0	64K
0x04108000	0x0410FFFF	I2S Global	64K
0x04110000	0x0411FFFF	I2S1	64K
0x04120000	0x0412FFFF	I2S2	64K
0x04130000	0x0413FFFF	I2S3	64K
0x04140000	0x0414FFFF	UART0	64K
0x04150000	0x0415FFFF	UART1	64K
0x04160000	0x0416FFFF	UART2	64K
0x04170000	0x0417FFFF	UART3	64K
0x04180000	0x0418FFFF	SPI0	64K
0x04190000	0x0419FFFF	SPI1	64K
0x041A0000	0x041AFFFF	SPI2	64K
0x041B0000	0x041BFFFF	SPI3	64K
0x041C0000	0x041CFFFF	UART4	64K
0x041D0000	0x041DFFFF	AUDSRC	64K
0x041E0000	0x042FFFFF	Reserved	
0x04300000	0x0430FFFF	eMMC	64K
0x04310000	0x0431FFFF	SD0	64K
0x04320000	0x0432FFFF	SD1	
0x04330000	0x0433FFFF	DMA	64K
0x04340000	0x0434FFFF	USB	64K
0x04350000	0x043FFFFF	Reserved	
0x04400000	0x0441FFFF	ROM	128K
0x04420000	0x04FFFFFF	Reserved	
0x05000000	0x05000FFF	Reserved	4KB
0x05020000	0x05020FFF	RTCSYS_Timer	4KB
0x05021000	0x05021FFF	RTCSYS_GPIO	4KB
0x05022000	0x05022FFF	RTCSYS_UART	4KB
0x05023000	0x05023FFF	RTCSYS_INTR	4KB
0x05024000	0x05024FFF	RTCSYS_MBOX	4KB
0x05025000	0x05025FFF	RTCSYS_CTRL	4KB
0x05026000	0x05026FFF	RTCSYS_CORE	4KB
0x05027000	0x05027FFF	RTCSYS_IO	4KB
0x05028000	0x05028FFF	RTCSYS_OSC	4KB
0x05029000	0x05029FFF	Reserved	4KB
0x0502A000	0x0502AFFF	RTCSYS_32kless	4KB
0x0502B000	0x0502BFFF	RTCSYS_I2C	4KB
0x0502C000	0x0502CFFF	RTCSYS_SARADC	4KB
0x0502D000	0x0502DFFF	RTCSYS_WDT	4KB
0x0502E000	0x0502EFFF	RTCSYS_IRRX	4KB
0x05200000	0x053FFFFF	RTCSYS_SRAM	8KB

continues on next page

Table 3.4 – continued from previous page

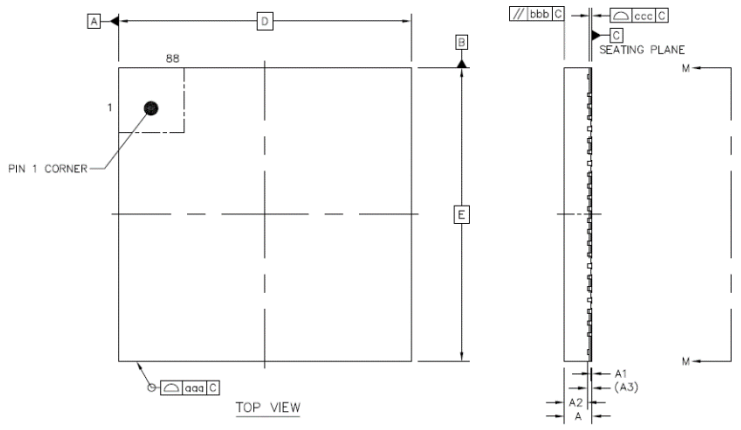
Start Address [31:0]	End Address [31:0]	Function Description	Size (Byte)
0x05400000	0x057FFFFF	RTCSYS_SPINOR	4MB
0x08000000	0x08001FFF	Reserved	8K
0x08004000	0x08005FFF	DDR Controler	8K
0x08006000	0x08007FFF	Reserved	8K
0x08008000	0x08009FFF	DDR AXI Monitor	8K
0x0800A000	0x0800BFFF	DDR Global	8K
0x08010000	0x08011FFF	Reserved	8K
0x08012000	0x08013FFF	Reserved	8K
0x08014000	0x09FFFFFF	Reserved	
0x0A000000	0x0A07FFFF	ISP	512K
0x0A080000	0x0A0803FF	sc_top	1K
0x0A080400	0x0A080BFF	Reserved	2K
0x0A080C00	0x0A080CFF	osd enc	256B
0x0A080D00	0x0A080FFF	Reserved	768B
0x0A081000	0x0A081FFF	Reserved	4K
0x0A082000	0x0A082FFF	img_v	4K
0x0A083000	0x0A083FFF	img_d	4K
0x0A084000	0x0A084FFF	sc_d	4K
0x0A085000	0x0A085FFF	sc_v1	4K
0x0A086000	0x0A086FFF	sc_v2	4K
0x0A087000	0x0A087FFF	sc_v3	4K
0x0A088000	0x0A088FFF	DISP	4K
0x0A089000	0x0A089FFF	Reserved	4K
0x0A08A000	0x0A08AFFF	dsi_mac	4K
0x0A08B000	0x0A08BFFF	cmdq	4K
0x0A08C000	0x0A08CFFF	Reserved	4K
0x0A08D000	0x0A08DFFF	Reserved	4K
0x0A08E000	0x0A09FFFF	Reserved	72K
0x0A0A0000	0x0A0AFFFF	IVE	64K
0x0A0A0000	0x0A0BFFFF	Reserved	64K
0x0A0C0000	0x0A0C1FFF	ldc	8K
0x0A0C2000	0x0A0C3FFF	VI0/MIPI_RX0	8K
0x0A0C4000	0x0A0C5FFF	VI1/MIPI_RX1	8K
0x0A0C6000	0x0A0C7FFF	VI2/MIPI_RX2	8K
0x0A0C8000	0x0A0C9FFF	VIPSYS	8K
0x0A0CA000	0x0A0CFFFF	Reserved	24K
0x0A0D0000	0x0A0D0FFF	CSI_PHY	4K
0x0A0D1000	0x0A0D1FFF	DSI_PHY	4K
0x0A0D2000	0x0AFFFFFF	Reserved	
0x0B000000	0x0B00FFFF	JPEG codec	64K
0x0B010000	0x0B01FFFF	H.264 codec	64K
0x0B020000	0x0B02FFFF	H.265 codec	64K
0x0B030000	0x0BFFFFFF	Reserved	
0x0C000000	0x0FFFFFFF	Reserved	
0x10000000	0x1FFFFFFF	SPI_NOR	256M
0x30000000	0x7FFFFFFF	Reserved	
0x80000000	0xFFFFFFFF	DDR	2G

* Reading and writing operations on the reserved address space may produce unpredictable results.

3.3 Packaging and Pin

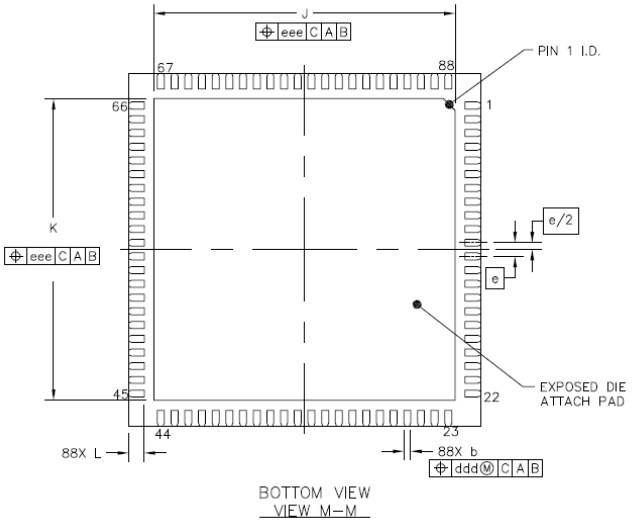
3.3.1 Packaging

Using QFN package, the package size is 9mm x 9mm x 0.9mm. Pin spacing is 0.35mm. The total number of pins is 88. Please refer to the figure below for detailed package dimensions.



		SYMBOL	MIN	NOM	MAX
TOTAL THICKNESS		A	0.8	0.85	0.9
STAND OFF		A1	0	0.035	0.05
MOLD THICKNESS		A2	---	0.7	---
L/F THICKNESS		A3		0.152 REF	
LEAD WIDTH		b	0.13	0.18	0.23
BODY SIZE	X	D		9 BSC	
	Y	E		9 BSC	
LEAD PITCH		e		0.35 BSC	
EP SIZE	X	J	7.6	7.7	7.8
	Y	K	7.6	7.7	7.8
LEAD LENGTH		L	0.35	0.4	0.45
PACKAGE EDGE TOLERANCE		aaa		0.1	
MOLD FLATNESS		bbb		0.1	
COPLANARITY		ccc		0.08	
LEAD OFFSET		ddd		0.1	
EXPOSED PAD OFFSET		eee		0.1	

Diagram 3.1: Package appearance dimensions, top view



		SYMBOL	MIN	NOM	MAX
TOTAL THICKNESS		A	0.8	0.85	0.9
STAND OFF		A1	0	0.035	0.05
MOLD THICKNESS		A2	---	0.7	---
L/F THICKNESS		A3		0.152 REF	
LEAD WIDTH		b	0.13	0.18	0.23
BODY SIZE	X	D		9 BSC	
	Y	E		9 BSC	
LEAD PITCH		e		0.35 BSC	
EP SIZE	X	J	7.6	7.7	7.8
	Y	K	7.6	7.7	7.8
LEAD LENGTH		L	0.35	0.4	0.45
PACKAGE EDGE TOLERANCE		aaa		0.1	
MOLD FLATNESS		bbb		0.1	
COPLANARITY		ccc		0.08	
LEAD OFFSET		ddd		0.1	
EXPOSED PAD OFFSET		eee		0.1	

Diagram 3.2: Package appearance dimensions, bottom view

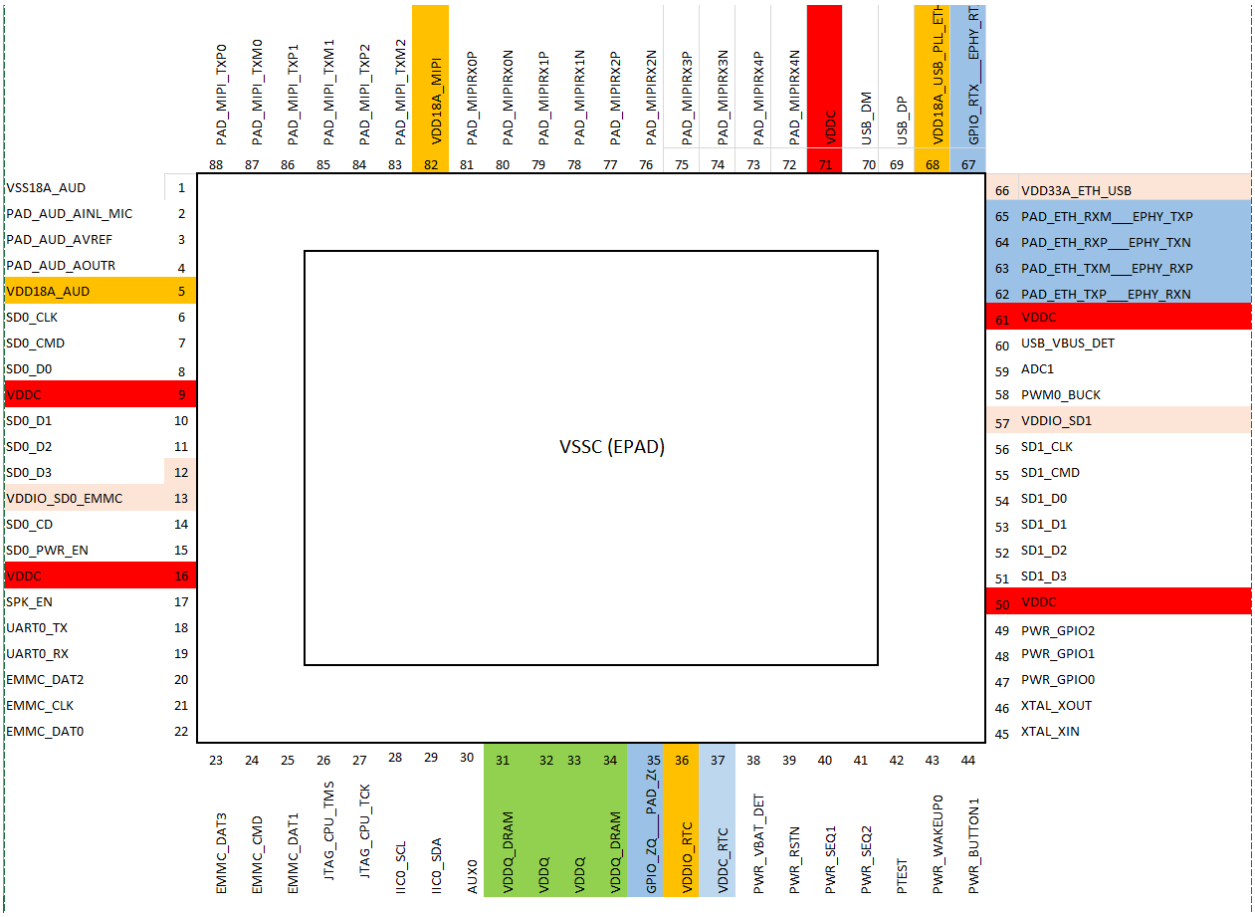


Diagram 3.3: Pin layout diagram

3.3.2 Pin information description

Table 3.5: Pin information table

Pin-Num	Pin Name	IO Type	IOGroup	PowerDomain	Note
6	SD0_CLK	18OD33 IO	G10	VDDIO_SD0_EMMC	
7	SD0_CMD	18OD33 IO	G10	VDDIO_SD0_EMMC	
8	SD0_D0	18OD33 IO	G10	VDDIO_SD0_EMMC	
10	SD0_D1	18OD33 IO	G10	VDDIO_SD0_EMMC	
11	SD0_D2	18OD33 IO	G10	VDDIO_SD0_EMMC	
12	SD0_D3	18OD33 IO	G10	VDDIO_SD0_EMMC	
14	SD0_CD	18OD33 IO	G7	VDDIO_SD0_EMMC	

continues on next page

Table 3.5 – continued from previous page

Pin-Num	Pin Name	IO Type	IOGroup	PowerDomain	Note
15	SD0_PWR_EN	18OD33 IO	G7	VDDIO_SD0_EMMC	
17	SPK_EN	18OD33 IO	G7	VDDIO_SD0_EMMC	
18	UART0_TX	18OD33 IO	G7	VDDIO_SD0_EMMC	
19	UART0_RX	18OD33 IO	G7	VDDIO_SD0_EMMC	
20	EMMC_DAT2	18OD33 IO	G7	VDDIO_SD0_EMMC	
21	EMMC_CLK	18OD33 IO	G7	VDDIO_SD0_EMMC	
22	EMMC_DAT0	18OD33 IO	G7	VDDIO_SD0_EMMC	
23	EMMC_DAT3	18OD33 IO	G7	VDDIO_SD0_EMMC	
24	EMMC_CMD	18OD33 IO	G7	VDDIO_SD0_EMMC	
25	EMMC_DAT1	18OD33 IO	G7	VDDIO_SD0_EMMC	
26	JTAG_CPU_TMS	18OD33 IO	G7	VDDIO_SD0_EMMC	
27	JTAG_CPU_TCK	18OD33 IO	G7	VDDIO_SD0_EMMC	
28	IIC0_SCL	18OD33 IO	G7	VDDIO_SD0_EMMC	
29	IIC0_SDA	18OD33 IO	G7	VDDIO_SD0_EMMC	
30	AUX0	18OD33 IO	G7	VDDIO_SD0_EMMC	
38	PWR_VBAT_DET	1.8V GPIO	GRTC	VDDIO_RTC	
39	PWR_RSTN	1.8V GPIO	GRTC	VDDIO_RTC	
40	PWR_SEQ1	1.8V GPIO	GRTC	VDDIO_RTC	
41	PWR_SEQ2	1.8V GPIO	GRTC	VDDIO_RTC	
43	PWR_WAKEUP0	1.8V GPIO	GRTC	VDDIO_RTC	
44	PWR_BUTTON1	1.8V GPIO	GRTC	VDDIO_RTC	
45	XTAL_XIN	Xtal	GRTC	VDDIO_RTC	
47	PWR_GPIO0	1.8V GPIO	GRTC	VDDIO_RTC	
48	PWR_GPIO1	1.8V GPIO	GRTC	VDDIO_RTC	

continues on next page

Table 3.5 – continued from previous page

Pin-Num	Pin Name	IO Type	IOGroup	PowerDomain	Note
49	PWR_GPIO2	1.8V GPIO	GRTC	VDDIO_RTC	
51	SD1_D3	18OD33 IO	GRTC	VDDIO_SD1	SD1_D3 or VO[32] Check 0x0502_70E4
52	SD1_D2	18OD33 IO	GRTC	VDDIO_SD1	SD1_D2 or VO[33] Check 0x0502_70E4
53	SD1_D1	18OD33 IO	GRTC	VDDIO_SD1	SD1_D1 or VO[34] Check 0x0502_70E4
54	SD1_D0	18OD33 IO	GRTC	VDDIO_SD1	SD1_D0 or VO[35] Check 0x0502_70E4
55	SD1_CMD	18OD33 IO	GRTC	VDDIO_SD1	SD1_CMD or VO[36] Check 0x0502_70E4
56	SD1_CLK	18OD33 IO	GRTC	VDDIO_SD1	SD1_CLK or VO[37] Check 0x0502_70E4
58	PWM0_BUCK	1.8V GPIO	G1	VDD18A_USB _PLL_ETH	
59	ADC1	1.8V GPIO	G1	VDD18A_USB _PLL_ETH	
60	USB_VBUS_DET	1.8V GPIO	G1	VDD18A_USB _PLL_ETH	
62	PAD_ETH_TXP ___EPHY_RXN	ETH GPIO (1.8V)		VDD18A_USB _PLL_ETH	
63	PAD_ETH_TXM ___EPHY_RXP	ETH GPIO (1.8V)		VDD18A_USB _PLL_ETH	
64	PAD_ETH_RXP ___EPHY_TXN	ETH GPIO (1.8V)		VDD18A_USB _PLL_ETH	
65	PAD_ETH_RXM ___EPHY_TXP	ETH GPIO (1.8V)		VDD18A_USB _PLL_ETH	
72	PAD_MIPIRX4N	1.8V GPIO	G12	VDD18A_MIPI	
73	PAD_MIPIRX4P	1.8V GPIO	G12	VDD18A_MIPI	
74	PAD_MIPIRX3N	1.8V GPIO	G12	VDD18A_MIPI	
75	PAD_MIPIRX3P	1.8V GPIO	G12	VDD18A_MIPI	
76	PAD_MIPIRX2N	1.8V GPIO	G12	VDD18A_MIPI	
77	PAD_MIPIRX2P	1.8V GPIO	G12	VDD18A_MIPI	
78	PAD_MIPIRX1N	1.8V GPIO	G12	VDD18A_MIPI	

continues on next page

Table 3.5 – continued from previous page

Pin-Num	Pin Name	IO Type	IOGroup	PowerDomain	Note
79	PAD_MIPIRX1P	1.8V GPIO	G12	VDD18A_MIPI	
80	PAD_MIPIRX0N	1.8V GPIO	G12	VDD18A_MIPI	
81	PAD_MIPIRX0P	1.8V GPIO	G12	VDD18A_MIPI	
83	PAD_MIPI_TXM2	1.8V GPIO	G12	VDD18A_MIPI	
84	PAD_MIPI_TXP2	1.8V GPIO	G12	VDD18A_MIPI	
85	PAD_MIPI_TXM1	1.8V GPIO	G12	VDD18A_MIPI	
86	PAD_MIPI_TXP1	1.8V GPIO	G12	VDD18A_MIPI	
87	PAD_MIPI_TXM0	1.8V GPIO	G12	VDD18A_MIPI	
88	PAD_MIPI_TXP0	1.8V GPIO	G12	VDD18A_MIPI	
2	PAD_AUD_AINL _MIC	AUDIO GPIO (1.8V)		VDD18A_MIPI	
4	PAD_AUD_AOUTR	AUDIO GPIO (1.8V)		VDD18A_MIPI	
67	GPIO_RTX ___EPHY_RTX	1.8V GPIO	G12	VDD18A_USB _PLL_ETH	
35	GPIO_ZQ ___PAD_ZQ	1.8V GPIO	GRTC	VDDIO_RTC	
#N/A	PKG_TYPE0	1.8V GPIO	G1	VDD18A_USB _PLL_ETH	no pin out
#N/A	PKG_TYPE1	1.8V GPIO	G1	VDD18A_USB _PLL_ETH	no pin out
#N/A	PKG_TYPE2	1.8V GPIO	G1	VDD18A_USB _PLL_ETH	no pin out
#N/A	MUX_SPI1_MISO	NA	NA	NA	Result is feed to PAD_MIPIRX3N func7
#N/A	MUX_SPI1_MOSI	NA	NA	NA	Result is feed to PAD_MIPIRX3P func7
#N/A	MUX_SPI1_CS	NA	NA	NA	Result is feed to PAD_MIPIRX4P func7
#N/A	MUX_SPI1_SCK	NA	NA	NA	Result is feed to PAD_MIPIRX4N func7

3.3.3 Pin default state

PinNum	PinName	Default Function	Default Function Select	Default Pull (PU/PD/NO PULL)	Default Function Direction (During reset)	IO TRAP (become input during reset)	IO Type	PowerDomain	IOBLK Group	Fail-safe
6	SD0_CLK	SDIO0_CLK	0	PD (61Kohm)->DRV_LO + PD	Output (Low)		18OD33	VDDIO_SD0_EMMC	G10	
7	SD0_CMD	SDIO0_CMD	0	PD (61Kohm)	Input		18OD33	VDDIO_SD0_EMMC	G10	
8	SD0_D0	SDIO0_D[0]	0	PD (61Kohm)	Input		18OD33	VDDIO_SD0_EMMC	G10	
10	SD0_D1	SDIO0_D[1]	0	PD (61Kohm)	Input		18OD33	VDDIO_SD0_EMMC	G10	
11	SD0_D2	SDIO0_D[2]	0	PD (61Kohm)	Input		18OD33	VDDIO_SD0_EMMC	G10	
12	SD0_D3	SDIO0_D[3]	0	PD (61Kohm)	Input		18OD33	VDDIO_SD0_EMMC	G10	
14	SD0_CD	SDIO0_CD	0	PU (60Kohm)	Input		18OD33	VDDIO_SD0_EMMC	G7	
15	SD0_PWR_EN	XGPIOA[14]	3	PD (61Kohm) -> DRV_HI/LO + PD	Input (Trap) -> Output	V	18OD33	VDDIO_SD0_EMMC	G7	
17	SPK_EN	XGPIOA[15]	3	PD (61Kohm)	Input		18OD33	VDDIO_SD0_EMMC	G7	
18	UART0_TX	UART0_TX	0	PU (60Kohm) -> DRV_HI + PU	Input (Trap) -> Output	V	18OD33	VDDIO_SD0_EMMC	G7	
19	UART0_RX	UART0_RX	0	PU (60Kohm)	Input		18OD33	VDDIO_SD0_EMMC	G7	
20	EMMC_DAT2	SPINOR_HOLD_X	1	PD (61Kohm) -> DRV_HI + PD	Input		18OD33	VDDIO_SD0_EMMC	G7	
21	EMMC_CLK	SPINOR_SCK	1	DRV_LO + PD (61Kohm)	Output (Low)		18OD33	VDDIO_SD0_EMMC	G7	
22	EMMC_DAT0	SPINOR_MOSI	1	PU (60Kohm)	Input (Trap)	V	18OD33	VDDIO_SD0_EMMC	G7	
23	EMMC_DAT3	SPINOR_WP_X	1	PU (60Kohm) -> DRV_HI + PU	Input (Trap)	V	18OD33	VDDIO_SD0_EMMC	G7	
24	EMMC_CMD	SPINOR_MISO	1	PU (60Kohm)	Input		18OD33	VDDIO_SD0_EMMC	G7	
25	EMMC_DAT1	SPINOR_CS_X	1	DRV_HI + PD (61Kohm)	Output (High)		18OD33	VDDIO_SD0_EMMC	G7	
26	JTAG_CPU_TMS	CR_4WTMS	0	PD (61Kohm)	Input		18OD33	VDDIO_SD0_EMMC	G7	
27	JTAG_CPU_TCK	CR_4WTCK	0	PD (61Kohm)	Input		18OD33	VDDIO_SD0_EMMC	G7	
28	IIC0_SCL	CR_4WTDI	0	PU (60Kohm)	Input		18OD33	VDDIO_SD0_EMMC	G7	
29	IIC0_SDA	CR_4WTDO	0	PU (60Kohm)	Output (CR)		18OD33	VDDIO_SD0_EMMC	G7	
30	AUX0	XGPIOA[30]	3	PD (61Kohm)	Input		18OD33	VDDIO_SD0_EMMC	G7	
38	PWR_VBAT_DET	PWR_VBAT_DET	0	NO PULL	Input		1.8V_IO	VDDIO_RTC	GRTC	
39	PWR_RSTN	PWR_RSTN	0	PU (60Kohm)	Input		1.8V_IO	VDDIO_RTC	GRTC	Fail-safe
40	PWR_SEQ1	PWR_SEQ1	0	PD (87Kohm) -> DRV + PD	Output (L->H)		1.8V_IO	VDDIO_RTC	GRTC	Fail-safe
41	PWR_SEQ2	PWR_SEQ2	0	PD (87Kohm) -> DRV + PD	Output (L->H)		1.8V_IO	VDDIO_RTC	GRTC	
43	PWR_WAKEUP0	PWR_WAKEUP0	0	PD (87Kohm)	Input		1.8V_IO	VDDIO_RTC	GRTC	Fail-safe
44	PWR_BUTTON1	PWR_BUTTON1	0	PU (60Kohm)	Input		1.8V_IO	VDDIO_RTC	GRTC	Fail-safe
47	PWR_GPIO0	PWR_GPIO[0]	0	PD (87Kohm)	Input		1.8V_IO	VDDIO_RTC	GRTC	Fail-safe
48	PWR_GPIO1	PWR_GPIO[1]	0	PD (87Kohm)	Input		1.8V_IO	VDDIO_RTC	GRTC	
49	PWR_GPIO2	PWR_GPIO[2]	0	PD (87Kohm)	Input		1.8V_IO	VDDIO_RTC	GRTC	
51	SD1_D3	PWR_SPINOR1_CS_X	6	DRV_HI + PD (61Kohm)	Output (High)		18OD33	VDDIO_SD1	GRTC	
52	SD1_D2	PWR_SPINOR1_HOLD_X	6	PD (61Kohm) -> DRV_HI + PD	Input		18OD33	VDDIO_SD1	GRTC	
53	SD1_D1	PWR_SPINOR1_WP_X	6	PD (61Kohm) -> DRV_HI + PD	Input		18OD33	VDDIO_SD1	GRTC	
54	SD1_D0	PWR_SPINOR1_MISO	6	PD (61Kohm)	Input		18OD33	VDDIO_SD1	GRTC	
55	SD1_CMD	PWR_SPINOR1_MOSI	6	PD (61Kohm)	Input		18OD33	VDDIO_SD1	GRTC	
56	SD1_CLK	PWR_SPINOR1_SCK	6	DRV_LO + PD (61Kohm)	Output (Low)		18OD33	VDDIO_SD1	GRTC	
58	PWM0_BUCK	XGPIOB[0]	3	NO PULL	Input		1.8V_IO	VDD18A_USB_PLL_ETH	G1	Fail-safe
59	ADC1	XGPIOB[3]	3	PD (87Kohm)	Input		1.8V_IO	VDD18A_USB_PLL_ETH	G1	Fail-safe
60	USB_VBUS_DET	USB_VBUS_DET	0	NO PULL	Input		1.8V_IO	VDD18A_USB_PLL_ETH	G1	
62	PAD_ETH_TXP___EPHY_RXN	XGPIOB[25]	3	NO PULL			ETH_GPIO	VDD18A_USB_PLL_ETH	G12	
63	PAD_ETH_TXM___EPHY_RXP	XGPIOB[24]	3	NO PULL			ETH_GPIO	VDD18A_USB_PLL_ETH	G12	
64	PAD_ETH_RXP___EPHY_TXN	XGPIOB[27]	3	NO PULL			ETH_GPIO	VDD18A_USB_PLL_ETH	G12	
65	PAD_ETH_RXM___EPHY_TXP	XGPIOB[26]	3	NO PULL			ETH_GPIO	VDD18A_USB_PLL_ETH	G12	
72	PAD_MIPIRX4N	CR_SCL0	0	PD (87Kohm)	Input		1.8V_IO	VDD18A_MIPI	G12	
73	PAD_MIPIRX4P	CR_SDA0	0	PD (87Kohm)	Input		1.8V_IO	VDD18A_MIPI	G12	
74	PAD_MIPIRX3N	XGPIOC[4]	3	PD (87Kohm)	Input		1.8V_IO	VDD18A_MIPI	G12	
75	PAD_MIPIRX3P	XGPIOC[5]	3	PD (87Kohm)	Input		1.8V_IO	VDD18A_MIPI	G12	
76	PAD_MIPIRX2N	XGPIOC[6]	3	PD (87Kohm)	Input		1.8V_IO	VDD18A_MIPI	G12	
77	PAD_MIPIRX2P	XGPIOC[7]	3	PD (87Kohm)	Input		1.8V_IO	VDD18A_MIPI	G12	
78	PAD_MIPIRX1N	XGPIOC[8]	3	PD (87Kohm)	Input		1.8V_IO	VDD18A_MIPI	G12	
79	PAD_MIPIRX1P	XGPIOC[9]	3	PD (87Kohm)	Input		1.8V_IO	VDD18A_MIPI	G12	
80	PAD_MIPIRX0N	XGPIOC[10]	3	PD (87Kohm)	Input		1.8V_IO	VDD18A_MIPI	G12	
81	PAD_MIPIRX0P	XGPIOC[11]	3	PD (87Kohm)	Input		1.8V_IO	VDD18A_MIPI	G12	
83	PAD_MIPI_TXM2	XGPIOC[16]	3	PD (87Kohm)	Input		1.8V_IO	VDD18A_MIPI	G12	
84	PAD_MIPI_TXP2	XGPIOC[17]	3	PD (87Kohm)	Input		1.8V_IO	VDD18A_MIPI	G12	
85	PAD_MIPI_TXM1	XGPIOC[14]	3	PD (87Kohm)	Input		1.8V_IO	VDD18A_MIPI	G12	
86	PAD_MIPI_TXP1	XGPIOC[15]	3	PD (87Kohm)	Input		1.8V_IO	VDD18A_MIPI	G12	
87	PAD_MIPI_TXM0	XGPIOC[12]	3	PD (87Kohm)	Input		1.8V_IO	VDD18A_MIPI	G12	
88	PAD_MIPI_TXP0	XGPIOC[13]	3	PD (87Kohm)	Input		1.8V_IO	VDD18A_MIPI	G12	
2	PAD_AUD_AINL_MIC	XGPIOC[23]	3	NO PULL	Input		AUDIO_GPIO	VDD18A_MIPI	G11	
4	PAD_AUD_AOUTR	XGPIOC[24]	3	NO PULL	Output		AUDIO_GPIO	VDD18A_MIPI	G11	
67	GPIO_RTX___EPHY_RTX	XGPIOB[23]	3	PD (87Kohm)	Input	V	1.8V_IO	VDD18A_USB_PLL_ETH	G12	Fail-safe
35	GPIO_ZQ___PAD_ZQ	PWR_GPIO[24]	3	PD (87Kohm)	Input	V	1.8V_IO	VDDIO_RTC	GRTC	Fail-safe

Diagram 3.4: Pin default state table (QFN)

CHAPTER

FOUR

BOOT AND UPGRADE

4.1 Overview

The chip is booted from the built-in ROM (BOOTROM). When the chip is reset, it will detect whether there is a weak pull-up or weak pull-down on the two pins (EMMC_DAT3, EMMC_DAT0) to confirm the type of memory device currently selected.

Secure boot chips will check during boot and chip upgrade to ensure that the software being executed or upgraded is safe.

4.2 Correspondence between startup mode and corresponding signal latch value

- Support booting from SPI Nor Flash (EMMC_DAT3 pull down, EMMC_DAT0 pull up)
- Support booting from SPI Nand Flash (EMMC_DAT3 pull down, EMMC_DAT0 pull down)
- Support booting from eMMC (EMMC_DAT3 pull up, EMMC_DAT0 pull up)

Note: SD0 and eMMC domain share IO power, because the SD card run with 3.0V by default, and eMMC is mostly 1.8V, basically eMMC is not supported unless SD0 does not connect with the SD card.

4.3 Image burning mode

- Supports image burning via SD card.
- Supports image burning through USB device.
- If there is an image in flash, support software upgrade through the network.

4.4 Secure boot

- Support secure boot and upgrade.
- AES/DES/SM4 hardware encryption and decryption.
- SHA/TRNG/Secure Efuse security hardware.

CHAPTER

FIVE

POWER MANAGEMENT

5.1 Overview

5.1.1 Power Domain

The chip supports two main power domains.

- Active Domain
- No-die Domain

5.1.1.1 Active Domain

That is the ordinary SoC power supply method.

5.1.1.2 No-die Domain

VDDIO_RTC power supply (VDDIO_SD1 as appropriate) is required for this Domain.

Inside the SoC, this power domain is implemented by a subsystem called RTCSYS.

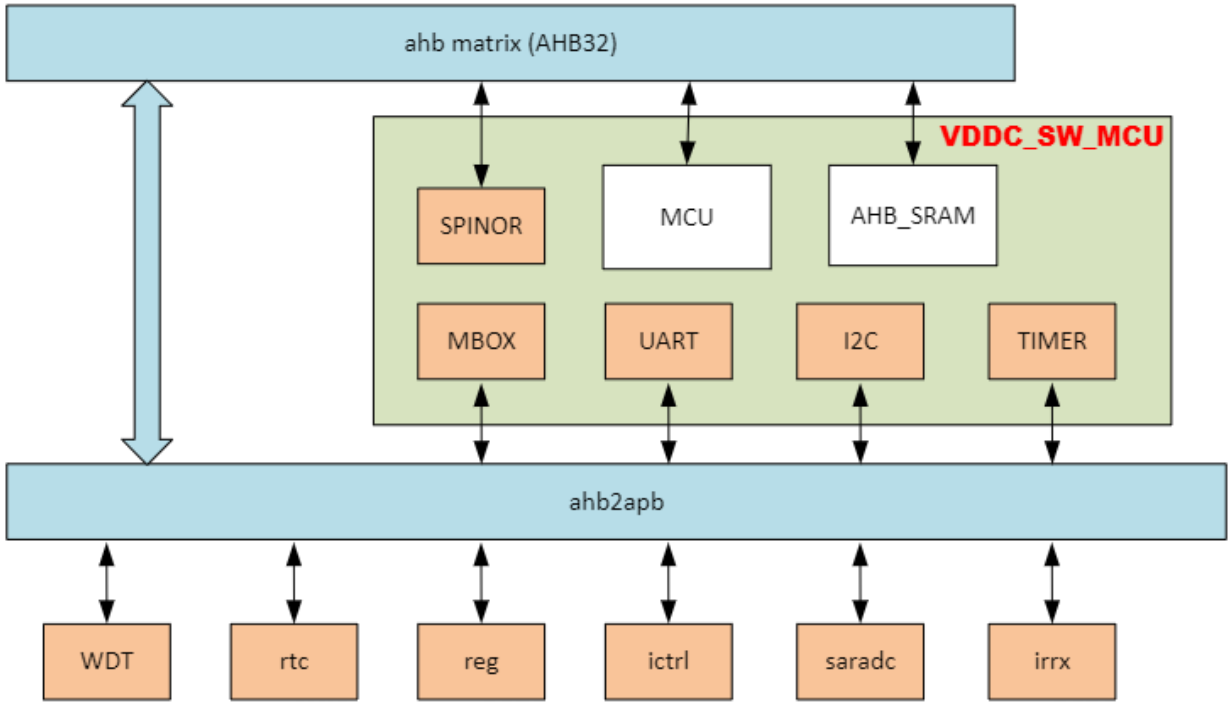


Diagram 5.1: RTCSYS subsystem architecture

The RTCSYS subsystem is internally divided into two sub power domains: AO (Always-On) domain and MCU domain (VDDC_SW_MCU area). The system can choose to turn off the power domain to which the MCU belongs through registers to achieve power saving needs. Please refer to the *Power Domain Control Process* chapter for details.

In the system sleep state, the MCU can handle interrupts and wake up the system through the configuration register, and can also communicate with external devices through I2C/UART.

The specific composition of the RTCSYS subsystem is as follows:

- AO (Always-On) domain
  - 1 set of WDT is used to issue an interrupt or reset signal after a certain period of time when an abnormality occurs in the system to interrupt or reset the entire system. Related control registers refer to the RTCSYS_WDT section of the *WDT Register Overview* chapter.
  - RTC (Real Time Clock), for detailed introduction, please refer to the *Real Time Clock* chapter.
  - 1 set of interrupt controllers to manage interrupt sources (ictrl). For detailed introduction, please refer to the *Interrupt Handling* chapter.
  - 1 set of GPIO, related control registers refer to the RTCSYS_GPIO section of the *GPIO Register Overview* chapter.
  - 1 set of SARADC, related control registers refer to the RTCSYS_SARADC section of the *SARADC Register Overview* chapter.
  - 1 group of IRRX, related control registers refer to the *IRRX Register Overview* chapter.
- MCU domain
  - An 8051 microcontroller (MCU), see chapter *8051 Subsystem* for details.
  - 8KB space AHB SRAM, which can be used by 8051 as instruction TCM or temporary data storage.

  - 2 sets of 32-bit counters, used for timing and counting functions, which can be used for application programs to implement timing and counting, or for the operating system to implement system clocks.
  - 1 group off-chip SPINOR control.
  - 2 sets of Mailbox, allowing ACPU and 8051 to communicate with each other. For detailed introduction, please refer to the *MAILBOX* chapter.
  - 1 set of I2C, related control registers refer to the RTCSYS_I2C section of the *I2C Register Overview* chapter.
  - 1 set of UART, related control registers refer to the RTCSYS_UART section of the *UART Register Overview* chapter.

5.1.2 Power operating mode

Based on Power Domain, the chip supports two main power operating modes:

- MCU-Only (32k-less) Mode:

  Only No-die Domain works. In this No-die domain, there is a small MCU system with its own clock, timer, uart, i2c, gpio, adc and other peripherals, which can replace the SoC's external MCU. The MCU in the No-die Domain can wake up the main system and return to Active Mode after receiving and filtering external input.

  There is a calibrated oscillator in the system to wake up quickly and fall asleep quickly. To further save power consumption every time it is woken up.

  When the MCU is in the Idle state, the power consumption is approximately 200uA.

- Active Mode

  Active mode is a state where the chip is fully awake and working. At this time, as long as the power supply is normal, the devices in the No-die Domain and Active Domain will work normally. But there are other power-saving techniques such as dynamic frequency scaling or dynamic clock gating.

5.2 Clock Control

5.2.1 Turn off unnecessary clock dividers

Refer to the *Clock* configuration chapter and turn off the unused clock divider according to the clock source required by each module, to achieve the purpose of saving power consumption.

5.2.2 Adjust the operating frequency of the module

Select a lower clock source according to the clock specifications required by each module. There are many frequency division configurations to reduce the module operating frequency. It should be noted that reducing the frequency of a single module may not necessarily reduce the overall power consumption.

5.2.3 Module-level low-power control

- Analog module: MIPI/USB/ETH/AUD related register settings, turn off unused modules or enter low power consumption mode.
- Digital modules: Turn off the clocks of unnecessary digital modules according to hardware and specifications.

5.2.4 Shut down unused PLLs

Referring to the PLL configuration, you can Powerdown the PLL that is not needed to save power consumption.

5.3 DDR low power control

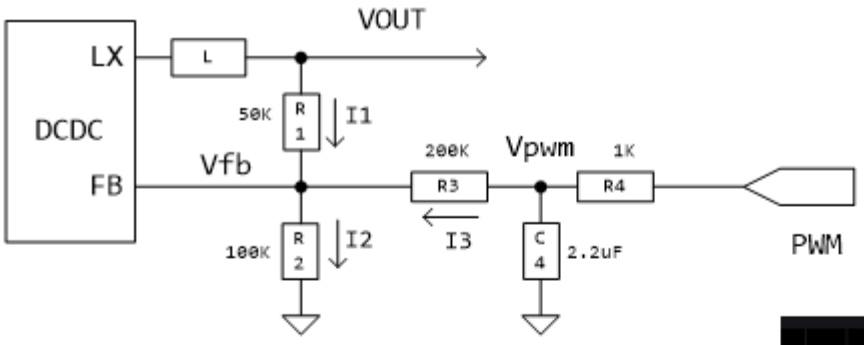
After the bus is not accessed for a period of time, the DDR controller will automatically enter the “Self refresh” and “Power down” states to reduce system power consumption.

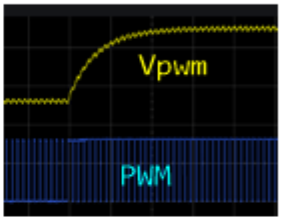
In some scenarios, it is impossible to find a large enough gap to enter “self refresh” due to intermittent access. At this time, you can also consider using the built-in statistical register to access the data volume. To confirm whether the bandwidth is excessive, you can consider directly reducing the frequency.

The DDR controller supports dynamic frequency adjustment. However, adjusting the frequency will temporarily stop DDR access for a period of time. Therefore, in order to reduce interruption time, which may make real-time application buffer underflow or overflow, it is limited to two levels: 50% and 100%.

5.4 Voltage Regulation

By default the chip uses PWM0 to control VDDC voltage regulation. The following is an example of using PWM to control the DCDC output voltage.





VOUT = Vfb + R1(I2-I3)
= Vfb + R1(Vfb/R2-(Vpwm-Vfb)/R3)
= Vfb + R1(Vfb/R2-(VDDIOxduty-Vfb)/R3)

Vfb	R1	R2	R3	VDDIO
0.6	50,000	100,000	200,000	1.8

Duty	Vpwm	Vout
0.00%	0.00	1.05
5.00%	0.09	1.03
10.00%	0.18	1.01
15.00%	0.27	0.98
20.00%	0.36	0.96
25.00%	0.45	0.94
30.00%	0.54	0.92
35.00%	0.63	0.89
40.00%	0.72	0.87
45.00%	0.81	0.85
50.00%	0.90	0.83
55.00%	0.99	0.80
60.00%	1.08	0.78
65.00%	1.17	0.76
70.00%	1.26	0.74
75.00%	1.35	0.71
80.00%	1.44	0.69
85.00%	1.53	0.67
90.00%	1.62	0.65
95.00%	1.71	0.62
100.00%	1.80	0.60

5 RC = 5x1Kx2.2uF = 11ms
(Vpwm settle time)

Vpwm = VDDIO x PWM duty cycle

Diagram 5.2: Example of controlling DCDC voltage using PWM

CHAPTER

SIX

REAL TIME CLOCK

6.1 Overview

The RTC (Real Time Clock) is an independently powered module in the chip. It contains a 32KHz oscillator and a Power-On-Reset (POR) sub-module, which can be used for time display and scheduled alarm produce. In addition, the hardware state machine provides triggering and timing control for chip power-on, power-off and reset.

6.2 Features

RTC has the following characteristics:

- Provide chip system reset signal
- Provides 32,768 Hz count clock (error < $\pm$1%)
- Provides 32-bit seconds counter and hardware calibration circuit, allowing second accuracy to reach 5ppm level
- Support alarm clock configuration and generate alarm interrupt
- Provide 2KB space SRAM, which can save software code or temporary data
- Support battery low voltage detection and generate interrupts
- Support button triggering the chip to wake up from sleep
- Support software to trigger chip sleep, reset, or restart due to overheating
- Support Watchdog to trigger chip system reset
- Support alarm triggering chip to wake up from sleep
- Configurable power-on and power-off sequence and reset time interval
- Provides 1 ultra-low power analog seconds counter (32 bit to count from 1970~2106 year)

6.3 Function description

The RTC is an independently powered constant current module in the chip. When the RTC is powered on for the first time, the internal POR module generates a low-level pulse, and the 32KHz oscillator starts to oscillate. The POR low level maintenance time is greater than 13 clock(32KHz) cycles, and the RTC enters the initial state. When the state machine detects that the battery voltage is in a normal state, it starts to complete the chip power-on process according to the default value sequence and releases the system reset signal. After the software is powered on for the first time, it is necessary to initialize the RTC and configure the initial counting value. When the system is to be shut down or enter sleep mode, the system control register (RTC_CTRL) can be configured to trigger the RTC state machine to complete the chip power-off process according to the configuration sequence.

When the chip is in power-off shutdown or sleep state, it will continue to work as long as the RTC power supply is maintained to save necessary software code or user data in SRAM and information registers (RTC_INFO0 ~ RTC_INFO3, RTC_NOPOR_INFO0 ~ RTC_NOPOR_INFO3); the counter will continue to count. At the same time, it will also detect whether the button triggers the chip to turn on or wake up from sleep. When the trigger is received, the state machine will complete the chip power-on process according to the configured timing and release the system reset signal. After the processor core is restarted, the software can determine the chip status by reading back the contents previously written to the information register. The RTC also provides two status registers (RTC_ST_ON_REASON, RTC_ST_OFF_REASON) to record the previous power-off or power-on of the chip respectively. As well as the trigger conditions for reset, and provide more detailed content, such as whether unexpected events have occurred: forced reset, chip overheating, or battery/power supply loss, etc. In addition, when the RTC receives the Watchdog, it will trigger the state machine to send out a system reset signal according to the configured timing to restart the chip.

The RTC's counting clock uses a 32KHz clock and runs based on a 32-bit adding counter to provide second counting. The initial counting value is loaded from the register RTC_SET_SEC_CNTR_VALUE. The software can read back the second value through the register RTC_SEC_CNTR_VALUE and convert it into the specific year, month, day, hour, and minute.

The 32KHz clock and second pulse period can be calibrated through the software process, or the hardware module can be turned on to perform automatic calibration periodically.

Software can enable Alarm by configuring the 32-bit register RTC_ALARM_TIME and writing 1 to RTC_ALARM_ENABLE. When the second count value RTC_SEC_CNTR_VALUE is incremented to be equal to the RTC_ALARM_TIME value, the RTC will generate an alarm interrupt. The interrupt status will remain until 0 is written to RTC_ALARM_ENABLE.

In addition, the RTC provides battery low-voltage detection. When the battery voltage is lower than a certain level, the RTC will generate an interrupt. After the software receives the low-voltage interrupt, it can immediately execute the shutdown program and trigger the RTC to complete the power-off process to prevent abnormal errors in the system.

6.4 Way of working

6.4.1 Count clock frequency

RTC second counter, the maximum counting time is:

2^32 = 49710 days = 136 years

6.4.2 RTC reset

As a chip power-on and power-off control unit, RTC itself cannot be soft-reset alone. In addition to the POR for the first power-on, it can support a forced full-chip reset (including RTC) by pressing the RSTN button in case of an abnormality. After the RSTN button is released, all RTC registers are restored to their default values, and the state machine returns to its initial state. If the state machine detects that the battery voltage is in a normal state, it starts to complete the chip power-on process according to the default value sequence and releases the system reset signal.

6.4.3 RTC initialization

After the chip is powered on for the first time, the system needs to initialize the RTC. However, before initialization, the 32KHz oscillator clock and second time need to be calibrated. The calibration module uses a 25MHz crystal oscillator clock to sample a 32KHz clock, and cooperates with the software process operation. In the coarse tune mode, a 32KHz clock cycle is sampled with a 25MHz crystal oscillator clock. The software adjusts the configuration register RTC_ANA_CALIB[8:0] according to the number of sampling pulses, speeding up or slowing down the 32KHz oscillation clock cycle time to improve 32KHz clock accuracy. After completing the coarse adjustment, you can further enter the fine tune mode. The default value is to sample 256 32KHz clock cycles with a 25MHz crystal oscillator clock. The software averages the number of sampled pulses to obtain the number of pulses required for counting the 32KHz clock in 1 second. And write the value to the register RTC_SEC_PULSE_GEN_INT, RTC_SEC_PULSE_GEN_FRAC to complete the second calibration.

The 32KHz clock coarse adjustment software process is as follows:

1. Set register RTC_ANA_SEL_FTUNE to 0, and the initial value of RTC_ANA_CALIB is 0x100
2. The following uses the bindary search method to achieve calibration:

   FTUNE = RTC_ANA_CALIB; offset = 0x80
3. Set the configuration register RTC_FC_COARSE_EN to 1 and start coarse adjustment. Poll the value of RTC_FC_COARSE_TIME until it is greater than the previous read value, configure RTC_FC_COARSE_EN to 0
4. Read RTC_FC_COARSE_VALUE and obtain the count value of a 32KHz clock cycle sampled by a 25MHz clock.

   if (RTC_FC_COARSE_VALUE > 770) FTUNE = FTUNE + offset;

   if (RTC_FC_COARSE_VALUE < 755) FTUNE = FTUNE - offset;

   Write FTUNE value back to register RTC_ANA_CALIB

   offset = offset >> 1;
5. When the value of RTC_FC_COARSE_VALUE is between 755~770, the 32KHz clock accuracy has reached 32,768Hz ± 1%. Finish coarse adjustment. Otherwise, wait for 0.5ms and repeat steps 3 ~ 5, up to 8 times to complete.

The 32KHz clock fine adjustment calibration software process is as follows:

1. Set register RTC_SEL_SEC_PULSE to 0. Set RTC_FC_FINE_EN to 1 and start fine tuning.
2. Poll the value of RTC_FC_FINE_TIME until it is greater than the previous read value
3. Read RTC_FC_FINE_VALUE and obtain the count value of 256 32KHz clock cycles sampled by 25MHz clock.
4. The 32KHz clock frequency can be obtained by the following formula:

   Frequency = 256 / (RTC_FC_FINE_VALUE x 40ns)

   For example: 256 / (195310 x 40) = 32768.4194357
5. Take the integer part 32768, write it into the register RTC_SEC_PULSE_GEN_INT, take the decimal part 8-bit = 0.4194357 x 256 = 107, write it into the register RTC_SEC_PULSE_GEN_FRAC
6. Configure RTC_FC_FINE_EN to 0 to end fine tuning

The clock calibration process can be executed once or periodically by the software depending on the needs of the system. In addition to the software calibration process, RTC also supports hardware to perform automatic calibration periodically.

After completing the clock calibration, further initialize the RTC. Only the necessary initialization process is listed below. Most of the remaining parameter registers are only necessary to be configured when it is necessary to optimize the chip power-on and power-off or power-off timing. It is generally recommended to use the default values.

1. Configure the register RTC_POR_DB_MAGIC_KEY value to 0x5AF0 to enable POR debounce to prevent POR false triggering due to a brief voltage drop in the RTC module supply voltage. Debounce time is about 1ms.
2. Configure RTC_SET_SEC_CNTR_VALUE and set the initial value of RTC time count.
3. Write 1 to RTC_SET_SEC_CNTR_TRIG to load the initial count value into the RTC second counter.
4. Poll the RTC_SEC_CNTR_VALUE register value until the read value equals the RTC_SET_SEC_CNTR_VALUE value.
5. Configure RTC_PWR_DET_COMP[0] to 1 to enable battery low-voltage detection, and configure RTC_PWR_DET_SEL[0] to 1 to issue a low-voltage detection interrupt. When the battery voltage is lower than the threshold value, the status signal enters low level. The threshold value can be adjusted through the configuration register RTC_PWR_DET_COMP[12:8].
6. After the RTC is powered on for the first time, the register reg_rtcsys_rstn_src_sel must be configured from the default value 0 to 1 so that the RTC subsystem can remain in the working state after the chip is suspend or power-down. If the value of this register is 0, the RTC subsystem will be reset when the chip is powered off.
7. After the RTC is powered on for the first time, the register RTC_EN_AUTO_POWER_UP must be configured from the default value 1 to 0. If the default value is maintained, when the chip enters the power-down state, the RTC will automatically enter the power-on state when it detects that PWR_VBAT_DET is high level.

6.4.4 Analog seconds clock initialization

1. Configure RTC_MACRO_RG_DEFD = 16'hC80 (32000 clock cycles in 32KHz)
2. Configure RTC_MACRO_DA_SOC_READY to 1
3. Configure RTC_MACRO_DA_CLEAR_ALL to 1
4. Configure RTC_MACRO_DA_CLEAR_ALL to 0
5. Configure RTC_MACRO_RG_SET_T as the required Counter value
6. Configure RTC_MACRO_DA_LATCH_PASS to 1

7. Configure RTC_MACRO_DA_LATCH_PASS to 0
8. Configure RTC_MACRO_DA_SOC_READY to 0
9. Read RTC_MACRO_RO_T to get the counter value

6.4.5 Interrupt handling

The RTC will send status signals of alarm interruption and low-voltage detection interruption. When the system receives the alarm interruption, it indicates that the scheduled time has expired, and the user can perform corresponding custom operations. Set register RTC_ALARM_ENABLE to 0 to clear the interrupt status. If you need to continue setting a new timing time, write the new value to the register RTC_ALARM_TIME and set RTC_ALARM_ENABLE to 1 again.

6.4.6 Sleep and wake up

The system software can power off the chip and enter the sleep state (suspend) by configuring the register req_suspend to 1. The configuration register RTC_EN_PWR_WAKEUP selects the source that triggers the chip wakeup. It should be noted that before configuring req_suspend, the register RTC_PG_REG must be written to 0 to allow the DDR IO to enter a constant state (retent) to avoid the DDR data to be damaged due to malfunction during power-off or power-on of the chip. When the chip wakes up, the system software must write 1 to the register RTC_PG_REG to release the protection state of the DDR IO before executing the DDR initialization process.

In addition, if you want to use buttons (PWR_ON, PWR_BUTTON, PWR_WAKEUP) to wake up, before entering the sleep state, you must first configure the relevant IO PINMUX register and lock the IO as RTC input function.

6.4.7 Power-Off and Powner-On

By configuring the register req_shdn to 1, the system software can make the chip including DDR power off and enter the shutdown state. The configuration register RTC_EN_PWR_UP selects the source that can trigger the chip power-on.

6.5 RTC Register Overview

The RTC register contains multiple parts, RTC_CORE_REG, RTC_MACRO_REG and RTC_CTRL_REG. The register base addresses are different and they are all accessed through the bus.

The overview of RTC_CORE_REG register is shown in *RTC_CORE_REG Overview (Base: 0x05026000)*.

Table 6.1: RTC_CORE_REG Overview (Base: 0x05026000)

Name	Address Offset	Description
RTC_ANA_CALIB	0x000	32K oscillator control
RTC_SEC_PULSE_GEN	0x004	Second pulse generator integer and decimal digits
RTC_ALARM_TIME	0x008	scheduled alarm time
RTC_ALARM_ENABLE	0x00c	Enable alarm
RTC_SET_SEC_CNTR_VALUE	0x010	Set seconds counter value
RTC_SET_SEC_CNTR_TRIG	0x014	Loading seconds counter value
RTC_SEC_CNTR_VALUE	0x018	Read current second counter value
RTC_INFO0	0x01c	Information register 0
RTC_INFO1	0x020	Information register 1
RTC_INFO2	0x024	Information register 2
RTC_INFO3	0x028	Information register 3
RTC_NOPOR_INFO0	0x02c	No reset information register 0
RTC_NOPOR_INFO1	0x030	No reset information register 1
RTC_NOPOR_INFO2	0x034	No reset information register 2
RTC_NOPOR_INFO3	0x038	No reset information register 3
RTC_DB_PWR_VBAT_DET	0x040	PWR_VBAT_DET debounce time
RTC_DB_BUTTON1	0x048	PWR_BUTTON1debounce time
RTC_DB_PWR_ON	0x04c	PWR_ONdebounce time
RTC_7SEC_RESET	0x050	Set the number of seconds to press and hold PWR_BUTTON to force reset
RTC_THM_SHDN_AUTO_REBOOT	0x064	Select REQ_THM_SHDN action
RTC_POR_DB_MAGIC_KEY	0x068	Enable POR long-term debounce
RTC_DB_SEL_PWR	0x06c	Select PWR_BUTTON debounce mode
RTC_UP_SEQ0	0x070	Power-on PWR_SEQ0 output timing
RTC_UP_SEQ1	0x074	Power-on PWR_SEQ1 output timing
RTC_UP_SEQ2	0x078	Power-on PWR_SEQ2 output timing
RTC_UP_SEQ3	0x07c	Power-on PWR_SEQ3 output timing
RTC_UP_IF_EN	0x080	Power-on ISO release timing
RTC_UP_RSTN	0x084	Power-on system reset release sequence
RTC_UP_MAX	0x088	Power-on process completion timing
RTC_DN_SEQ0	0x090	Power off PWR_SEQ0 output timing
RTC_DN_SEQ1	0x094	Power off PWR_SEQ1 output timing
RTC_DN_SEQ2	0x098	Power off PWR_SEQ2 output timing
RTC_DN_SEQ3	0x09c	Power off PWR_SEQ3 output timing
RTC_DN_IF_EN	0x0a0	Power off ISO open timing
RTC_DN_RSTN	0x0a4	Power-off system reset sequence
RTC_DN_MAX	0x0a8	Power-off process completion timing
RTC_PWR_CYC_MAX	0x0b0	Power-cycle completion timing
RTC_WARM_RST_MAX	0x0b4	Warm-reset completion timi
RTC_EN_7SEC_RST	0x0b8	Set PWR_BUTTON1 7SEC reset mode

continues on next page

Table 6.1 – continued from previous page

Name	Address Offset	Description
RTC_EN_PWR_WAKEUP	0x0bc	Set sleep wake-up source
RTC_EN_SHDN_REQ	0x0c0	Enable REQ_SHDN
RTC_EN_THM_SHDN	0x0c4	Enable REQ_THM_SHDN
RTC_EN_PWR_CYC_REQ	0x0c8	Enable REQ_PWR_CYC
RTC_EN_WARM_RST_REQ	0x0cc	Enable REQ_WARM_RST
RTC_EN_PWR_VBAT_DET	0x0d0	Enable state machine reference PWR_VBAT_DET
FSM_STATE	0x0d4	RTC state machine value
RTC_EN_WDG_RST_REQ	0x0e0	Enable REQ_WDG_RST
RTC_EN_SUSPEND_REQ	0x0e4	Enable REQ_SUSPEND
RTC_DB_REQ_WDG_RST	0x0e8	REQ_WDG_RST debounce time
RTC_DB_REQ_SUSPEND	0x0ec	REQ_SUSPEND debounce time
RTC_PG_REG	0x0f0	Power Good Register
RTC_ST_ON_REASON	0x0f8	Power-on status register
RTC_ST_OFF_REASON	0x0fc	Power-down status register
RTC_EN_WAKEUP_REQ	0x120	Enable REQ_WAKEUP
RTC_PWR_WAKEUP_POLARITY	0x128	Select PWR_WAKEUP low level
RTC_DB_SEL_REQ	0x130	Select debounce mode
RTC_PWR_DET_SEL	0x140	Select low voltage detection signal source

The overview of RTC_MACRO_REG register is shown in *RTC_MACRO_REG Overview (Base 0x05026400)*.

Table 6.2: RTC_MACRO_REG Overview (Base 0x05026400)

Name	Address Offset	Description
RTC_PWR_DET_COMP	0x044	Low voltage detection control
RTC_MACRO_DA_CLEAR_ALL	0x080	DA_CLEAR_ALL
RTC_MACRO_DA_SET_ALL	0x084	DA_SEL_ALL
RTC_MACRO_DA_LATCH_PASS	0x088	DA_LATCH_PASS
RTC_MACRO_DA_SOC_READY	0x08c	DA_SOC_READY
RTC_MACRO_PD_SLDO	0x090	PD_SLDO
RTC_MACRO_RG_DEFD	0x094	RG_DEFD
RTC_MACRO_RG_SET_T	0x098	RG_SET_T
RTC_MACRO_RO_CLK_STOP	0x0a0	RO_CLK_STOP
RTC_MACRO_RO_DEFQ	0x0a4	RO_DEFQ
RTC_MACRO_RO_T	0x0a8	RO_T

The overview of RTC_CTRL_REG register is shown in *RTC_CTRL_REG Overview (Base: 0x05025000)*.

Table 6.3: RTC_CTRL_REG Overview (Base: 0x05025000)

Name	Address Offset	Description
rtc_ctrl_version	0x000	rtc_ctrl_version
rtc_ctrl_unlockkey	0x004	rtc_ctrl_unlockkey
rtc_ctrl0	0x008	rtc_ctrl0
rtc_ctrl_status0	0x00c	rtc_ctrl_status0
rtc_ctrl_status1	0x010	rtc_ctrl_status1
rtc_ctrl_status2gpio	0x014	rtc_ctrl_status2gpio
rtcsys_rst_ctrl	0x018	rtcsys_rst_ctrl

continues on next page

Table 6.3 – continued from previous page

Name	Address Offset	Description
rtcsys_clkmux	0x01c	rtcsys_clkmux
rtcsys_mcu51_ctrl0	0x020	rtcsys_mcu51_ctrl0
rtcsys_mcu51_ctrl1	0x024	rtcsys_mcu51_ctrl1
rtcsys_pmu	0x028	rtcsys_pmu
rtcsys_status	0x02c	rtcsys_status
rtcsys_clkbyp	0x030	rtcsys_clkbyp
rtcsys_clk_en	0x034	rtcsys_clk_en
rtcsys_wkup_ctrl	0x038	rtcsys_wkup_ctrl
rtcsys_clkdiv	0x03c	rtcsys_clkdiv
fc_coarse_en	0x040	fc_coarse_en
fc_coarse_cal	0x044	fc_coarse_cal
fc_fine_en	0x048	fc_fine_en
fc_fine_period	0x04c	fc_fine_period
fc_fine_cal	0x050	fc_fine_cal
rtcsys_pmu2	0x054	rtcsys_pmu2
rtcsys_clkdiv1	0x058	rtcsys_clkdiv1
rtcsys_mcu51_dbg	0x05c	rtcsys_mcu51_dbg
sw_reg0	0x060	sw_reg0
sw_reg1_por	0x064	sw_reg1_por
fab_lp_ctrl	0x068	fab_lp_ctrl
fab_option	0x06c	fab_option
rtcsys_mcu51_ictrl1	0x07c	rtcsys_mcu51_ictrl1
rtc_ip_pwr_req	0x080	rtc_ip_pwr_req
rtc_ip_iso_ctrl	0x084	rtc_ip_iso_ctrl
rtcsys_spare_reg0	0x088	rtcsys_spare_reg0
rtcsys_spare_reg1	0x08c	rtcsys_spare_reg1
rtcsys_spare_ro	0x090	rtcsys_spare_ro
rtcsys_wkup_ctrl1	0x094	rtcsys_wkup_ctrl1
rtcsys_sram_ctrl	0x098	rtcsys_sram_ctrl
rtcsys_io_ctrl	0x09c	rtcsys_io_ctrl
rtcsys_wdt_ctrl	0x0a0	rtcsys_wdt_ctrl
rtcsys_irrx_clk_ctrl	0x0a4	rtcsys_irrx_clk_ctrl
rtcsys_rtc_wkup_ctrl	0x0a8	rtcsys_rtc_wkup_ctrl
rtcsys_por_rst_ctrl	0x0ac	rtcsys_por_rst_ctrl

6.6 RTC register description

6.6.1 RTC_CORE_REG

6.6.1.1 RTC_ANA_CALIB

Table 6.4: RTC_ANA_CALIB, Offset Address: 0x000

Bits	Name	Access	Description	Reset
15:0	RTC_ANA_CALIB	R/W	Adjust analog module 32K oscillator frequency	0x100
17:16	RTC_ANA_ISEL	R/W	Adjust analog module 32K XTAL current 00 = 2uA, 01 = 1.5uA, 11 = 0.5uA	0x3
30:18	Reserved			
31	RTC_ANA_SEL_FTUNE	R/W	Select 32K OSC calibration value source 0 = Controlled by register RTC_ANA_CALIB 1 = Controlled by hardware circuit	0x1

6.6.1.2 RTC_SEC_PULSE_GEN

Table 6.5: RTC_SEC_PULSE_GEN, Offset Address: 0x004

Bits	Name	Access	Description	Reset
7:0	RTC_SEC_PULSE_GEN_FRAC	R/W	Decimal part of second pulse generator	0x0
23:8	RTC_SEC_PULSE_GEN_INT	R/W	second pulse generator integer part When the counter increment value is greater than the integer bit value, a second pulse signal is generated to increase the second counter by 1.	0x8000
30:24	Reserved			
31	RTC_SEL_SEC_PULSE	R/W	Select second pulse signal source 0 = signal generated internally 1 = signal generated by an external hardware circuit When set as 1 RTC_SEL_PULSE_GEN_FRAC & RTC_SEL_P ULSE_GEN_INThas no effect	0x1

6.6.1.3 RTC_ALARM_TIME

Table 6.6: RTC_ALARM_TIME, Offset Address: 0x008

Bits	Name	Access	Description	Reset
31:0	RTC_ALARM_TIME	R/W	Set scheduled alarm time	0xffffffff

6.6.1.4 RTC_ALARM_ENABLE

Table 6.7: RTC_ALARM_ENABLE, Offset Address: 0x00c

Bits	Name	Access	Description	Reset
0	RTC_ALARM_ENABLE	R/W	Alarm enable 1 = enable the alarm 0 = disable the alarm or clear the alarm interrupt status	0x0
31:1	Reserved			

6.6.1.5 RTC_SET_SEC_CNTR_VALUE

Table 6.8: RTC_SET_SEC_CNTR_VALUE, Offset Address: 0x010

Bits	Name	Access	Description	Reset
31:0	RTC_SET_SEC_CNTR_VALUE	R/W	Set seconds counter value	0x0

6.6.1.6 RTC_SET_SEC_CNTR_TRIG

Table 6.9: RTC_SET_SEC_CNTR_TRIG, Offset Address: 0x014

Bits	Name	Access	Description	Reset
0	RTC_SET_SEC_CNTR_TRIG	W1C	Load seconds counter value 1 = make the value of RTC_SET_SEC_CN TR_VALUE effective, then the register will be cleared to 0	
31:1	Reserved			

6.6.1.7 RTC_SEC_CNTR_VALUE

Table 6.10: RTC_SEC_CNTR_VALUE, Offset Address: 0x018

Bits	Name	Access	Description	Reset
31:0	RTC_SEC_CNTR_VALUE	RO	Read the current seconds counter value	

6.6.1.8 RTC_INFO0

Table 6.11: RTC_INFO0, Offset Address: 0x01c

Bits	Name	Access	Description	Reset
31:0	RTC_INFO0	R/W	Information register 0	0xABCD1234

6.6.1.9 RTC_INFO1

Table 6.12: RTC_INFO1, Offset Address: 0x020

Bits	Name	Access	Description	Reset
31:0	RTC_INFO1	R/W	Information register 1	0xDEADBEEF

6.6.1.10 RTC_INFO2

Table 6.13: RTC_INFO2, Offset Address: 0x024

Bits	Name	Access	Description	Reset
31:0	RTC_INFO2	R/W	Information register 2	0xABCD1234

6.6.1.11 RTC_INFO3

Table 6.14: RTC_INFO3, Offset Address: 0x028

Bits	Name	Access	Description	Reset
31:0	RTC_INFO3	R/W	Information register 3	0xDEADBEEF

6.6.1.12 RTC_NOPOR_INFO0

Table 6.15: RTC_NOPOR_INFO0, Offset Address: 0x02c

Bits	Name	Access	Description	Reset
31:0	RTC_NOPOR_INFO0	R/W	No reset information register 0	Ra ndom

6.6.1.13 RTC_NOPOR_INFO1

Table 6.16: RTC_NOPOR_INFO1, Offset Address: 0x030

Bits	Name	Access	Description	Reset
31:0	RTC_NOPOR_INFO1	R/W	No reset information register 1	Ra ndom

6.6.1.14 RTC_NOPOR_INFO2

Table 6.17: RTC_NOPOR_INFO2, Offset Address: 0x034

Bits	Name	Access	Description	Reset
31:0	RTC_NOPOR_INFO2	R/W	No reset information register 2	Ra ndom

6.6.1.15 RTC_NOPOR_INFO3

Table 6.18: RTC_NOPOR_INFO3, Offset Address: 0x038

Bits	Name	Access	Description	Reset
31:0	RTC_NOPOR_INFO3	R/W	No reset information register 3	Random

6.6.1.16 RTC_APB_BUSY_SEL

Table 6.19: RTC_APB_BUSY_SEL, Offset Address: 0x03c

Bits	Name	Access	Description	Reset
3:0	Reserved	R/W		
4	rtc_apb_32k_busy_sel	R/W	Select rtc pclk busy as signal source 0 = pclk busy signal generated by hardware circuits 1 = pclk busy signal controlled by rtc_apb_32k_force_busy	0x0
7:5	Reserved			
8	rtc_apb_32k_force_busy	R/W	1 = pclk Always operates at full speed 0 = pclk only operates at full speed during psel	0x0
31:9	Reserved			

6.6.1.17 RTC_DB_PWR_VBAT_DET

Table 6.20: RTC_DB_PWR_VBAT_DET, Offset Address: 0x040

Bits	Name	Access	Description	Reset
15:0	RTC_DB_PWR_VBAT_DET	R/W	PWR_VBAT_DET debounce time(unit: 32K clock)	0x2
31:16	Reserved			

6.6.1.18 RTC_DB_BUTTON1

Table 6.21: RTC_DB_BUTTON1, Offset Address: 0x048

Bits	Name	Access	Description	Reset
15:0	RTC_DB_BUTTON1	R/W	PWR_BUTTON debounce time(unit: 32K clock) default value 0x100 is about 8ms	0x100
31:16	Reserved			

6.6.1.19 RTC_DB_PWR_ON

Table 6.22: RTC_DB_PWR_ON, Offset Address: 0x04c

Bits	Name	Access	Description	Reset
15:0	RTC_DB_PWR_ON	R/W	PWR_ON debounce time(unit: 32K clock)	0x100
31:16	Reserved			

6.6.1.20 RTC_7SEC_RESET

Table 6.23: RTC_7SEC_RESET, Offset Address: 0x050

Bits	Name	Access	Description	Reset
7:0	RTC_7SEC_RESET	R/W	Long press PWR_BUTTON1 reset debounce time (unit:second)	0x7
15:8	Reserved			
31:16	RTC_7SEC_UNLOCK_KEY	WO	Writing 0xDC78 at the same time, releasing RTC_7SEC_RESET write protection	0x0

6.6.1.21 RTC_THM_SHDN_AUTO_REBOOT

Table 6.24: RTC_THM_SHDN_AUTO_REBOOT, Offset Address: 0x064

Bits	Name	Access	Description	Reset
0	RTC_THM_SHDN_AUTO_REBOOT	R/W	When choosing to receive REQ_THM_SHDN: 0 = Start the power-off process 1 = Start the power-cycle process	0x0
31:1	Reserved			

6.6.1.22 RTC_POR_DB_MAGIC_KEY

Table 6.25: RTC_POR_DB_MAGIC_KEY, Offset Address: 0x068

Bits	Name	Access	Description	Reset
15:0	RTC_POR_DB_MAGIC_KEY	R/W	Writing 0x5AF0, will cause POR debounce (about 1ms)	Random
31:16	Reserved			

6.6.1.23 RTC_DB_SEL_PWR

Table 6.26: RTC_DB_SEL_PWR, Offset Address: 0x06c

Bits	Name	Access	Description	Reset
0	Reserved			
1	DB_SEL_PWR_BUTTON1	R/W	Choose PWR_BUTTON1 debounce mode 0 = The state machine is triggered by the falling edge of PWR_BUTTON1 debounce signal 1 = The state machine is triggered by the low level of PWR_BUTTON1 debounce signal	0x1
2	DB_SEL_PWR_ON	R/W	Choose PWR_ON debounce mode 0 = The state machine is triggered by the rising edge of PWR_ON debounce signal 1 = The state machine is triggered by the high level of PWR_ON debounce signal	0x1
3	DB_SEL_PWR_WAKEUP0	R/W	Choose PWR_WAKEUP0 debounce mode 0 = The state machine is triggered by the rising edge of PWR_WAKEUP0 debounce signal 1 = The state machine is triggered by the high level of PWR_WAKEUP0 debounce signal	0x1
4	DB_SEL_PWR_WAKEUP1	R/W	Choose PWR_WAKEUP1 debounce mode 0 = The state machine is triggered by the rising edge of PWR_WAKEUP1 debounce signal 1 = The state machine is triggered by the high level of PWR_WAKEUP1 debounce signal	0x1
31:5	Reserved			

6.6.1.24 RTC_UP_SEQ0

Table 6.27: RTC_UP_SEQ0, Offset Address: 0x070

Bits	Name	Access	Description	Reset
15:0	RTC_UP_SEQ0	R/W	Required time for PWR_SEQ0 process output change from 0 to 1 (unit:32K clock)	0x0
31:16	Reserved			

6.6.1.25 RTC_UP_SEQ1

Table 6.28: RTC_UP_SEQ1, Offset Address: 0x074

Bits	Name	Access	Description	Reset
15:0	RTC_UP_SEQ1	R/W	Required time for PWR_SEQ1 process output change from 0 to 1 (unit:32K clock)	0x40
31:16	Reserved			

6.6.1.26 RTC_UP_SEQ2

Table 6.29: RTC_UP_SEQ2, Offset Address: 0x078

Bits	Name	Access	Description	Reset
15:0	RTC_UP_SEQ2	R/W	Required time for PWR_SEQ2 process output change from 0 to 1 (unit:32K clock)	0x80
31:16	Reserved			

6.6.1.27 RTC_UP_SEQ3

Table 6.30: RTC_UP_SEQ3, Offset Address: 0x07c

Bits	Name	Access	Description	Reset
15:0	RTC_UP_SEQ3	R/W	Required time for PWR_SEQ3 process output change from 0 to 1 (unit:32K clock)	0xc0
31:16	Reserved			

6.6.1.28 RTC_UP_IF_EN

Table 6.31: RTC_UP_IF_EN, Offset Address: 0x080

Bits	Name	Access	Description	Reset
15:0	RTC_UP_IF_EN	R/W	Power-on process releases the power-off area ISO time (unit:32K clock)	0x100
31:16	Reserved			

6.6.1.29 RTC_UP_RSTN

Table 6.32: RTC_UP_RSTN, Offset Address: 0x084

Bits	Name	Access	Description	Reset
15:0	RTC_UP_RSTN	R/W	Power-on process system reset release time (unit:32K clock)	0x140
31:16	Reserved			

6.6.1.30 RTC_UP_MAX

Table 6.33: RTC_UP_MAX, Offset Address: 0x088

Bits	Name	Access	Description	Reset
15:0	RTC_UP_MAX	R/W	Complete power-on process completion time (unit:32K clock) RTC_UP_SEQ0~RTC_UP_MAX is the absolute time of each stage of the power-on process. It is recommended to use the default value	0x180
31:16	Reserved			

6.6.1.31 RTC_DN_SEQ0

Table 6.34: RTC_DN_SEQ0, Offset Address: 0x090

Bits	Name	Access	Description	Reset
15:0	RTC_DN_SEQ0	R/W	Required time for PWR_SEQ0 process output change from 1 to 0 (unit:32K clock)	0x140
31:16	Reserved			

6.6.1.32 RTC_DN_SEQ1

Table 6.35: RTC_DN_SEQ1, Offset Address: 0x094

Bits	Name	Access	Description	Reset
15:0	RTC_DN_SEQ1	R/W	Required time for PWR_SEQ1 process output change from 1 to 0 (unit:32K clock)	0x100
31:16	Reserved			

6.6.1.33 RTC_DN_SEQ2

Table 6.36: RTC_DN_SEQ2, Offset Address: 0x098

Bits	Name	Access	Description	Reset
15:0	RTC_DN_SEQ2	R/W	Required time for PWR_SEQ2 process output change from 1 to 0 (unit:32K clock)	0xc0
31:16	Reserved			

6.6.1.34 RTC_DN_SEQ3

Table 6.37: RTC_DN_SEQ3, Offset Address: 0x09c

Bits	Name	Access	Description	Reset
15:0	RTC_DN_SEQ3	R/W	Required time for PWR_SEQ3 process output change from 1 to 0 (unit:32K clock)	0x80
31:16	Reserved			

6.6.1.35 RTC_DN_IF_EN

Table 6.38: RTC_DN_IF_EN, Offset Address: 0x0a0

Bits	Name	Access	Description	Reset
15:0	RTC_DN_IF_EN	R/W	Power-off process releases the power-off area ISO time (unit:32K clock)	0x40
31:16	Reserved			

6.6.1.36 RTC_DN_RSTN

Table 6.39: RTC_DN_RSTN, Offset Address: 0x0a4

Bits	Name	Access	Description	Reset
15:0	RTC_DN_RSTN	R/W	Power-off process system reset release time (unit:32K clock)	0x0
31:16	Reserved			

6.6.1.37 RTC_DN_MAX

Table 6.40: RTC_DN_MAX, Offset Address: 0x0a8

Bits	Name	Access	Description	Reset
15:0	RTC_DN_MAX	R/W	Complete power-off process completion time (unit:32K clock) RTC_DN_SEQ0~RTC_DN_MAX is the absolute time of each stage of the power-OFF process. It is recommended to use the default value	0x180
31:16	Reserved			

6.6.1.38 RTC_PWR_CYC_MAX

Table 6.41: RTC_PWR_CYC_MAX, Offset Address: 0x0b0

Bits	Name	Access	Description	Reset
15:0	RTC_PWR_CYC_MAX	R/W	Complete power-cycle process completion time (unit:32K clock)	0x4000
31:16	Reserved			

6.6.1.39 RTC_WARM_RST_MAX

Table 6.42: RTC_WARM_RST_MAX, Offset Address: 0x0b4

Bits	Name	Access	Description	Reset
15:0	RRTC_WARM_RST_MAX	R/W	Complete WARM_RESET process completion time (unit:32K clock)	0x40
31:16	Reserved			

6.6.1.40 RTC_EN_7SEC_RST

Table 6.43: RTC_EN_7SEC_RST, Offset Address: 0x0b8

Bits	Name	Access	Description	Reset
0	RTC_EN_7SEC_RST	R/W	Enable long press PWR_BUTTON1 to trigger 7-second RTC forced reset	0x0
1	RTC_7SEC_RST_MODE	R/W	7-second forced reset mode 0 = Low level mode, 1 = Short pulse mode When 7-second forced reset occurs, select to generate a short reset signal or hold reset until releasing PWR_BUTTON1	0x0
2	DB_SEL_PWR_BUTTON1_7SEC	R/W	0 = If PWR_BUTTON1 is triggered by the power-on current after PWR_BUTTON1 is triggered, reset the 7-sec reset counter 1 = NOP	0x0
3	SEL_7SEC_RST_RTCSYS	R/W	0 = 7-second forced reset signal will reset the RTC subsystem sec reset will not reset rtcsys 1 = 7-second forced reset signal will not reset RTC subsystem	0x1
15:4	Reserved			
31:16	**RTC_EN_7SEC_UNLOCK_KEY | WO**		Simultaneously write 0xDC78 to unlock [3:0] write protection	0x0

6.6.1.41 RTC_EN_PWR_WAKEUP

Table 6.44: RTC_EN_PWR_WAKEUP, Offset Address: 0x0bc

Bits	Name	Access	Description	Reset
6:0	RTC_EN_PWR_WAKEUP	R/W	Set the sources capable of waking up from sleep mode 0 = Cannot trigger wakeup 1 = Can trigger wakeup [0] = PWR_WAKEUP0 [1] = PWR_WAKEUP1 [2] = PWR_ON [3] = REQ_POWERUP [4] = PWR_BUTTON1 [5] = Alarm [6] = REQ_WAKEUP	0x0
7	Reserved			
14:8	RTC_EN_PWR_UP	R/W	Set the sources capable of power up from off state 0 = Cannot trigger power on 1 = Can trigger power on [8] = PWR_WAKEUP0 [9] = PWR_WAKEUP1 [10] = PWR_ON [11] = REQ_POWERUP [12] = PWR_BUTTON1 [13] = Alarm [14] = REQ_WAKEUP	0x14
31:15	Reserved			

6.6.1.42 RTC_EN_SHDN_REQ

Table 6.45: RTC_EN_SHDN_REQ, Offset Address: 0x0c0

Bits	Name	Access	Description	Reset
0	RTC_EN_SHDN_REQ	R/W	Enable software to equest power-off =(REQ_SHDN) 0 = disable, 1 = enable	0x0
31:1	Reserved			

6.6.1.43 RTC_EN_THM_SHDN

Table 6.46: RTC_EN_THM_SHDN, Offset Address: 0x0c4

Bits	Name	Access	Description	Reset
0	RTC_EN_THM_SHDN	R/W	Enable request thermal shutdown or reboot (REQ_THM_SHDN) 0 = disable, 1 = enable	0x0
31:1	Reserved			

6.6.1.44 RTC_EN_PWR_CYC_REQ

Table 6.47: RTC_EN_PWR_CYC_REQ, Offset Address: 0x0c8

Bits	Name	Access	Description	Reset
0	RTC_EN_PWR_CYC_REQ	R/W	Enable request Power- cycle (REQ_PWR_CYC) 0 = disable, 1 = enable	0x0
31:1	Reserved			

6.6.1.45 RTC_EN_WARM_RST_REQ

Table 6.48: RTC_EN_WARM_RST_REQ, Offset Address: 0x0cc

Bits	Name	Access	Description	Reset
0	RTC_EN_WARM_RST_REQ	R/W	Enable system warm reset request (REQ_WARM_RST) 0 = disable, 1 = enable	0x0
31:1	Reserved			

6.6.1.46 RTC_EN_PWR_VBAT_DET

Table 6.49: RTC_EN_PWR_VBAT_DET, Offset Address: 0x0d0

Bits	Name	Access	Description	Reset
0	RTC_EN_PWR_VBAT_DET _UP C_EN_PWR_VBAT_DET_UP	R/W	Enable battery low voltage state machine reference detection status (PWR_VBAT_DET) 0 = disable, 1 = enable If set to 1, when any key attempts to power up or wake up, the state machine will check the low voltage detection output value. If the low voltage detection output is low (battery voltage too low or no power source), the RTC state machine will maintain the current state.	0x1
1	RTC_EN_PWR_VBAT_DET _DN C_EN_PWR_VBAT_DET_DN	R/W	Enable battery low voltage state trigger down power 0 = disable, 1 = enable If set to 1, when the chip is in normal power up or has entered sleep, the RTC state machine will check the low voltage detection output. If the low voltage detection output changes from high to low (battery voltage too low or power loss), the state machine will trigger the down power process.	0x1
2	RTC_EN_AUTO_POWER_UP	R/W	Enable RTC state machine automatic entry into power up state 1 = When entering power down and PWR_VBAT_DET is high, automatically power up 0 = Stay in this state when entering power down, until any power source is triggered	0x1
31:3	Reserved			

6.6.1.47 FSM_STATE

Table 6.50: FSM_STATE, Offset Address: 0x0d4

Bits	Name	Access	Description	Reset
3:0	FSM_STATE	RO	RTC state machine value 4'h0 = ST_OFF (System power off completed) 4'h1 = ST_UP (Power-up process ongoing) 4'h2 = ST_DN (Power-down process ongoing) 4'h3 = ST_ON (System power on completed) 4'h4 = ST_PWR_CYC2 (Power-cycle power down process completed) 4'h6 = ST_PWR_CYC (Power-cycle power down in progress) 4'h7 = ST_WARM_RESET (System reset in progress) 4'h9 = ST_SUSP (System suspended) 4'hB = ST_PRE_SUSP (Suspend power-down process ongoing)	
31:4	Reserved			

6.6.1.48 RTC_EN_WDG_RST_REQ

Table 6.51: RTC_EN_WDG_RST_REQ, Offset Address: 0x0e0

Bits	Name	Access	Description	Reset
0	RTC_EN_WDG_RST_REQ	R/W	Enable Watchdog reset request (REQ_WDG_RST) 0 = disable, 1 = enable	0x0
1	RTC_EN_SUS_WDG_RST_REQ	R/W	Enable Watchdog reset request when in sleep state 0 = disable, 1 = enable	0x1
31:2	Reserved			

6.6.1.49 RTC_EN_SUSPEND_REQ

Table 6.52: RTC_EN_SUSPEND_REQ, Offset Address: 0x0e4

Bits	Name	Access	Description	Reset
0	RTC_EN_SUSPEND_REQ	R/W	Enable request sleep (REQ_SUSPEND) 0 = disable, 1 = enable	0x0
31:1	Reserved			

6.6.1.50 RTC_PG_REG

Table 6.53: RTC_PG_REG, Offset Address: 0x0f0

Bits	Name	Access	Description	Reset
3:0	RTC_PG_REG	R/W	Chip Power Good Status 1 = Chip has power (Power Good), IO signal can pass 0 = Chip power down, IO signal retains state (retent) [0] = Control DDR IO [3:1] = reserved This register signal is used to control whether the chip IO interface with DDR is normal pass or retent state. Before the system enters sleep the software must set this register value to 0 to keep DDR IO in a fixed state. After the system wakes up from sleep, it must set this register value to 1 to allow DDR to resume normal operation. When entering power down state, this register value will be automatically cleared to all 1s.	0xF
31:4	Reserved			

6.6.1.51 RTC_ST_ON_REASON

Table 6.54: RTC_ST_ON_REASON, Offset Address: 0x0f8

Bits	Name	Access	Description	Reset
3:0	ST_ON_REASON_LAST_STATE	RO	RTC state machine return to on from the following states: 4'h0 = ST_OFF (Power-off to on) 4'h3 = ST_PWR_CYC (Power-cycle or Warm-reset to on) 4'h9 = ST_SUSP (Suspended to on) After system restart, the software can determine the chip boot-up reason by reading this register.	
15:4	Reserved			
31:16	ST_ON_REASON_LAST_INPUT	RO	The reason (recorded signal state) for returning to the on state of state machine [0] = PWR_VBAT_DET (0: Power- off triggered) [1] = PWR_ON (1: Key triggered power-on) [2] = RTC_EN_AUTO_POWER_UP [3] = PWR_BUTTON1 (0: Key triggered power-on) [4] = PWR_BUTTON1_7SEC [5] = PWR_WAKEUP0 (1: Key triggered wake-up) [6] = PWR_WAKEUP1 (1: Key triggered wake-up) [7] = Alarm (1: Timed alarm) [8] = REQ_PWR_CYC (1: Software triggered power-cycle) [9] = REQ_THM_SHDN (1: Software triggered power-off/ power-cycle) [10] = REQ_WARM_RST (1: Software triggered reset) [11] = REQ_WDG_RST (1: Watchdog triggered reset) [12] = REQ_SHDN (1: Software triggered power-off) [13] = REQ_SUSPEND (1: Software triggered sleep) [14] = REQ_WAKEUP (1: Event triggered wake-up) [15] = REQ_POWERUP	

6.6.1.52 RTC_ST_OFF_REASON

Table 6.55: RTC_ST_OFF_REASON, Offset Address: 0x0fc

Bits	Name	Access	Description	Reset
3:0	ST_OFF_REASON_LAST_STATE	RO	The RTC state machine transitioned to off (ST_OFF) from the following states: 4'h3 = ST_ON (Transitioned from on to off) 4'h9 = ST_SUSP (Transitioned from suspend to off) Other = 7-second forced reset occurred After system restart, software can determine the chip's last power-off reason by reading this register.	
15:4	Reserved			
31:16	ST_OFF_REASON_LAST_INPUT	RO	The reason (recorded signal state) for entering the off state of the state machine [14:0] Same as ST_ON_REASON_LAST_INPUT [15] = 0: 7-second forced reset occurred	

6.6.1.53 RTC_EN_WAKEUP_REQ

Table 6.56: RTC_EN_WAKEUP_REQ, Offset Address: 0x120

Bits	Name	Access	Description	Reset
0	RTC_EN_WAKEUP_REQ	R/W	Enable Wakeup request from sleep state 0 = disable, 1 = enable	0x0
1	RTC_EN_POWERUP_REQ	R/W	Enable power-up event request 0 = disable, 1 = enable	0x0
31:2	Reserved			

6.6.1.54 RTC_PWR_WAKEUP_POLARITY

Table 6.57: RTC_PWR_WAKEUP_POLARITY, Offset Address: 0x128

Bits	Name	Access	Description	Reset
0	RTC_EN_WAKEUP_REQ	R/W	Enable Wakeup request event from sleep state 0 = disable, 1 = enable	0x0
1	RTC_EN_POWERUP_REQ	R/W	Enable event request power up 0 = disable, 1 = enable	0x0
31:2	Reserved			

6.6.1.55 RTC_DB_SEL_REQ

Table 6.58: RTC_DB_SEL_REQ, Offset Address: 0x130

Bits	Name	Access	Description	Reset
0	DB_SEL_REQ_SHDN	R/W	Select software signal REQ_SHDN debounce mode 0 = Rising edge of register value triggers 1 = Pulse signal of register triggers	0x1
1	DB_SEL_REQ_THM_SHDN	R/W	Select signal REQ_THM_SHDN debounce mode 0 = High level of signal triggers 1 = Rising edge of signal triggers	0x1
2	DB_SEL_REQ_PWR_CYC	R/W	Select software signal REQ_PWR_CYC debounce mode 0 = Rising edge of register value triggers 1 = Pulse signal of register triggers	0x1
3	DB_SEL_REQ_WARM_RST	R/W	Select software signal REQ_WARM_RST debounce mode 0 = Rising edge of register value triggers 1 = Pulse signal of register triggers	0x1
4	DB_SEL_REQ_WDG_RST	R/W	Select signal REQ_WDG_RST debounce mode 0 = High level of signal triggers 1 = Rising edge of signal triggers	0x1
5	DB_SEL_REQ_SUSPEND	R/W	Select software signal REQ_SUSPEND debounce mode 0 = Rising edge of register value triggers 1 = Pulse signal of register triggers	0x1
6	DB_SEL_REQ_WAKEUP	R/W	Select signal REQ_WAKEUP debounce mode 0 = High level of signal triggers 1 = Rising edge of signal triggers	0x1
7	DB_SEL_REQ_POWERUP	R/W	Select signal REQ_POWERUP debounce mode 0 = High level of signal triggers 1 = Rising edge of signal triggers	0x1
31:8	Reserved			

6.6.1.56 RTC_PWR_DET_SEL

Table 6.59: RTC_PWR_DET_SEL, Offset Address: 0x140

Bits	Name	Access	Description	Reset
0	pwr_det_o_sel_comp	R/W	Select low voltage detection status signal output source 0 = Direct from IO PWR_VBAT_DET 1 = From analog low voltage detection circuit output Low voltage detection status value can be read from register TC_CTRL_STATUS0[0]	0x0
1	pwr_det_i_sel_comp	R/W	Select RTC state machine low voltage triggered power off signal source 0 = Direct from IO PWR_VBAT_DET 1 = From analog low voltage detection circuit output	0x0
31:2	Reserved			

6.6.2 RTC_MACRO_REG

6.6.2.1 RTC_PWR_DET_COMP

Table 6.60: RTC_PWR_DET_COMP, Offset Address: 0x44

Bits	Name	Access	Description	Reset
0	pwr_det_comp_enable	R/W	Enable analog module low voltage detection 1 = enable 0 = disable	0x0
7:1	Reserved			
12:8	pwr_det_comp_sel	R/W	Set low voltage detection voltage comparison threshold Threshold = 1.20V + (pwr_det_comp_sel * 12.5mV)	0xf
31:13	Reserved			

6.6.2.2 RTC_MACRO_DA_CLEAR_ALL

Table 6.61: RTC_MACRO_DA_CLEAR_ALL, Offset Address: 0x080

Bits	Name	Access	Description	Reset
0	DA_CLEAR_ALL	R/W		0x0
31:1	Reserved			

6.6.2.3 RTC_MACRO_DA_SET_ALL

Table 6.62: RTC_MACRO_DA_SET_ALL, Offset Address: 0x084

Bits	Name	Access	Description	Reset
0	DA_SEL_ALL	R/W		0x0
31:1	Reserved			

6.6.2.4 RTC_MACRO_DA_LATCH_PASS

Table 6.63: RTC_MACRO_DA_LATCH_PASS, Offset Address: 0x088

Bits	Name	Access	Description	Reset
0	DA_LATCH_PASS	R/W		0x0
31:1	Reserved			

6.6.2.5 RTC_MACRO_DA_SOC_READY

Table 6.64: RTC_MACRO_DA_SOC_READY, Offset Address: 0x08c

Bits	Name	Access	Description	Reset
0	DA_SOC_READY	R/W		0x0
31:1	Reserved			

6.6.2.6 RTC_MACRO_PD_SLDO

Table 6.65: RTC_MACRO_PD_SLDO, Offset Address: 0x090

Bits	Name	Access	Description	Reset
0	PD_SLDO	R/W		0x0
31:1	Reserved			

6.6.2.7 RTC_MACRO_RG_DEFD

Table 6.66: RTC_MACRO_RG_DEFD, Offset Address: 0x094

Bits	Name	Access	Description	Reset
15:0	RG_DEFD	R/W		0x7fff
31:16	Reserved			

6.6.2.8 RTC_MACRO_RG_SET_T

Table 6.67: RTC_MACRO_RG_SET_T, Offset Address: 0x098

Bits	Name	Access	Description	Reset
31:0	RG_SET_T	R/W		0x0

6.6.2.9 RTC_MACRO_RO_CLK_STOP

Table 6.68: RTC_MACRO_RO_CLK_STOP, Offset Address: 0x0a0

Bits	Name	Access	Description	Reset
0	RO_CLK_STOP	RO		
31:1	Reserved			

6.6.2.10 RTC_MACRO_RO_DEFQ

Table 6.69: RTC_MACRO_RO_DEFQ, Offset Address: 0x0a4

Bits	Name	Access	Description	Reset
15:0	RO_DEFQ	RO		
31:16	Reserved			

6.6.2.11 RTC_MACRO_RO_T

Table 6.70: RTC_MACRO_RO_T, Offset Address: 0x0a8

Bits	Name	Access	Description	Reset
31:0	RO_T	RO		

6.6.3 RTC_CTRL_REG

6.6.3.1 RTC_CTRL0_UNLOCKKEY

Table 6.71: RTC_CTRL0_UNLOCKKEY, Offset Address: 0x004

Bits	Name	Access	Description	Reset
15:0	rtc_ctrl0_unlockkey	R/W	To configure the RTC_CTRL0 register, you must first write the value 0xAB18 to unlock it from write protection. If unlockkey_clear is set to 1, after writing to RTC_CTRL0 once, this register will automatically clear to 0, and RTC_CTRL0 will revert to write protection.	0x0000
31:16	Reserved			

6.6.3.2 RTC_CTRL0

Table 6.72: RTC_CTRL0, Offset Address: 0x008

Bits	Name	Access	Description	Reset
0	req_shdn	R/W	Request Power Off 0 = No action, 1 = Request to RTC The RTC_EN_SHDN_REQ register must be set to 1 to take effect.	0x0
1	req_sw_thm_shdn	R/W	Software mode request for overheat shutdown or reboot 0 = No action, 1 = Request to RTC The RTC_EN_THM_SHDN register must be set to 1 to take effect.	0x0
2	hw_thm_shdn_en	R/W	Enable hardware mode request for overheat shutdown or reboot 0 = disable, 1 = enable	0x0
3	req_pwr_cyc	R/W	Request Power-cycle 0 = No action, 1 = Request to RTC The RTC_EN_PWR_CYC_REQ register must be set to 1 to take effect.	0x0
4	req_warm_rst	R/W	Request Warm-reset 0 = No action, 1 = Request to RTC The RTC_EN_WARM_RST_REQ register must be set to 1 to take effect.	0x0
5	req_sw_wdg_rst	R/W	Software mode request for Watchdog reset 0 = No action, 1 = Request to RTC The RTC_EN_WDG_RST_REQ register must be set to 1 to take effect.	0x0
6	hw_wdg_rst_en	R/W	Enable hardware mode request for Watchdog reset 0 = disable, 1 = enable	0x0
7	req_suspend	R/W	Request Suspend 0 = No action, 1 = Request to RTC The RTC_EN_SUSPEND_REQ register must be set to 1 to take effect.	0x0
8	unlockkey_clear	R/W	Enable automatic clear of register unlock	0x0
10	reg_rtc_mode	R/W	32K Clock Source 0 = OSC32K, 1 = XTAL32K	0x0
11	reg_clk32k_cg_en	R/W	32K Clock Switch 0 = Off, 1 = On	0x1
31:12	Reserved			

6.6.3.3 RTC_CTRL_STATUS0

Table 6.73: RTC_CTRL_STATUS0, Offset Address: 0x00c

Bits	Name	Access	Description	Reset
0	rtc_pwr_vbat_det_o	RO	Low Voltage Detection Status Signal Output	
1	rtc_pwr_button0_o	RO	PWR_BUTTON0 IO Signal Output	
2	rtc_pwr_button1_o	RO	PWR_BUTTON1 IO Signal Output	
4	rtc_pwr_on_o	RO	PWR_ON IO Signal Output	
5	rtc_pwr_wakeup0_o	RO	PWR_WAKEUP0 IO Signal Output	
6	rtc_pwr_wakeup1_o	RO	PWR_WAKEUP1 IO Signal Output	
7	rtc_mode_o	RO	RTC_MODE IO Signal Output	
21	rtc_alarm_o	RO	Alarm Status	
22	hw_thm_shdn_sta_i	RO	Overheat Restart Status Signal	
23	hw_wdg_rst_sta_i	RO	Watchdog Reset Status Signal	
24	sys_reset_x_i	RO		
25	cg_en_out_clk_32k	RO		
29:26	rtc_fsm_st	RO	RTC State Machine Value	
31:30	Reserved			

6.6.3.4 RTC_CTRL_STATUS1

Table 6.74: RTC_CTRL_STATUS1, Offset Address: 0x010

Bits	Name	Access	Description	Reset
31:0	rtc_sec_value_o	RO	RTC seconds counter value	

6.6.3.5 rtc_ctrl_status2gpio

Table 6.75: rtc_ctrl_status2gpio, Offset Address: 0x014

Bits	Name	Access	Description	Reset
7:0	status2gpio_en	R/W		0x0
31:8	Reserved			

6.6.3.6 rtcsys_rst_ctrl

Table 6.76: rtcsys_rst_ctrl, Offset Address: 0x018

Bits	Name	Access	Description	Reset
0	Reserved			
1	reg_soft_rstn_mcu	R/W	0 : reest MCU	0x0
2	reg_soft_rstn_sdio	R/W	0 : reset SD1	0x1
3	reg_soft_rstn_uart	R/W	0 : reset Uart	0x1
4	reg_soft_rstn_spinor	R/W	0 : reset spinor1	0x1
5	reg_soft_rstn_ictl	R/W	0 : reset dw_ictl	0x1
6	reg_soft_rstn_mbox	R/W	0 : reset mbox	0x1
7	reg_soft_rstn_fab_hs2rtc	R/W	0 : reset hs2rtc	0x1
8	reg_soft_rstn_fab_rtc2ap	R/W	0 : reset rtc2ap	0x1
9	reg_soft_rstn_fab_sram	R/W	0 : reset ahb sram logic	0x1
10	reg_soft_rstn_apb	R/W	no load	0x1
11	reg_soft_rstn_apb_timer	R/W	0 : reset dw timer apb logic	0x1
12	reg_soft_rstn_timer0	R/W	0 : reset dw timer0	0x1
13	reg_soft_rstn_timer1	R/W	0 : reset dw timer1	0x1
14	reg_soft_rstn_osc	R/W	0 : reset osc	0x1
15	reg_soft_rstn_gpio	R/W	0 : reset gpio	0x1
16	reg_soft_rstn_i2c	R/W	0 : reset i2c	0x1
17	reg_soft_rstn_saradc	R/W	0 : reset saradc	0x1
18	reg_soft_rstn_wdt	R/W	0 : reset wdt	0x1
19	reg_soft_rstn_irrx	R/W	0 : reset irrx	0x1
20	reg_soft_rstn_f32kless	R/W	0: reset f32kless	0x1
31:21	Reserved			

6.6.3.7 rtcsys_clkmux

Table 6.77: rtcsys_clkmux, Offset Address: 0x01c

Bits	Name	Access	Description	Reset
3:0	reg_sdio_clk_mux	R/W	clk_sd1_pre 0 : fpll/4 1: osc_div	0x0
7:4	reg_fab_clk_mux	R/W	clk_fab_pre 0 : 32K, 1: fpll/5, 2: osc_div	0x0
9:8	reg_timer0_clk_mux	R/W	0: xtal 1: 32K	0x0
11:10	reg_timer1_clk_mux	R/W	0: xtal 1: 32K	0x0
13:12	reg_apb_clk_mux	R/W	00 : cgdiv and refer to apbactive 01 : force clk_apb, clk_fab 1:1 (default) 10 : force clk_apb, clk_fab 1:2 11 : force clk_apb, clk_fab 1:4	0x1
15:14	Reserved			
17:16	reg_i2c_clk_mux	R/W	0: xtal 1: osc div	0x0
19:18	reg_sd_mclk_clk_mux	R/W	0: 100Khz from OSC, 1: 32K	0x0
20	reg_saradc_clk_mux	R/W	0 : XTAL, 1: OSC DIV	0x0
21	reg_irrx_clk_mux	R/W	0 : XTAL, 1: OSC DIV	0x0
31:22	Reserved			

6.6.3.8 rtcsys_mcu51_ctrl0

Table 6.78: rtcsys_mcu51_ctrl0, Offset Address: 0x020

Bits	Name	Access	Description	Reset
4:0	reg_51_rom_addr_size	R/W	Determines how many of the sixteen internal ROM address bits (irom_addr) are used (0 = no internal ROM present);	0xc
5	reg_51_mem_ea_n	R/W	0 : external rom exist, 1: external rom not exist	0x0
6	reg_51_xdata_mode	R/W	0 : fetch xdata with clock gating 1 : fetch xdata wo clock gating (to support 51 timer and 51 uart)	0x0
7	reg_51_rom_addr_def	R/W	0: mercury define , max internal rom = 2^reg_51_rom_addr_size -1 internal rom offset = 4K*reg_51irom_ioffset 1: mars define , max internal rom = 2 K*reg_51_rom_addr_size -1 internal rom offset = 2K*reg_51irom_ioffset	0x0
10:8	Reserved			
31:11	reg_51xdata_ioffset0	R/W	Set offset address[31:12] to select mcu8051 boot device	0x05200

6.6.3.9 rtcsys_mcu51_ctrl1

Table 6.79: rtcsys_mcu51_ctrl1, Offset Address: 0x024

Bits	Name	Access	Description	Reset
4:0	reg_51irom_ioffset	R/W	boot rom offset to rtcsys_sram	0x0
5	Reserved			
9:6	reg_51_pf_mode	R/W	reg_51_pf_mode	0x0
10	Reserved			
31:11	reg_51xdata_doffset0	R/W	Set offset address[31:12] to select mcu8051 xdata	0x05200

6.6.3.10 rtcsys_pmu

Table 6.80: rtcsys_pmu, Offset Address: 0x028

Bits	Name	Access	Description	Reset
3:0	Reserved			
4	reg_dis_pmu_ldo_ctrl	R/W	disable pmu ldo ctrl 0: enable pmu to ctrl RTC_LDO sleep mode 1: disable pmu to ctrl RTC_LDO sleep mode	0x0
5	reg_wdt_clkoff_by_pmu	R/W	wdt_clk gate by pmu when mcu into idle mode 1. wdt clock gate by pmu	0x0
6	reg_force_osc_off	R/W	1 : force osc off	0x0
7	reg_force_osc_on	R/W	1 : force osc on	0x0
8	reg_pmu_sleep_mode	R/W	pmu enter light sleep mode when mcu idle 1 : enable pmu light sleep mode when mcu idle (pmu control osc_req/ sram slp) 0 disable pmu light sleep mode	0x0
9	reg_pmu_lowpwr_mode	R/W	mcu_pmu into sleep state when rtc at suspend state & mcu idle & reg_pmu_sleep_mode enable 1 : enable mcu_pmu into sleep mode (trigger rtc ldo step down power) 0 disable mcu_pmu sleep mode	0x0
13:10	reg_pmu_stable_cnt	R/W	Stable timer when mcu_pmu leave sleep state, clock unit : 31.25us (32khz), wait for 1~16 tick cycle	0x3
14	reg_xtal_off_by_pmu	R/W	pmu control xtal request 1: xtal request disable by pmu sleep mode	0x0
15	reg_rtcsys_clk25m_req	R/W	xtal request1 for rtcsys 0: disable 25m xtal request1(rtcsys) 1: enable 25m xtal request1 (rtcsys)	0x1
19:16	reg_rtc_vbat_det_db_cnt	R/W	vbat det int debounce time (cycle unit : 32K)	0x2
20	reg_rtc_vbat_det_db_en	R/W	0: disable vbat det int debounce 1: enable vbat det int debounce	0x1
21	reg_ahb_sram_auto_slp_en	R/W	1: enable ahb sram into slp md when bus idle	0x0
23:22	reg_ahb_sram_busy_sel	R/W	2'd0: cs | cs_d1 2'd1: cs | cs_d1 | cs_d2 2'd2: cs | cs_d1 | cs_d2 | cs_d3 3'd3: cs | cs_d1 | cs_d2 | cs_d3 | cs_d4	0x0
24	reg_rtc_stint_clr	W1P	clear rtc state change interrupt	
25	reg_vbat_det_int_clr	W1P	clear vbet det interrupt	
26	reg_rtcsys_clk25m_hw_req	R/W	xtal request1 for rtcsys from hw ip 0: disable 25m xtal request1 from hw ip(rtcsys) 1: enable 25m xtal request1 from hw ip(rtcsys)	0x0
27	Reserved			

To be continued ……

Table 6.81: rtcsys_pmu, Offset Address: 0x028 (continued)

Bits	Name	Access	Description	Reset
28	reg_vbat_det_force_clk	R/W	1: when vbat det happen, change rtcsys bus clock to OSC	0x0
29	reg_mcu_clkoff_by_pmu	R/W	mcu_clk gate by pmu when into idle mode 1. mcu clock gate by pmu	0x1
30	reg_xtal_off_by_susp	R/W	ISO off control xtal request 1: xtal request disable by ISO_OFF	0x0
31	reg_osc_off_by_susp	R/W	ISO off control osc request 1: osc request disable by ISO_OFF	0x0

6.6.3.11 rtcsys_status

Table 6.82: rtcsys_status, Offset Address: 0x02c

Bits	Name	Access	Description	Reset
31:0	reg_rtcsys_status	RO	[0] enable rtc2apb ahb path 0: rtcsys ip can only access 0x05000000+16MB 1: rtcsys ip can access full range address [1] flag of vbat_det_force_clk	

6.6.3.12 rtcsys_clkbyp

Table 6.83: rtcsys_clkbyp, Offset Address: 0x030

Bits	Name	Access	Description	Reset
31:0	reg_clk_byp	R/W	[0] : clk_fab , 0: clk_fab_pre, 1: xtal (default) [1] : clk_sdio, 1: clk_sd1_pre, 1: xtal (default) [31:2]: NA	0xffff ffff

6.6.3.13 rtcsys_clk_en

Table 6.84: rtcsys_clk_en, Offset Address: 0x034

Bits	Name	Access	Description	Reset
31:0	reg_clk_en	R/W	[0]: NA [1]: clk_sd1 (sd1 card clock) [2]: clk_fab_sd1 (sd1 core clock) [3]: clk_mcu [4]: clk_hs2rtc_mst [5]: clk_rtc2ap_slv [6]: clk_spinor1 [7]: clk_fab_sram (AHB sram) [8]: NA [9]: clk_apb_timer [10]: clk_timer0 [11]: clk_timer1 [12]: clk_apb_uart [13]: clk_uart [14]: clk_apb_ictrl [15]: clk_apb_mbox [16]: clk_apb_gpio [17]: clk_apb_osc [18]: clk_gpio_db [19]: clk_apb_i2c [20]: clk_i2c [21]: NA [22]: clk_sd1_tmclk [23]: clk_apb_saradc [24]: clk_saradc [25]: clk_apb_wdt [26]: clk_wdt [27]: clk_irrx [31:28]: NA	0xffffffff

6.6.3.14 rtcsys_wkup_ctrl

Table 6.85: rtcsys_wkup_ctrl, Offset Address: 0x038

Bits	Name	Access	Description	Reset
14:0	reg_rtcsys_wkint_mask	R/W	mask int to RTC_CORE.REQ_WAKEUP/ MCU_PMU [0]: irrrx_intr [1]: gpio_int [2]: timer0_int [3]: timer1_int [4]: saradc_int [5]: rtcsys_ictrl_int [6]: wdt_int [7]: irrx_wakeup	0xff
15	reg_vbat_det_wkup_mask	R/W	1: mask vbat det int	0x1
16	reg_sw_wkint_req	R/W	mcu sw wakeup interrupt to RTC_CORE 1: interrupt active	0x0
2 3:17	Reserved			
24	reg_wkint2rtc_mask	R/W	1: mask wakeup int (rtcsys int) to RTC core	0x1
3 1:25	Reserved			

6.6.3.15 rtcsys_clkdiv

Table 6.86: rtcsys_clkdiv, Offset Address: 0x03c

Bits	Name	Access	Description	Reset
3:0	reg_div_clk_osc_fab_div_val	R/W	Clock Divider Factor	0x1
4	reg_div_clk_osc_fab_dis	R/W	Clock gate	0x0
5	reg_div_clk_osc_fab_hwide	R/W	Select High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider	0x0
15:6	Reserved			
19:16	reg_div_clk_osc_i2c_div_val	R/W	Clock Divider Factor	0x1
20	reg_div_clk_osc_i2c_dis	R/W	Clock gate	0x0
21	reg_div_clk_osc_i2c_hwide	R/W	Select High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider	0x0
23:22	Reserved			
29:24	reg_div_clk_osc_saradc_div_val	R/W	Clock Divider Factor	0x1
30	reg_div_clk_osc_saradc_dis	R/W	Clock gate	0x0
31	reg_div_clk_osc_saradc_hwide	R/W	Select High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider	0x0

6.6.3.16 fc_coarse_en

Table 6.87: fc_coarse_en, Offset Address: 0x040

Bits	Name	Access	Description	Reset
0	fc_coarse_en	R/W	Enable 32K coarse fine tuning 0 = disable, 1 = enable	0x0
31:1	Reserved			

6.6.3.17 fc_coarse_cal

Table 6.88: fc_coarse_cal, Offset Address: 0x044

Bits	Name	Access	Description	Reset
15:0	fc_coarse_value	RO	32K coarse fine tuning counter value (unit: 32K clock) 25MHz clock counts one fc_fine period	
31:16	fc_coarse_time	RO	32K coarse fine tuning completion times	

6.6.3.18 fc_fine_en

Table 6.89: fc_fine_en, Offset Address: 0x048

Bits	Name	Access	Description	Reset
0	fc_fine_en	R/W	Enable 32K fine tuning 0 =disable, 1 = enable	0x0
31:1	Reserved			

6.6.3.19 fc_fine_period

Table 6.90: fc_fine_period, Offset Address: 0x04c

Bits	Name	Access	Description	Reset
15:0	fc_fine_period	R/W	32K fine-tuning count cycle(unit: 32K clock) count 32K clock cycles each time a 25MHz being used	0x0100
31:16	Reserved			

6.6.3.20 fc_fine_cal

Table 6.91: fc_fine_cal, Offset Address: 0x050

Bits	Name	Access	Description	Reset
23:0	fc_fine_value	RO	32K fine counter value (unit 25MHz clock) 25MHz clock counts one fc_fine period	
31:24	fc_fine_time	RO	32K fine-tuning completion times	

6.6.3.21 rtcsys_pmu2

Table 6.92: rtcsys_pmu2, Offset Address: 0x054

Bits	Name	Access	Description	Reset
0	reg_rtc_sys_wkint_db_en	R/W	PMU wakeup int debounce enable	0x1
4:1	reg_rtc_sys_wkint_db_cnt	R/W	PMU wakeup int debounce cycle (32K)	0x2
31:5	Reserved			

6.6.3.22 rtcsys_clkdiv1

Table 6.93: rtcsys_clkdiv1, Offset Address: 0x058

Bits	Name	Access	Description	Reset
15:0	Reserved			
21:16	reg_div_clk_osc_irrx_div_val	R/W	Clock Divider Factor	0x0
22	Reserved			
23	reg_div_clk_osc_irrx_dis	R/W	Clock gate	0x1
31:24	Reserved			

6.6.3.23 rtcsys_mcu51_dbg

Table 6.94: rtcsys_mcu51_dbg, Offset Address: 0x05c

Bits	Name	Access	Description	Reset
3:0	reg_51_dbg_sel	R/W	select mcu51 debug bus (check mcu design review ppt)	0x0
4	reg_51_dbg_snap_shot	W1P	snap shot mcu51 internal register to dbg register (reg_rtcsys_dbg)	
5	reg_51_dbg_step_en	R/W	0: disable mcu debug function 1: enable mcu debug function, and mcu stop at current PC	0x0
6	reg_51_dbg_step	W1P	1: mcu jump to next PC	
7	reg_51_dbg_jump	W1P	1: mcu jump to target pc value (reg_51_dbg_jump2pc)	
15:8	Reserved			
31:16	reg_51_dbg_jump2pc	R/W	16 bit mcu target pc value	0x0

6.6.3.24 sw_reg0

Table 6.95: sw_reg0, Offset Address: 0x060

Bits	Name	Access	Description	Reset
7:0	sw_reg0	R/W	reg for SW	0x0
31:8	Reserved			

6.6.3.25 sw_reg1_por

Table 6.96: sw_reg1_por, Offset Address: 0x064

Bits	Name	Access	Description	Reset
7:0	sw_reg1_por	R/W	reg for SW could only be reset by power reset	0x0
31:8	Reserved			

6.6.3.26 fab_lp_ctrl

Table 6.97: fab_lp_ctrl, Offset Address: 0x068

Bits	Name	Access	Description	Reset
7:0	rtcsys_fab_busy_sel	R/W	select signal to request sys_ctrl to speed up fab clock	0xDF
9:8	rtcsys_fab_busy_ctrl	R/W	rtcsys_fab_busy signal is combi or register out	0x0
11:10	apdbg_busy_ctrl	R/W	apdbg_busy signal is combi or register out	0x0
13:12	reg_apb_busy_ctrl	R/W	apb bridge_busy signal is combi or register out	0x3
15:14	reg_mcu_busy_ctrl	R/W	mcu_busy signal is combi or register out	0x3
31:16	Reserved			

6.6.3.27 rtcsys_mcu51_ictrl1

Table 6.98: rtcsys_mcu51_ictrl1, Offset Address: 0x07c

Bits	Name	Access	Description	Reset
15:0	reg_51_int1_src_mask	R/W	select rtcsys_int src to mcu int1_n 1: mask, 0: un-mask [0]: vbat_det [1]: mbox0_int [2]: NA [3]: irrx_int [4]: gpio_int [5]: uart_int [6]: spinor1_int [7]: timer0_int [8]: timer1_int [9]: irq_ap2rtc[0] [10]: irq_ap2rtc[1] [11]: i2c_int [12]: rtc_state_change_int [13]: hw_thm_shdn [14]: saradc [15]: wdt_int	0xffff
31:16	reg_51_int1_final_status	R0	mcu int1_n status [0]: vbat_det [1]: mbox0_int [2]: NA [3]: irrx_int [4]: gpio_int [5]: uart_int [6]: spinor1_int [7]: timer0_int [8]: timer1_int [9]: irq_ap2rtc[0] [10]: irq_ap2rtc[1] [11]: i2c_int [12]: rtc_state_change_int [13]: hw_thm_shdn [14]: saradc [15]: wdt_int	

6.6.3.28 rtc_ip_pwr_req

Table 6.99: rtc_ip_pwr_req, Offset Address: 0x080

Bits	Name	Access	Description	Reset
0	reg_sd1_pwr_req	R/W	power fence control 1: power on, 0: power off [0]: sd1	0x1
1	reg_sd1_pwr_req_2nd	R/W	power fence control 1: power on, 0: power off [0]: sd1	0x1
2	reg_mcu_pwr_req	R/W	power fence control 1: power on, 0: power off [1]: mcu subsys	0x1
3	reg_mcu_pwr_req_2nd	R/W	power fence control 1: power on, 0: power off [1]: mcu subsys	0x1
15:4	Reserved			
16	reg_sd1_pwr_ack	R0	power fence power status 1: power on, 0: power off [0]: sd1	
17	reg_sd1_pwr_ack_2nd	R0	power fence power status 1: power on, 0: power off [0]: sd1	
18	reg_mcu_pwr_ack	R0	power fence power status 1: power on, 0: power off [1]: mcu subsys	
19	reg_mcu_pwr_ack_2nd	R0	power fence power status 1: power on, 0: power off [1]: mcu subsys	
31:20	Reserved			

6.6.3.29 rtc_ip_iso_ctrl

Table 6.100: rtc_ip_iso_ctrl, Offset Address: 0x084

Bits	Name	Access	Description	Reset
0	reg_sd1_iso_en	R/W	sd1 iso enablel 1: iso enable, 0: iso disable	0x0
1	reg_mcu_iso_en	R/W	mcu iso enablel 1: iso enable, 0: iso disable	0x0
15:2	Reserved			
17:16	reg_ip_por_en	R/W	1: pwr_island reset assert when power ack is 0	0x3
31:18	Reserved			

6.6.3.30 rtcsys_wkup_ctrl1

Table 6.101: rtcsys_wkup_ctrl1, Offset Address: 0x094

Bits	Name	Access	Description	Reset
7:0	reg_rtcsys_wkint_final_status	RO	wkint final status [0]: sd1_wakeup_intr [1]: gpio_int [2]: timer0_int [3]: timer1_int [4]: saradc_int [5]: rtcsys_ictrl_int [6]: NA [7]: NA	
31:8	Reserved			

6.6.3.31 rtcsys_sram_ctrl

Table 6.102: rtcsys_sram_ctrl, Offset Address: 0x098

Bits	Name	Access	Description	Reset
0	reg_ahb_sram_slp	R/W	1 : ahb sram into sleep mode	0x0
1	reg_ahb_sram_sd	R/W	1 : ahb sram into shut down mode	0x0
2	reg_ahb_sram_ctrl_ov	R/W	0 : ahb sram ctrl by PMU FSM and ahb sram busy 1: sram ctrol by register reg_ahb_sr am_slp/reg_ahb_sram_sd	0x1
3	reg_sdio_sram_slp	R/W	1 : sdio sram into sleep mode	0x0
4	reg_sdio_sram_sd	R/W	1 : sdio sram into shut down mode	0x0
5	reg_sdio_sram_ctrl_ov	R/W	0 : sram's sd pin = 1'b0 1: sram ctrol by register reg_sdio_sram_sd	0x1
6	reg_mcu_sram_slp	R/W	1 : mcu iram sram into sleep mode	0x0
7	reg_mcu_sram_sd	R/W	1 : mcu iram sram into shut down mode	0x0
8	reg_mcu_sram_ctrl_ov	R/W	0 : mcu iram sram ctrl by PMU FSM 1: sram ctrol by register reg_ahb_sr am_slp/reg_ahb_sram_sd	0x1
9	reg_rtc_sram_slp	R/W	1 : mcu iram sram into sleep mode	0x0
10	reg_rtc_sram_sd	R/W	1 : mcu iram sram into shut down mode	0x0
11	reg_rtc_sram_ctrl_ov	R/W	0 : mcu iram sram ctrl by PMU FSM 1: sram ctrol by register reg_ahb_sr am_slp/reg_ahb_sram_sd	0x1
2 7:12	Reserved			
28	reg_mcu_sram_force_ce	R/W	1: force mcu_iram cs = 1	0x1
3 1:29	Reserved			

6.6.3.32 rtcsys_io_ctrl

Table 6.103: rtcsys_io_ctrl, Offset Address: 0x09c

Bits	Name	Access	Description	Reset
0	reg_i2c_mux_opt0	R/W	0: pwr_gpio6/8 control by dw_gpio 1: pwr_gpio6 is PWR_IIC_SDA pwr_gpio8 is PWR_IIC_SCL	0x0
31:1	Reserved			

6.6.3.33 rtcsys_wdt_ctrl

Table 6.104: rtcsys_wdt_ctrl, Offset Address: 0x0a0

Bits	Name	Access	Description	Reset
0	reg_rtc_hw_wdg_rst_en	R/W	0: disable rtc wdt trigger warm reset or pwrcyc reset 1: enable rtc wdt trigger warm reset or pwrcyc reset	0x0
1	reg_rtc_wdt_ctrl_mask_en	R/W	no load	0x1
31:2	Reserved			

6.6.3.34 rtcsys_irrx_clk_ctrl

Table 6.105: rtcsys_irrx_clk_ctrl, Offset Address: 0x0a4

Bits	Name	Access	Description	Reset
0	reg_irrx_clk_sw_force_on	R/W	force on clk ctrl of irrx	0x1
1	reg_irrx_xtal_req_en	R/W	enable irrx clk ctrl requet XTAL	0x0
2	reg_irrx_osc_req_en	R/W	enable irrx clk ctrl requet OSC	0x0
3	reg_irrx_ldo_req_en	R/W	enable irrx clk ctrl requet LDO	0x0
7:4	Reserved			
15:8	reg_irrx_xtal_filter_cyc	R/W	irrx xtal filter cycle (default 2ms)	0x40
1 9:16	reg_irrx_clk_ctrl_st	RO	irrx clock ctrol state	
3 1:20	Reserved			

6.6.3.35 rtcsys_rtc_wkup_ctrl

Table 6.106: rtcsys_rtc_wkup_ctrl, Offset Address: 0x0a8

Bits	Name	Access	Description	Reset
7:0	reg_rtc_wkint_mask	R/W	wakeup source mask int to RTC_CORE [0]: irrrx_intr [1]: gpio_int [2]: timer0_int [3]: timer1_int [4]: saradc_int [5]: rtcsys_ictrl_int [6]: wdt_int [7]: irrx_wakeup	0xff
15:8	Reserved			
23:16	reg_rtc_puint_mask	R/W	power-up source mask int to RTC_CORE [0]: irrrx_intr [1]: gpio_int [2]: timer0_int [3]: timer1_int [4]: saradc_int [5]: rtcsys_ictrl_int [6]: wdt_int [7]: irrx_wakeup	0xff
31:24	Reserved			

6.6.3.36 rtcsys_por_rst_ctrl

Table 6.107: rtcsys_por_rst_ctrl, Offset Address: 0x0ac

Bits	Name	Access	Description	Reset
0	reg_rtcsys_reset_en	R/W	0: not allow rtcsys reset by pwr cyc/ wdt warm reset 1 : allow rtcsys reset by pwr cyc/ wdt warm reset	0x0
1	reg_rtcsys_rstn_src_sel	R/W	select rtcsys rstn src 0: rtc_core fsm (reset with die domain) 1: por_pwr_rstn	0x0
31:2	Reserved			

CHAPTER

SEVEN

RESET

7.1 Overview

The reset management module uniformly manages the reset and timing of the entire chip, subsystem, and functional modules.

7.2 Reset Control

There are three levels of reset management modules inside the chip to manage the reset of the entire chip, subsystems, and various functional modules.

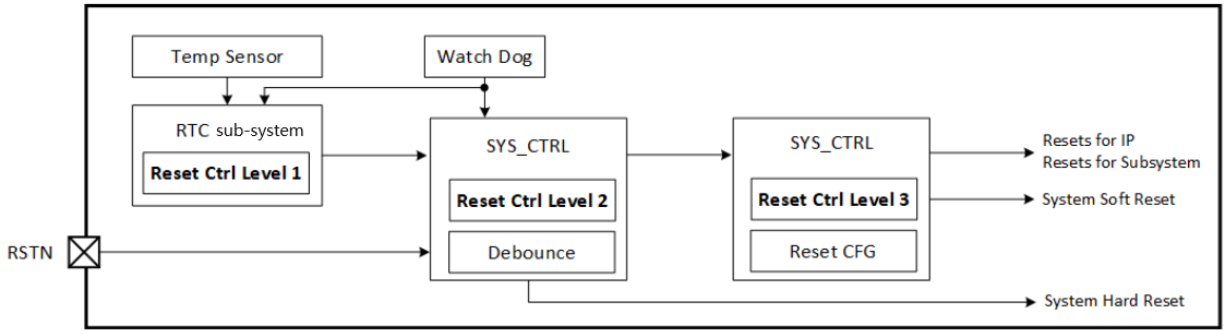


Diagram 7.1: Reset management module block diagram

The Reset Ctrl Level 1 circuit is responsible for the system power-on reset function. The power-on reset (POR) is generated by the real-time clock module in conjunction with the global power management and crystal timing. For details, refer to the section *Real Time Clock*. Level 1 Reset can be triggered in the following ways:

- Power on reset
- Overheat protection reset
- Watchdog reset: When RCT_CTRL0.hw_wdg_rst_en is 1 (see *RTC_CTRL0, Offset Address: 0x008*) and bit[0] of sys_ctrl_reg.reg_sw_root_reset_en is 0 (see *sys_ctrl_reg, Offset Address: 0x008*), the watchdog timer times out and triggers a system reset.

The Reset Ctrl Level 2 circuit is responsible for generating a System Hard Reset, which performs a hard reset on the chip globally including subsystems and functional modules. Level 2 Reset can be triggered in the following ways:

- Watchdog reset: When bit[0] of sys_ctrl_reg.reg_sw_root_reset_en is 1 (see *sys_ctrl_reg, Offset Address: 0x008*), the watchdog timer times out and triggers a system reset.
- External reset pin (RSTN), which has beuilt-in debounce circuit, RSTN high and low level effective signals must reach 6.56ms.

The Reset Ctrl Level 3 circuit is responsible for providing the reset configuration register (Reset CRG) corresponding to the soft reset control. For details, refer to *Reset configuration register*. It includes:

- System soft reset: Reset the entire chip, except for a few circuits and RTC internal circuits.
- Processor subsystem reset: Resets the processor and processor subsystem. Programming of register SOFT_CPUAC_RSTN can soft reset the processor and subsystem. After writing 0 to the configuration register, the reset controller will wait for a 24us delay before triggering the corresponding processor reset. During this period, the processor should end access to the bus to avoid the bus hanging after reset. After triggering the reset, the corresponding reset signal will last for 8us and then be automatically released. The processor and processor subsystem will complete the reset and start booting.
- Functional subsystem and modules reset: Reset each functional subsystem and functional modules. Programming of register SOFT_RSTN_0 ~ 3 can soft reset each functional module. The reset configuration is active low, and the reset signal will not be cleared automatically. Therefore, after the software configures the corresponding register to 0 to trigger the reset, it also needs to be configured to 1 to release the reset. Before reset-

ting, make sure that the built-in DMA of each functional module and functional subsystem to the bus and the processor to the module are idle. Otherwise, the reset will fail and the system may hang.

7.3 Reset configuration register

7.3.1 Reset configuration register overview

Base address 0x03003000

Table 7.1: Overview of Reset Configuration Registers

Name	Address Offset	Description
SOFT_RSTN_0	0x000	soft-reset ctrl register 0
SOFT_RSTN_1	0x004	soft-reset ctrl register 1
SOFT_RSTN_2	0x008	soft-reset ctrl register 2
SOFT_RSTN_3	0x00c	soft-reset ctrl register 3
SOFT_CPUAC_RSTN	0x020	CPU auto clear soft-reset ctrl register
SOFT_CPU_RSTN	0x024	CPU soft-reset ctrl register

7.3.2 Reset configuration register description

7.3.2.1 SOFT_RSTN_0

Table 7.2: SOFT_RSTN_0, Offset Address: 0x000

Bits	Name	Access	Description	Reset
1:0	Reserved			
2	reg_soft_reset_x_ddr	R/W	DDR system software reset (active low)	0x1
3	reg _soft_reset_x_h264c	R/W	H264 IP software reset (active low)	0x1
4	reg _soft_reset_x_jpeg	R/W	JPEG IP software reset (active low)	0x1
5	reg _soft_reset_x_h265c	R/W	H265 IP software reset (active low)	0x1
6	reg _soft_reset_x_vipsys	R/W	VIP system software reset (active low)	0x1
7	reg _soft_reset_x_tdma	R/W	TPU_DMA IP software reset (active low)	0x1
8	reg_soft_reset_x_tpu	R/W	TPU IP software reset (active low)	0x1
9	reg _soft_reset_x_tpusys	R/W	TPU system software reset (active low)	0x1
10	Reserved			
11	reg_soft_reset_x_usb	R/W	USB IP software reset (active low)	0x1
12	reg _soft_reset_x_eth0	R/W	ETH0 IP software reset (active low)	0x1
13	reg _soft_reset_x_eth1	R/W	ETH1 IP software reset (active low)	0x1
14	reg _soft_reset_x_nand	R/W	NAND IP software reset (active low)	0x1
15	reg _soft_reset_x_emmc	R/W	EMMC IP software reset (active low)	0x1
16	reg_soft_reset_x_sd0	R/W	SD0 IP software reset (active low)	0x1
17	Reserved			
18	reg _soft_reset_x_sdma	R/W	SDMA IP software reset (active low)	0x1
19	reg _soft_reset_x_i2s0	R/W	I2S0 IP software reset (active low)	0x1

continues on next page

Table 7.2 – continued from previous page

Bits	Name	Access	Description	Reset
20	reg _soft_reset_x_i2s1	R/W	I2S1 IP software reset (active low)	0x1
21	reg _soft_reset_x_i2s2	R/W	I2S2 IP software reset (active low)	0x1
22	reg _soft_reset_x_i2s3	R/W	I2S3 IP software reset (active low)	0x1
23	reg _soft_reset_x_uart0	R/W	UART0 IP software reset (active low)	0x1
24	reg _soft_reset_x_uart1	R/W	UART1 IP software reset (active low)	0x1
25	reg _soft_reset_x_uart2	R/W	UART2 IP software reset (active low)	0x1
26	reg _soft_reset_x_uart3	R/W	UART3 IP software reset (active low)	0x1
27	reg _soft_reset_x_i2c0	R/W	I2C0 IP software reset (active low)	0x1
28	reg _soft_reset_x_i2c1	R/W	I2C1 IP software reset (active low)	0x1
29	reg _soft_reset_x_i2c2	R/W	I2C2 IP software reset (active low)	0x1
30	reg _soft_reset_x_i2c3	R/W	I2C3 IP software reset (active low)	0x1
31	reg _soft_reset_x_i2c4	R/W	I2C4 IP software reset (active low)	0x1

7.3.2.2 SOFT_RSTN_1

Table 7.3: SOFT_RSTN_1, Offset Address: 0x004

Bits	Name	Access	Description	Reset
0	reg _soft_reset_x_pwm0	R/W	PWM0 IP software reset (active low)	0x1
1	reg _soft_reset_x_pwm1	R/W	PWM1 IP software reset (active low)	0x1
2	reg _soft_reset_x_pwm2	R/W	PWM2 IP software reset (active low)	0x1
3	reg _soft_reset_x_pwm3	R/W	PWM3 IP software reset (active low)	0x1
7:4	Reserved			
8	reg _soft_reset_x_spi0	R/W	SPI0 IP software reset (active low)	0x1
9	reg _soft_reset_x_spi1	R/W	SPI1 IP software reset (active low)	0x1
10	reg _soft_reset_x_spi2	R/W	SPI2 IP software reset (active low)	0x1
11	reg _soft_reset_x_spi3	R/W	SPI3 IP software reset (active low)	0x1
12	reg _soft_reset_x_gpio0	R/W	GPIO0 IP software reset (active low)	0x1
13	reg _soft_reset_x_gpio1	R/W	GPIO1 IP software reset (active low)	0x1
14	reg _soft_reset_x_gpio2	R/W	GPIO2 IP software reset (active low)	0x1
15	reg _soft_reset_x_efuse	R/W	EFUSE IP software reset (active low)	0x1
16	reg_soft_reset_x_wdt	R/W	WDT0 IP software reset (active low)	0x1
17	reg _soft_reset_x_ahb_rom	R/W	ROM IP software reset (active low)	0x1
18	reg _soft_reset_x_spic	R/W	SPIC IP software reset (active low)	0x1
19	reg _soft_reset_x_tempsen	R/W	TEMPSEN IP software reset (active low)	0x1
20	reg _soft_reset_x_saradc	R/W	SARADC IP software reset (active low)	0x1
2 5:21	Reserved			
26	reg_soft _reset_x_combo_phy0	R/W	USB_PHY IP software reset (active low)	0x1
2 8:27	Reserved			
29	reg_soft _reset_x_spi_nand	R/W	NAND IP software reset (active low)	0x1
30	reg_soft_reset_x_se	R/W	SE IP software reset (active low)	0x1
31	Reserved			

7.3.2.3 SOFT_RSTN_2

Table 7.4: SOFT_RSTN_2, Offset Address: 0x008

Bits	Name	Access	Description	Reset
9:0	Reserved			
10	reg _soft_reset_x_uart4	R/W	UART4 IP software reset (active low)	0x1
11	reg _soft_reset_x_gpio3	R/W	GPIO3 IP software reset (active low)	0x1
12	reg _soft_reset_x_system	R/W	SYSTEM software reset (active low)	0x1
13	reg _soft_reset_x_timer	R/W	TIMER IP software reset (active low)	0x1
14	reg _soft_reset_x_timer0	R/W	TIMER0 IP software reset (active low)	0x1
15	reg _soft_reset_x_timer1	R/W	TIMER1 IP software reset (active low)	0x1
16	reg _soft_reset_x_timer2	R/W	TIMER2 IP software reset (active low)	0x1
17	reg _soft_reset_x_timer3	R/W	TIMER3 IP software reset (active low)	0x1
18	reg _soft_reset_x_timer4	R/W	TIMER4 IP software reset (active low)	0x1
19	reg _soft_reset_x_timer5	R/W	TIMER5 IP software reset (active low)	0x1
20	reg _soft_reset_x_timer6	R/W	TIMER6 IP software reset (active low)	0x1
21	reg _soft_reset_x_timer7	R/W	TIMER7 IP software reset (active low)	0x1
22	reg _soft_reset_x_wgn0	R/W	WGN0 IP software reset (active low)	0x1
23	reg _soft_reset_x_wgn1	R/W	WGN1 IP software reset (active low)	0x1
24	reg _soft_reset_x_wgn2	R/W	WGN2 IP software reset (active low)	0x1
25	reg _soft_reset_x_keyscan	R/W	KEYSCAN IP software reset (active low)	0x1
26	Reserved			
27	reg _soft_reset_x_auddac	R/W	AUDDAC IP software reset (active low)	0x1
28	reg_soft _reset_x_auddac_apb	R/W	AUDDAC APB software reset (active low)	0x1
29	reg _soft_reset_x_audadc	R/W	AUDADC IP software reset (active low)	0x1
30	Reserved			
31	reg _soft_reset_x_vcsys	R/W	VCSYS SYS software reset (active low)	0x1

7.3.2.4 SOFT_RSTN_3

Table 7.5: SOFT_RSTN_3, Offset Address: 0x00c

Bits	Name	Access	Description	Reset
0	reg _soft_reset_x_ethphy	R/W	ETHPHY IP software reset (active low)	0x1
1	reg_soft _reset_x_ethphy_apb	R/W	ETHPHY APB REG software reset (active low)	0x1
2	reg _soft_reset_x_audsrc	R/W	AUDSRC IP software reset (active low)	0x1
3	reg_soft _reset_x_vip_cam0	R/W	VIP CAM0 IP software reset (active low)	0x1
4	reg _soft_reset_x_wdt1	R/W	WDT1 IP software reset (active low)	0x1
5	reg _soft_reset_x_wdt2	R/W	WDT2 IP software reset (active low)	0x1
31:6	Reserved			

7.3.2.5 SOFT_CPUAC_RSTN

Write Lock: SOFT_CPUAC_RSTN_wr_lock

Table 7.6: SOFT_CPUAC_RSTN, Offset Address: 0x020

Bits	Name	Access	Description	Reset
0	reg_auto_clear _reset_x_cpucore0	R/W	CPUCORE0 auto_clear_reset (active low)	0x1
1	reg_auto_clear _reset_x_cpucore1	R/W	CPUCORE1 auto_clear_reset (active low)	0x1
2	reg_auto_clear _reset_x_cpucore2	R/W	CPUCORE2 auto_clear_reset (active low)	0x1
3	reg_auto_clear _reset_x_cpucore3	R/W	CPUCORE3 auto_clear_reset (active low)	0x1
4	reg_auto_clear _reset_x_cpusys0	R/W	CPUSYS0 auto_clear_reset (active low)	0x1
5	reg_auto_clear _reset_x_cpusys1	R/W	CPUSYS1 auto_clear_reset (active low)	0x1
6	reg_auto_clear _reset_x_cpusys2	R/W	CPUSYS2 auto_clear_reset (active low)	0x1
31:7	Reserved			

7.3.2.6 SOFT_CPU_RSTN

Table 7.7: SOFT_CPU_RSTN, Offset Address: 0x024

Bits	Name	Access	Description	Reset
0	reg_soft _reset_x_cpucore0	R/W	CPUCORE0 soft reset (active low)	0x0
1	reg_soft _reset_x_cpucore1	R/W	CPUCORE1 soft reset (active low)	0x0
2	reg_soft _reset_x_cpucore2	R/W	CPUCORE2 soft reset (active low)	0x0
3	reg_soft _reset_x_cpucore3	R/W	CPUCORE3 soft reset (active low)	0x0
4	reg_soft _reset_x_cpusys0	R/W	CPUSYS0 soft reset (active low)	0x0
5	reg _soft_reset_x_cpusys1	R/W	CPUSYS1 soft reset (active low)	0x0
6	reg _soft_reset_x_cpusys2	R/W	CPUSYS2 soft reset (active low)	0x0
31:7	Reserved			

CHAPTER

EIGHT

CLOCK

8.1 Overview

The clock management module manages the chip clock, including:

- Management and control of clock inputs
- PLL clock source and related frequency multiplication and frequency division control
- Clock division and control
- Generate the working clock of each module
- Management and control of clock outputs

8.2 Functional block diagram

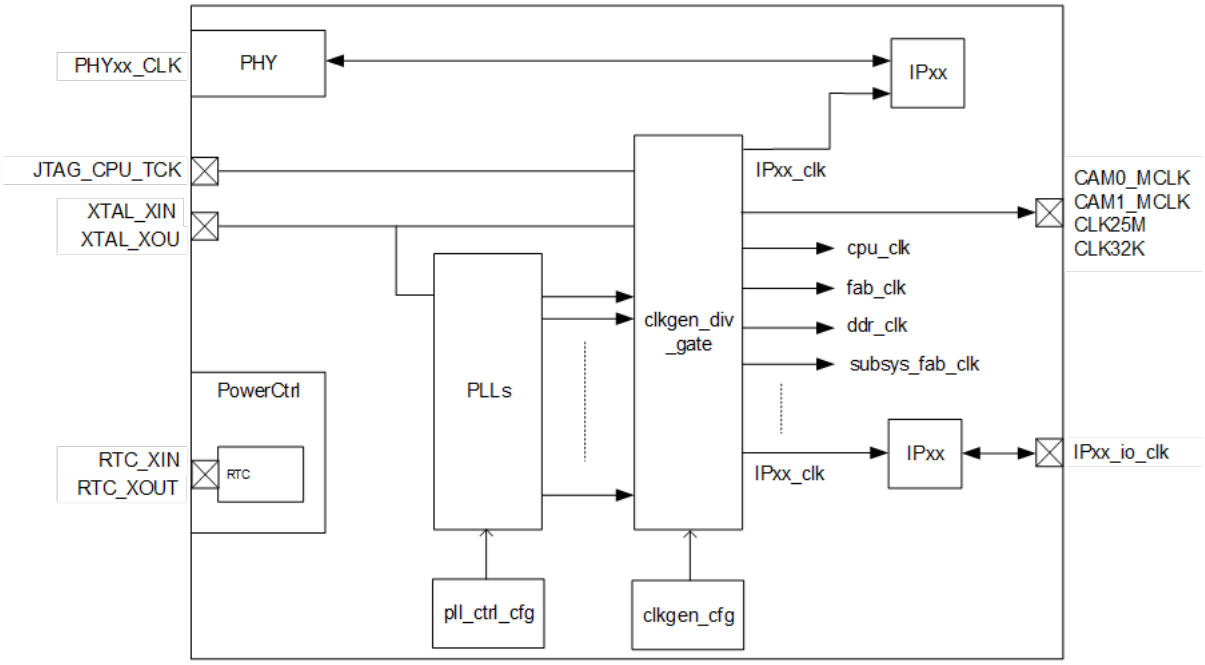


Diagram 8.1: Clock management block diagram

XTAL_XIN is the PLL input clock, fixedly connected to the 25MHz crystal; RTC_XIN is the RTC input clock, fixedly connected to the 32.768KHz crystal.

8.3 Clock resources and frequency division structure

The clock of the entire system is mainly divided into two parts: PLL clocks and IP/Subsystem clocks.

The system clock source mainly comes from external XTAL. XTAL has its frequency multiplied by a PLL. As shown in *Clock source frequency division diagram*, for each IP, XTAL is generally used as the initial clock source. After being multiplied by one or more PLL clocks, each is processed by a frequency division circuit and then used as clcok input for an IP or subsystem.

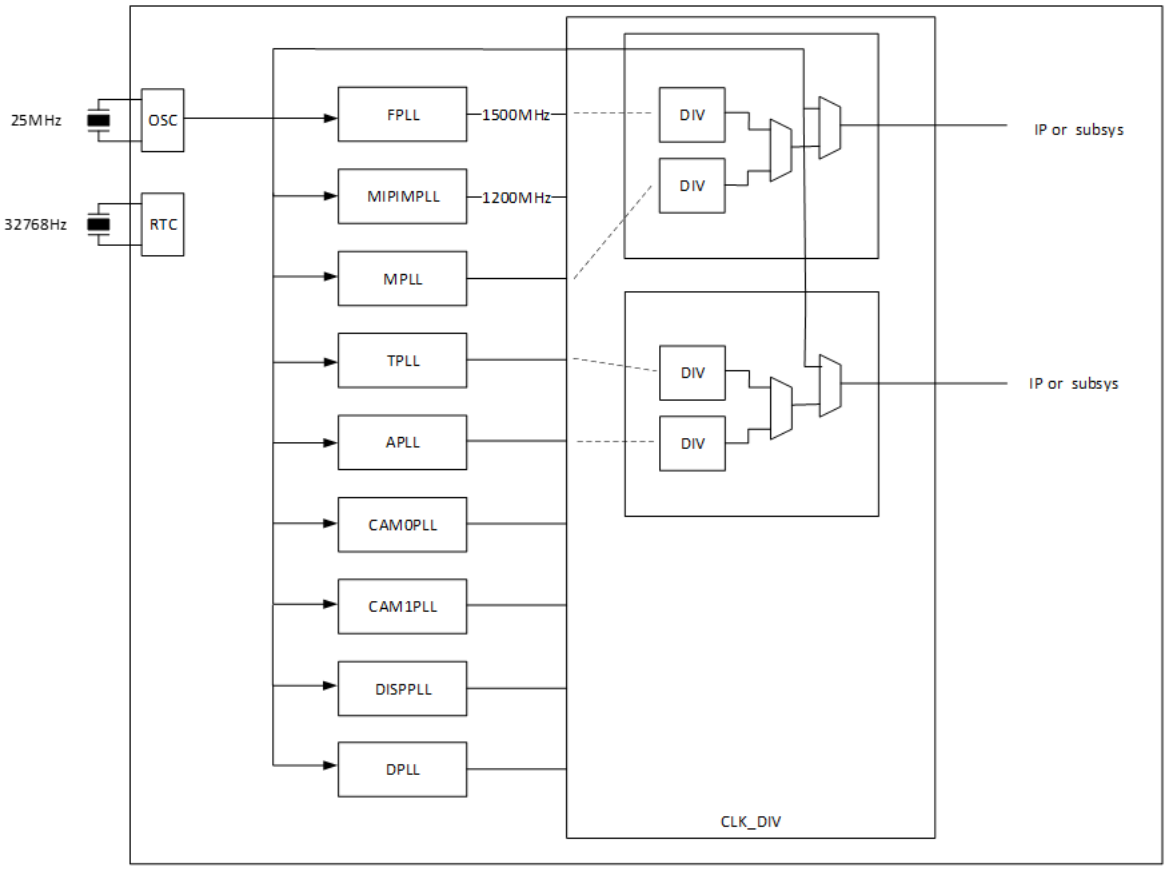


Diagram 8.2: Clock source frequency division diagram

8.4 PLL configuration

See *PLL Configuration Parameters*, the chip has 8 built-in PLLs (not counting Analog IP built-in PLL), which are divided into two types of PLL: integer frequency multiplication and fractional frequency multiplication.

For more information on PLL Configuration/Control Registers, refer to *PLL Control Register Overview* and *PLL Control Register Description*.

Table 8.1: PLL Configuration Parameters

PLL	Configuration Register	Power-down Control Register	Preset Frequency	PLL Type
FPLL	fpll_csr	fpll_pwd (default On)	1500MHz	Integer frequency multiplier
MIPIMPLL	mipimpll_csr	mipimpll_pwd (default On)	900MHz	Integer frequency multiplier
MPLL	mpll_csr mpll_ssc_syn_ctrl mpll_ssc_syn_set mpll_ssc_syn_span mpll_ssc_syn_setp	mpll_pwd (default On)	1000MHz	Integer/fractional frequency multiplier
TPLL	tpll_csr tpll_ssc_syn_ctrl tpll_ssc_syn_set tpll_ssc_syn_span tpll_ssc_syn_setp	tpll_pwd (default On)	1400MHz	Integer/fractional frequency multiplier
APLL	apll_csr apll_ssc_syn_ctrl apll_ssc_syn_set apll_ssc_syn_span apll_ssc_syn_setp	apll_pwd(default On)	1050MHz	Integer/fractional frequency multiplier
CAM0PLL	cam0pll_csr cam0pll_ssc_syn_ctrl cam0pll_ssc_syn_set cam0pll_ssc_syn_span cam0pll_ssc_syn_setp	cam0pll_pwd (default On)	1050MHz	Integer/fractional frequency multiplier
CAM1PLL	cam1pll_csr cam1pll_ssc_syn_ctrl cam1pll_ssc_syn_set cam1pll_ssc_syn_span cam1pll_ssc_syn_setp	cam1pll_pwd (default On)	1025MHz	Integer/fractional frequency multiplier
DISPPLL	dispplll_csr dispplll_ssc_syn_ctrl dispplll_ssc_syn_set dispplll_ssc_syn_span dispplll_ssc_syn_setp	dispplll_pwd (default On)	1200MHz	Integer/fractional frequency multiplier

8.4.1 Integer Multiplier PLL

Tuning the integer PLL flow:

1. When adjusting, you must turn off the clock related to this PLL or select the clock source as XTAL or other PLL.
2. Configure the *_pll_csr register according to the integer PLL parameter table
3. Clear *_pll_pwd

Table 8.2: Integer PLL configuration parameter table

PLL Parameters	Range	Precautions
PLL_REF	25MHz~2500MHz	
PLL_VCO	800MHz~2500MHz	
Pre_div_sel	1~127	PLL_VCO = PLL_REF*Div_sel/Pre_div_sel PLL_FOUT = PLL_VCO/Post_div_sel
Div_sel	6~127	
Post_div_sel	1~127	
Ictrl	0~7	0.2 < 1.84*(1+Mode)*(1+Ictrl)/2/Div_sel <= 0.35
Mode	0~3	

8.4.2 Fractional Multiplier PLL

Adjust fractional PLL process:

1. When using this PLL, the IP clock must be turned off or XTAL or other stable clock sources must be selected.
2. Configure *_ssc_syn_src_en to enable the synthesizer clock
3. Configure *_ssc_syn_set according to PLL frequency requirements,
4. Toggle *_ssc_syn_up to make the configuration take effect
5. Configure the *_pll_csr register according to the integer PLL parameter table
6. Clear *_pll_pwd

Table 8.3: Fractional PLL parameter configuration table

PLL Parameters	Range	Precautions
ssc_freq_in	DSIPLL: 1.2GHz DDRPLL: 1.5GHz Others 600MHz	Default clock gated, need enable
ssc_syn_set	> 4.x * 2^26	ssc_freq_in*div_sel*2^26/PLL_VCO
PLL_REF	100M~2500MHz	
PLL_VCO	800MHz~2500MHz	
Div_sel	6~127	PLL_VCO = PLL_REF*div_sel FOUT = PLL_VCO/Post_div_sel
Post_div_sel	1~127	
Ictrl	0~7	0.1 < 1.84*(1+Mode)*(1+Ictrl)/2/Div_sel <= 0.24
Mode	0~3	

8.5 IP/Subsystem clock configuration

Clock source and preset frequency division parameter table lists all IP/Subsystem clocks and related information in the system. The following is an introduction to IP/Subsystem clock related information based on this table.

- "CLK_NAME" is the clock name of the IP/Subsystem clock.
- Each IP/Subsystem clock can be switched on and off via registers (Gate function). The specific settings are controlled by the registers clk_en_0, clk_en_1, clk_en_2, clk_en_3 and clk_en_4. Each bit controls a clock. See *clk_en_0, Offset Address: 0x000*, *clk_en_1, Offset Address: 0x004*, *clk_en_2, Offset Address: 0x008*, *clk_en_3, Offset Address: 0x00c* and *clk_en_4, Offset Address: 0x010*.
- The "DIV" column identifies whether the clock supports frequency division (Divide function), "Y" indicates support, and empty indicates not support. Each clock that supports frequency division has a corresponding DIV register for setting the division factor (Divider Factor). For example, clk_tpu corresponds to the [20:16] bit field of div_clk_tpu.
- If a clock supports frequency division, there may be multiple parent clocks (PLL clock or xtal) as clock sources for frequency division. These parent clocks are divided into two groups, DIV_IN0 and DIV_IN1, which correspond to "DIV_IN0" column and the "DIV_IN1" column on the table respectively. Each group contains one or more Sources. We can select the Source through the clk_src bit field of the DIV register corresponding to the clock. Most clocks that support frequency division only need one set of Source, namely DIV_IN0. A small number of clocks have two sets of Sources, including clk_a53, clk_c906_0 and clk_c906_1. If a clock has two sets of Sources, then the clock has two corresponding DIV registers. For example, clk_a53 corresponds to div_clk_a53_0 and div_clk_a53_1. We can select groups DIV_IN0 and DIV_IN1 through register clk_sel_0, the default is DIV_IN1 (Reset). The "PLL SRC/DIV/FREQ" column gives the "Parent Clock"/"Division Factor"/"Frequency Value" selected by default for this clock. The software can switch the clock source from XTAL to PLL after Boot, and adjust the clock frequency division configuration.
- The "XTAL" column identifies whether the clock supports bypassing its parent clock to xtal. "Y" indicates support, and empty indicates not support. The specific settings are controlled by the registers clk_byp_0 and clk_byp_1, see *clk_byp_0, Offset Address: 0x030* and *clk_byp_1, Offset Address: 0x034*.

Table 8.4: Clock source and preset frequency division parameter table

CLK_NAME	XTAL	DIV	PLL SRC /DIV/FREQ	DIV_IN0				DIV_IN1
clk_a53	Y	Y	fpll /(2)/750M	tpll	a0pll	mip-impll	mpll	fpll
clk_cpu_axi0	Y	Y	fpll /(3)/500M	fpll	disp-pll			
clk_cpu_gic	Y	Y	fpll /(5)/300M	fpll				
clk_xtal_a53				xtal				
clk_tpu	Y	Y	fpll /(3)/500M	tpll	a0pll	mip-impll	fpll	
clk_tpu_fab				mipimpll				
clk_ahb_rom				clk_axi4				
clk_ddr_axi_reg				clk_axi6				
clk_rtc_25m				xtal				
clk_tempsen				xtal				
clk_saradc				xtal				
clk_efuse				xtal				
clk_apb_efuse				xtal				
clk_debug				xtal				
clk_xtal_misc				xtal				

continues on next page

Table 8.4 – continued from previous page

CLK_NAME	XTAL	DIV	PLL SRC /DIV/FREQ	DIV_IN0				DIV_IN1
clk_axi4_emmc				clk_axi4				
clk_emmc	Y	Y	fpll/ (15)/100M	fpll	disp-pll			
clk_100k_emmc		Y		clk_1m				
clk_axi4_sd0				clk_axi4				
clk_sd0	Y	Y	fpll/ (15)/100M	fpll	disp-pll			
clk_100k_sd0		Y		clk_1m				
clk_axi4_sd1				clk_axi4				
clk_sd1	Y	Y	fpll/ (15)/100M	fpll	disp-pll			
clk_100k_sd1		Y		clk_1m				
clk_spi_nand	Y	Y	fpll/(8)/187.5M	fpll	disp-pll			
clk_500m_eth0	Y	Y		fpll				
clk_axi4_eth0				clk_axi4				
clk_500m_eth1	Y	Y		fpll				
clk_axi4_eth1				clk_axi4				
clk_apb_gpio				clk_axi6				
clk_apb_gpio_intr				clk_axi6				
clk_gpio_db		Y		clk_1m				
clk_ahb_sf				clk_axi4				
clk_sdma_axi				clk_axi4				
clk_sdma_aud0	Y	Y	a0pll/(18)/58.3M	a0pll	a24k			
clk_sdma_aud1	Y	Y	a0pll/(18)/58.3M	a0pll	a24k			
clk_sdma_aud2	Y	Y	a0pll/(18)/58.3M	a0pll	a24k			
clk_sdma_aud3	Y	Y	a0pll/(18)/58.3M	a0pll	a24k			
clk_apb_i2c				clk_axi4				
clk_apb_wdt				xtal				
clk_pwm				clk_pwm_src				
clk_apb_spi0				clk_axi4				
clk_apb_spi1				clk_axi4				
clk_apb_spi2				clk_axi4				
clk_apb_spi3				clk_axi4				
clk_cam0_200	Y	Y	xtal/(1)/ 25M	xtal	disp-pll			
clk_uart0				clk_cam0_200				
clk_apb_uart0				clk_axi4				
clk_uart1				clk_cam0_200				
clk_apb_uart1				clk_axi4				
clk_uart2				clk_cam0_200				
clk_apb_uart2				clk_axi4				
clk_uart3				clk_cam0_200				
clk_apb_uart3				clk_axi4				
clk_uart4				clk_cam0_200				
clk_apb_uart4				clk_axi4				
clk_apb_i2s0				clk_axi4				
clk_apb_i2s1				clk_axi4				

continues on next page

Table 8.4 – continued from previous page

CLK_NAME	XTAL	DIV	PLL SRC /DIV/FREQ	DIV_IN0				DIV_IN1
clk_apb_i2s2				clk_axi4				
clk_apb_i2s3				clk_axi4				
clk_axi4_usb				clk_axi4				
clk_apb_usb				clk_axi4				
clk_125m_usb	Y	Y		fpll				
clk_33k_usb		Y		clk_1m				
clk_12m_usb	Y	Y		fpll				
clk_axi4	Y	Y	fpll /(5)/300M	fpll	disp-pll			
clk_axi6	Y	Y	fpll/ (15)/100M	fpll				
clk_dsi_esc	Y	Y		clk_axi6				
clk_axi_vip	Y	Y	fpll /(6)/250M	mip-impll	cam0pll	disp pll	fpll	
clk_src_vip_sys_0	Y	Y	fpll /(6)/250M	mip-impll	cam0pll	disp-pll	fpll	
clk_src_vip_sys_1	Y	Y	fpll /(5)/300M	mip-impll	cam0pll	disp-pll	fpll	
clk_axi_video_codec	Y	Y	mipimpll /(2)/450M	a0pll	mip-impll	cam1pll	fpll	
clk_vc_src0	Y	Y	mipimpll /(2)/450M	disp-pll	mip-impll	cam1pll	fpll	
clk_h264c				clk_axi_video_codec				
clk_h265c				clk_axi_video_codec				
clk_jpeg				clk_axi_video_codec				
clk_apb_jpeg				clk_axi6				
clk_apb_h264c				clk_axi6				
clk_apb_h265c				clk_axi6				
clk_cam0				cam0pll				
clk_cam1				cam0pll				
clk_csi_mac0_vip				clk_axi_vip				
clk_csi_mac1_vip				clk_axi_vip				
clk_isp_top_vip				clk_axi_vip				
clk_img_d_vip				clk_axi_vip				
clk_img_v_vip				clk_axi_vip				
clk_sc_top_vip				clk_axi_vip				
clk_sc_d_vip				clk_axi_vip				
clk_sc_v1_vip				clk_axi_vip				
clk_sc_v2_vip				clk_axi_vip				
clk_sc_v3_vip				clk_axi_vip				
clk_dwa_vip				clk_axi_vip				
clk_bt_vip				clk_axi_vip				
clk_disp_vip				clk_axi_vip				
clk_dsi_mac_vip				clk_axi_vip				
clk_lvds0_vip				clk_axi_vip				
clk_lvds1_vip				clk_axi_vip				
clk_csi0_rx_vip				clk_axi_vip				
clk_csi1_rx_vip				clk_axi_vip				
clk_pad_vi_vip				clk_axi_vip				

continues on next page

Table 8.4 – continued from previous page

CLK_NAME	XTAL	DIV	PLL SRC /DIV/FREQ	DIV_IN0				DIV_IN1
clk_1m		Y		xtal_free				
clk_spi	Y	Y	fpll/(8)/187.5M	fpll				
clk_i2c	Y	Y	clk_axi6 /(1)/100M	clk_axi6				
clk_pm				clk_axi6				
clk_timer0				clk_xtal_misc				
clk_timer1				clk_xtal_misc				
clk_timer2				clk_xtal_misc				
clk_timer3				clk_xtal_misc				
clk_timer4				clk_xtal_misc				
clk_timer5				clk_xtal_misc				
clk_timer6				clk_xtal_misc				
clk_timer7				clk_xtal_misc				
clk_apb_i2c0				clk_axi4				
clk_apb_i2c1				clk_axi4				
clk_apb_i2c2				clk_axi4				
clk_apb_i2c3				clk_axi4				
clk_apb_i2c4				clk_axi4				
clk_wgn				xtal				
clk_wgn0				xtal				
clk_wgn1				xtal				
clk_wgn2				xtal				
clk_keyscan				xtal				
clk_ahb_sf1				clk_axi4				
clk_vc_src1	Y	Y		cam1pll				
clk_src_vip_sys_2	Y	Y	disppll /(2)/600M	mip-impll	cam0pll	disp-pll	fpll	
clk_pad_vi1_vip				clk_axi_vip				
clk_cfg_reg_vip				clk_axi6				
clk_cfg_reg_vc				clk_axi6				
clk_audsrc	Y	Y	a0pll/(18)/58.3M	a0pll	a24k			
clk_apb_audsrc				clk_axi4				
clk_vc_src2	Y	Y		fpll				
clk_pwm_src	Y	Y	fpll/(10)/ 150M	fpll	disp-pll			
clk_ap_debug	Y	Y	fpll /(5)/300M	fpll				
clk_src_rtc_sys_0	Y	Y	fpll /(5)/300M	fpll				
clk_pad_vi2_vip				clk_axi_vip				
clk_csi_be_vip				clk_axi_vip				
clk_vip_ip0				clk_axi_vip				
clk_vip_ip1				clk_axi_vip				
clk_vip_ip2				clk_axi_vip				
clk_vip_ip3				clk_axi_vip				
clk_c906_0	Y	Y	fpll /(2)/750M	tpll	a0pll	mip-impll	mpll	fpll
clk_c906_1	Y	Y	fpll /(3)/500M	tpll	a0pll	disp-pll	mpll	fpll
clk_src_vip_sys_3	Y	Y	mipimpll /(2)/450M	mip-impll	cam0pll	disp-pll	fpll	

continues on next page

Table 8.4 – continued from previous page

CLK_NAME	XTAL	DIV	PLL SRC /DIV/FREQ	DIV_IN0				DIV_IN1
clk_src_vip_sys_4	Y	Y	disppll /(3)/400M	mip-impll	cam0pll	disp-pll	fpll	
clk_ive_vip				clk_axi_vip				
clk_raw_vip				clk_axi_vip				
clk_osdc_vip				clk_axi_vip				
clk_fbc_vip				clk_axi_vip				
clk_cam0_vip				clk_axi_vip				

8.5.1 IP/SYS clock source and clock division configuration

1. Turn off the IP clock; if the clock cannot be turned off, you should first configure it to a stable frequency division clock
    1. CPU frequency conversion: configure clk_sel_0 to choose to switch to SRC1 to avoid too low frequency
    2. IP frequency conversion: configure clk_byp_0/1 to switch the clock to XTAL
2. Configure the clock source and divider configuration to be adjusted
3. The frequency divider register [3] needs to be configured for this clock divider configuration to take effect.
4. Select the clock source to the configured clock divider

8.5.2 MCLK0/MCLK1

1. MCLK0/MCLK1 provides the external sensor reference clock.
2. Configure CAM0PLL, clk_cam0_src_div and clk_cam0_src_div to provide the appropriate MCLK0/MCLK1 output frequency.

8.5.3 Clk_A24M

1. clk_a24m can be used as audio clock under acceptable performance conditions.
2. Output the required audio clock source by configuring apll_frac_div_ctrl, apll_frac_div_m and apll_frac_div_n.
3. The frequency of clk_a24m is 900MHz * N/M/2.

8.6 PLL Control Register Overview

Table 8.5: PLL_G2 base address : 0x03002800

Name	Address Offset	Description
pll_g2_ctrl	0x000	Group2 PLL Ctrl register
pll_g2_status	0x004	Group2 PLL Status register
mipimpll_csr	0x008	MIPIMPLL Ctrl register
apll0_csr	0x00c	APLL0 Ctrl register
disppll_csr	0x010	DISPPLL Ctrl register
cam0pll_csr	0x014	CAM0PLL Ctrl register
cam1pll_csr	0x018	CAM1PLL Ctrl register
pll_g2_ssc_syn_ctrl	0x040	Group2 PLL Synthesizer ctrl register
apll_ssc_syn_ctrl	0x050	APLL synthesizer ctrl register
apll_ssc_syn_set	0x054	APLL synthesizer set register
disppll_ssc_syn_ctrl	0x060	DISPPLL synthesizer ctrl register
disppll_ssc_syn_set	0x064	DISPPLL synthesizer set register
cam0pll_ssc_syn_ctrl	0x070	CAM0PLL synthesizer ctrl register
cam0pll_ssc_syn_set	0x074	CAM0PLL synthesizer set register
cam1pll_ssc_syn_ctrl	0x080	CAM1PLL synthesizer ctrl register
cam1pll_ssc_syn_set	0x084	CAM1PLL synthesizer set register
apll_frac_div_ctrl	0x090	APLL frac divider ctrl register
apll_frac_div_m	0x094	APLL frac divider M parameter
apll_frac_div_n	0x098	APLL frac divider N parameter
mipimpll_clk_csr	0x0a0	MIPIMPLL clock Ctrl register
a0pll_clk_csr	0x0a4	a0pll clock Ctrl register
disppll_clk_csr	0x0a8	disppll clock Ctrl register
cam0pll_clk_csr	0x0ac	cam0pll clock Ctrl register
cam1pll_clk_csr	0x0b0	cam1pll clock Ctrl register
clk_cam0_src_div	0x0c0	clk_cam0_src_div
clk_cam1_src_div	0x0c4	clk_cam1_src_div

Table 8.6: PLL_G6 base address : 0x03002900

Name	Address Offset	Description
pll_g6_ctrl	0x000	Group6 PLL Ctrl register
pll_g6_status	0x004	Group6 PLL Status register
mpll_csr	0x008	MPLL Ctrl register
tpll_csr	0x00c	TPLL Ctrl register
fpll_csr	0x010	FPLL Ctrl register
pll_g6_ssc_syn_ctrl	0x040	Group6 PLL Synthesizer ctrl register
dpll_ssc_syn_ctrl	0x050	dpll synthesizer ctrl register
dpll_ssc_syn_set	0x054	dpll synthesizer set register
dpll_ssc_syn_span	0x058	dpll synthesizer span register
dpll_ssc_syn_step	0x05c	dpll synthesizer step register
mpll_ssc_syn_ctrl	0x060	mpll synthesizer ctrl register
mpll_ssc_syn_set	0x064	mpll synthesizer set register
mpll_ssc_syn_span	0x068	mpll synthesizer span register
mpll_ssc_syn_step	0x06c	mpll synthesizer step register
tpll_ssc_syn_ctrl	0x070	tpll synthesizer ctrl register
tpll_ssc_syn_set	0x074	tpll synthesizer set register
tpll_ssc_syn_span	0x078	tpll synthesizer span register
tpll_ssc_syn_step	0x07c	tpll synthesizer step register

8.7 PLL Control Register Description

8.7.1 PLL_G2 Register Description

8.7.1.1 pll_g2_ctrl

Table 8.7: pll_g2_ctrl, Offset Address: 0x000

Bits	Name	Access	Description	Reset
0	mipimpll_pwd	R/W	pll power down	0x0
3:1	Reserved			
4	apll0_pwd	R/W	pll power down	0x0
7:5	Reserved			
8	disppll_pwd	R/W	pll power down	0x0
11:9	Reserved			
12	cam0pll_pwd	R/W	pll power down	0x0
15:13	Reserved			
16	cam1pll_pwd	R/W	pll power down	0x0
31:17	Reserved			

8.7.1.2 pll_g2_status

Table 8.8: pll_g2_status, Offset Address: 0x004

Bits	Name	Access	Description	Reset
0	updating_mipimpll_val	RO	pll setting update status	
1	updating_apll0_val	RO	pll setting update status	
2	updating_disppll_val	RO	pll setting update status	
3	updating_cam0pll_val	RO	pll setting update status	
4	updating_cam1pll_val	RO	pll setting update status	
15:5	Reserved			
16	mipimpll_lock	RO	pll lock status	
17	apll0_lock	RO	pll lock status	
18	disppll_lock	RO	pll lock status	
19	cam0pll_lock	RO	pll lock status	
20	cam1pll_lock	RO	pll lock status	
31:21	Reserved			

8.7.1.3 mipimpll_csr

Table 8.9: mipimpll_csr, Offset Address: 0x008

Bits	Name	Access	Description	Reset
6:0	mipimpll_pre_div_sel	R/W	pll pre_div_sel setting	0x0
7	Reserved			
14:8	mipimpll_post_div_sel	R/W	pll post_div_sel setting	0x0
16:15	mipimpll_sel_mode	R/W	pll mode setting	0x0
23:17	mipimpll_div_sel	R/W	pll div_sel setting	0x0
26:24	mipimpll_ictrl	R/W	pll ictrl setting	0x0
31:27	Reserved			

8.7.1.4 apll0_csr

Table 8.10: apll0_csr, Offset Address: 0x00c

Bits	Name	Access	Description	Reset
6:0	apll0_pre_div_sel	R/W	pll pre_div_sel setting	0x0
7	Reserved			
14:8	apll0_post_div_sel	R/W	pll post_div_sel setting	0x0
16:15	apll0_sel_mode	R/W	pll mode setting	0x0
23:17	apll0_div_sel	R/W	pll div_sel setting	0x0
26:24	apll0_ictrl	R/W	pll ictrl setting	0x0
31:27	Reserved			

8.7.1.5 disppll_csr

Table 8.11: disppll_csr, Offset Address: 0x010

Bits	Name	Access	Description	Reset
6:0	disppll_pre_div_sel	R/W	pll pre_div_sel setting	0x0
7	Reserved			
14:8	disppll_post_div_sel	R/W	pll post_div_sel setting	0x0
16:15	disppll_sel_mode	R/W	pll mode setting	0x0
23:17	disppll_div_sel	R/W	pll div_sel setting	0x0
26:24	disppll_ictrl	R/W	pll ictrl setting	0x0
31:27	Reserved			

8.7.1.6 cam0pll_csr

Table 8.12: cam0pll_csr, Offset Address: 0x014

Bits	Name	Access	Description	Reset
6:0	cam0pll_pre_div_sel	R/W	pll pre_div_sel setting	0x0
7	Reserved			
14:8	cam0pll_post_div_sel	R/W	pll post_div_sel setting	0x0
16:15	cam0pll_sel_mode	R/W	pll mode setting	0x0
23:17	cam0pll_div_sel	R/W	pll div_sel setting	0x0
26:24	cam0pll_ictrl	R/W	pll ictrl setting	0x0
31:27	Reserved			

8.7.1.7 cam1pll_csr

Table 8.13: cam1pll_csr, Offset Address: 0x018

Bits	Name	Access	Description	Reset
6:0	cam1pll_pre_div_sel	R/W	pll pre_div_sel setting	0x0
7	Reserved			
14:8	cam1pll_post_div_sel	R/W	pll post_div_sel setting	0x0
16:15	cam1pll_sel_mode	R/W	pll mode setting	0x0
23:17	cam1pll_div_sel	R/W	pll div_sel setting	0x0
26:24	cam1pll_ictrl	R/W	pll ictrl setting	0x0
31:27	Reserved			

8.7.1.8 pll_g2_ssc_syn_ctrl

Table 8.14: pll_g2_ssc_syn_ctrl, Offset Address: 0x040

Bits	Name	Access	Description	Reset
0	reg_mipimpll_sel_syn_clk	R/W	mipimpll gen synthesizer clock source 0:450M 1:900M	0x1
1	reg_dsi_ssc_syn_src_en	R/W	pll synthesizer clock enable	0x0
2	reg_apll_ssc_syn_src_en	R/W	pll synthesizer clock enable	0x0
3	reg_disppll_ssc_syn_src_en	R/W	pll synthesizer clock enable	0x0
4	reg_cam0pll_ssc_syn_src_en	R/W	pll synthesizer clock enable	0x0
5	reg_cam1pll_ssc_syn_src_en	R/W	pll synthesizer clock enable	0x0
31:6	Reserved			

8.7.1.9 apll_ssc_syn_ctrl

Table 8.15: apll_ssc_syn_ctrl, Offset Address: 0x050

Bits	Name	Access	Description	Reset
0	reg_apll_ssc_syn_sw_up	W1T	pll synthesizer software update	
5:1	Reserved			
6	reg_apll_ssc_syn_fix_div	R/W		0x0
31:7	Reserved			

8.7.1.10 apll_ssc_syn_set

Table 8.16: apll_ssc_syn_set, Offset Address: 0x054

Bits	Name	Access	Description	Reset
31:0	reg_apll_ssc_syn_set	R/W	pll synthesizer fraction setting: [31:26] integer 6 bits [26:0] decimal 26 bits	0x0

8.7.1.11 disppll_ssc_syn_ctrl

Table 8.17: disppll_ssc_syn_ctrl, Offset Address: 0x060

Bits	Name	Access	Description	Reset
0	reg_disppll_ssc_syn_sw_up	W1T	pll synthesizer software update	
5:1	Reserved			
6	reg_disppll_ssc_syn_fix_div	R/W		0x0
31:7	Reserved			

8.7.1.12 disppll_ssc_syn_set

Table 8.18: disppll_ssc_syn_set, Offset Address: 0x064

Bits	Name	Access	Description	Reset
31:0	reg_disppll_ssc_syn_set	R/W	pll synthesizer fraction setting: [31:26] integer 6 bits [26:0] decimal 26 bits	0x0

8.7.1.13 cam0pll_ssc_syn_ctrl

Table 8.19: cam0pll_ssc_syn_ctrl, Offset Address: 0x070

Bits	Name	Access	Description	Reset
0	reg_cam0pll_ssc_syn_sw_up	W1T	pll synthesizer software update	
5:1	Reserved			
6	reg_cam0pll_ssc_syn_fix_div	R/W		0x0
31:7	Reserved			

8.7.1.14 cam0pll_ssc_syn_set

Table 8.20: cam0pll_ssc_syn_set, Offset Address: 0x074

Bits	Name	Access	Description	Reset
31:0	reg_cam0pll_ssc_syn_set	R/W	pll synthesizer fraction setting: [31:26] integer 6 bits [26:0] decimal 26 bits	0x0

8.7.1.15 cam1pll_ssc_syn_ctrl

Table 8.21: cam1pll_ssc_syn_ctrl, Offset Address: 0x080

Bits	Name	Access	Description	Reset
0	reg_cam1pll_ssc_syn_sw_up	W1T	pll synthesizer software update	
5:1	Reserved			
6	reg_cam1pll_ssc_syn_fix_div	R/W		0x0
31:7	Reserved			

8.7.1.16 cam1pll_ssc_syn_set

Table 8.22: cam1pll_ssc_syn_set, Offset Address: 0x084

Bits	Name	Access	Description	Reset
31:0	reg_cam1pll_ssc_syn_set	R/W	pll synthesizer fraction setting: [31:26] integer 6 bits [26:0] decimal 26 bits	0x0

8.7.1.17 apll_frac_div_ctrl

Table 8.23: apll_frac_div_ctrl, Offset Address: 0x090

Bits	Name	Access	Description	Reset
0	reg_apll_frac_div_clk_en	R/W	a24m clock src enable	0x0
1	reg_apll_frac_div_en	R/W	a24m clock div enable	0x0
2	reg_apll_frac_div_up	W1T		
3	reg_apll_frac_reg_out_en	R/W	a24m clock output enable	0x0
31:4	Reserved			

8.7.1.18 apll_frac_div_m

Table 8.24: apll_frac_div_m, Offset Address: 0x094

Bits	Name	Access	Description	Reset
21:0	reg_apll_frac_div_m	R/W	a24m clock freq is 900*N/M/2 (MHz)	0x0
31:22	Reserved			

8.7.1.19 apll_frac_div_n

Table 8.25: apll_frac_div_n, Offset Address: 0x098

Bits	Name	Access	Description	Reset
21:0	reg_apll_frac_div_n	R/W	a24m clock freq is 900*N/M/2 (MHz)	0x0
31:22	Reserved			

8.7.1.20 mipimpll_clk_csr

Table 8.26: mipimpll_clk_csr, Offset Address: 0x0a0

Bits	Name	Access	Description	Reset
0	reg_mipimpll_pdiv_pd	R/W	pd post div	0x0
1	reg_mipimpll_d2_pd	R/W	pd div2 div	0x1
2	reg_mipimpll_d3_pd	R/W	pd div3 div	0x1
3	reg_mipimpll_d5_pd	R/W	pd div5 div	0x1
4	reg_mipimpll_d7_pd	R/W	pd div7 div	0x1
7:5	Reserved			
8	reg_mipimpll_pdiv_auto_pd	R/W	auto pd pdiv clk	0x0
9	reg_mipimpll_d2_auto_pd	R/W	auto pd div2 clk	0x1
10	reg_mipimpll_d3_auto_pd	R/W	auto pd div3 clk	0x1
11	reg_mipimpll_d5_auto_pd	R/W	auto pd div5 clk	0x1
12	reg_mipimpll_d7_auto_pd	R/W	auto pd div7 clk	0x1
31:13	Reserved			

8.7.1.21 a0pll_clk_csr

Table 8.27: a0pll_clk_csr, Offset Address: 0x0a4

Bits	Name	Access	Description	Reset
0	reg_a0pll_pdiv_pd	R/W	pd post div	0x0
1	reg_a0pll_d2_pd	R/W	pd div2 div	0x1
2	reg_a0pll_d3_pd	R/W	pd div3 div	0x1
3	reg_a0pll_d5_pd	R/W	pd div5 div	0x1
4	reg_a0pll_d7_pd	R/W	pd div7 div	0x1
7:5	Reserved			
8	reg_a0pll_pdiv_auto_pd	R/W	auto pd pdiv clk	0x0
9	reg_a0pll_d2_auto_pd	R/W	auto pd div2 clk	0x1
10	reg_a0pll_d3_auto_pd	R/W	auto pd div3 clk	0x1
11	reg_a0pll_d5_auto_pd	R/W	auto pd div5 clk	0x1
12	reg_a0pll_d7_auto_pd	R/W	auto pd div7 clk	0x1
31:13	Reserved			

8.7.1.22 disppll_clk_csr

Table 8.28: disppll_clk_csr, Offset Address: 0x0a8

Bits	Name	Access	Description	Reset
0	reg_disppll_pdiv_pd	R/W	pd post div	0x0
1	reg_disppll_d2_pd	R/W	pd div2 div	0x1
2	reg_disppll_d3_pd	R/W	pd div3 div	0x1
3	reg_disppll_d5_pd	R/W	pd div5 div	0x1
4	reg_disppll_d7_pd	R/W	pd div7 div	0x1
7:5	Reserved			
8	reg_disppll_pdiv_auto_pd	R/W	auto pd pdiv clk	0x0
9	reg_disppll_d2_auto_pd	R/W	auto pd div2 clk	0x1
10	reg_disppll_d3_auto_pd	R/W	auto pd div3 clk	0x1
11	reg_disppll_d5_auto_pd	R/W	auto pd div5 clk	0x1
12	reg_disppll_d7_auto_pd	R/W	auto pd div7 clk	0x1
31:13	Reserved			

8.7.1.23 cam0pll_clk_csr

Table 8.29: cam0pll_clk_csr, Offset Address: 0x0ac

Bits	Name	Access	Description	Reset
0	reg_cam0pll_pdiv_pd	R/W	pd post div	0x0
1	reg_cam0pll_d2_pd	R/W	pd div2 div	0x1
2	reg_cam0pll_d3_pd	R/W	pd div3 div	0x1
3	reg_cam0pll_d5_pd	R/W	pd div5 div	0x1
4	reg_cam0pll_d7_pd	R/W	pd div7 div	0x1
7:5	Reserved			
8	reg_cam0pll_pdiv_auto_pd	R/W	auto pd pdiv clk	0x0
9	reg_cam0pll_d2_auto_pd	R/W	auto pd div2 clk	0x1
10	reg_cam0pll_d3_auto_pd	R/W	auto pd div3 clk	0x1
11	reg_cam0pll_d5_auto_pd	R/W	auto pd div5 clk	0x1
12	reg_cam0pll_d7_auto_pd	R/W	auto pd div7 clk	0x1
31:13	Reserved			

8.7.1.24 cam1pll_clk_csr

Table 8.30: cam1pll_clk_csr, Offset Address: 0x0b0

Bits	Name	Access	Description	Reset
0	reg_cam1pll_pdiv_pd	R/W	pd post div	0x0
1	reg_cam1pll_d2_pd	R/W	pd div2 div	0x1
2	reg_cam1pll_d3_pd	R/W	pd div3 div	0x1
3	reg_cam1pll_d5_pd	R/W	pd div5 div	0x1
4	reg_cam1pll_d7_pd	R/W	pd div7 div	0x1
7:5	Reserved			
8	reg_cam1pll_pdiv_auto_pd	R/W	auto pd pdiv clk	0x0
9	reg_cam1pll_d2_auto_pd	R/W	auto pd div2 clk	0x1
10	reg_cam1pll_d3_auto_pd	R/W	auto pd div3 clk	0x1
11	reg_cam1pll_d5_auto_pd	R/W	auto pd div5 clk	0x1
12	reg_cam1pll_d7_auto_pd	R/W	auto pd div7 clk	0x1
31:13	Reserved			

8.7.1.25 clk_cam0_src_div

Table 8.31: clk_cam0_src_div, Offset Address: 0x0c0

Bits	Name	Access	Description	Reset
0	reg_cam0_div_rstn	R/W	[0] Divider Reset Control 0: Assert Reset	0x1
3:1	Reserved			
4	reg_cam0_div_dis	R/W	[4] Divider Reset Control 0: Assert Reset	0x0
7:5	Reserved			
9:8	reg_cam0_src	R/W	[9:8] Clock source 0: cam0pll 1: cam0pll_d2 2: cam0pll_d3 3: mipimpll_d3	0x0
15:10	Reserved			
21:16	reg_cam0_div	R/W	[21:16] Clock Divider Factor	0x20
31:22	Reserved			

8.7.1.26 clk_cam1_src_div

Table 8.32: clk_cam1_src_div, Offset Address: 0x0c4

Bits	Name	Access	Description	Reset
0	reg_cam1_div_rstn	R/W	[0] Divider Reset Control 0: Assert Reset	0x1
3:1	Reserved			
4	reg_cam1_div_dis	R/W	[4] divider disable	0x0
7:5	Reserved			
9:8	reg_cam1_src	R/W	[9:8] Clock source 0: cam0pll 1: cam0pll_d2 2: cam0pll_d3 3: mipimpll_d3	0x0
15:10	Reserved			
21:16	reg_cam1_div	R/W	[21:16] Clock Divider Factor	0x20
31:22	Reserved			

8.7.2 PLL_G6 Register Description

8.7.2.1 pll_g6_ctrl

Table 8.33: pll_g6_ctrl, Offset Address: 0x000

Bits	Name	Access	Description	Reset
0	mpll_pwd	R/W	pll power down	0x0
3:1	Reserved			
4	tpll_pwd	R/W	pll power down	0x0
7:5	Reserved			
8	fpll_pwd	R/W	pll power down	0x0
31:9	Reserved			

8.7.2.2 pll_g6_status

Table 8.34: pll_g6_status, Offset Address: 0x004

Bits	Name	Access	Description	Reset
0	updating_mpll_val	RO	pll setting update status	
1	updating_tpll_val	RO	pll setting update status	
2	updating_fpll_val	RO	pll setting update status	
15:3	Reserved			
16	mpll_lock	RO	pll lock status	
17	tpll_lock	RO	pll lock status	
18	fpll_lock	RO	pll lock status	
31:19	Reserved			

8.7.2.3 mpll_csr

Table 8.35: mpll_csr, Offset Address: 0x008

Bits	Name	Access	Description	Reset
6:0	mpll_pre_div_sel	R/W	pll pre_div_sel setting	0x0
7	Reserved			
14:8	mpll_post_div_sel	R/W	pll post_div_sel setting	0x0
16:15	mpll_sel_mode	R/W	pll mode setting	0x0
23:17	mpll_div_sel	R/W	pll div_sel setting	0x0
26:24	mpll_ictrl	R/W	pll ictrl setting	0x0
31:27	Reserved			

8.7.2.4 tpll_csr

Table 8.36: tpll_csr, Offset Address: 0x00c

Bits	Name	Access	Description	Reset
6:0	tpll_pre_div_sel	R/W	pll pre_div_sel setting	0x0
7	Reserved			
14:8	tpll_post_div_sel	R/W	pll post_div_sel setting	0x0
16:15	tpll_sel_mode	R/W	pll mode setting	0x0
23:17	tpll_div_sel	R/W	pll div_sel setting	0x0
26:24	tpll_ictrl	R/W	pll ictrl setting	0x0
31:27	Reserved			

8.7.2.5 fpll_csr

Table 8.37: fpll_csr, Offset Address: 0x010

Bits	Name	Access	Description	Reset
6:0	fpll_pre_div_sel	R/W	pll pre_div_sel setting	0x0
7	Reserved			
14:8	fpll_post_div_sel	R/W	pll post_div_sel setting	0x0
16:15	fpll_sel_mode	R/W	pll mode setting	0x0
23:17	fpll_div_sel	R/W	pll div_sel setting	0x0
26:24	fpll_ictrl	R/W	pll ictrl setting	0x0
31:27	Reserved			

8.7.2.6 pll_g6_ssc_syn_ctrl

Table 8.38: pll_g6_ssc_syn_ctrl, Offset Address: 0x040

Bits	Name	Access	Description	Reset
0	reg_fpll_sel_syn_clk	R/W	fpll gen synthesizer clock source 0:750M 1:1.5G	0x1
1	reg_ddr_ssc_syn_src_en	R/W	ddr pll synthesizer clock enable	0x1
2	reg_mpll_ssc_syn_src_en	R/W	mpll synthesizer clock enable	0x0
3	reg_tpll_ssc_syn_src_en	R/W	tpll synthesizer clock enable	0x0
31:4	Reserved			

8.7.2.7 dpll_ssc_syn_ctrl

Table 8.39: dpll_ssc_syn_ctrl, Offset Address: 0x050

Bits	Name	Access	Description	Reset
0	reg_dpll_ssc_syn_sw_up	W1T	pll synthesizer software update	
1	reg_dpll_ssc_syn_en_ssc	R/W	pll synthesizer ssc enable	0x0
3:2	reg_dpll_ssc_syn_ssc_mode	R/W		0x0
4	reg_dpll_ssc_syn_bypass	R/W		0x0
5	reg_dpll_ssc_syn_extpulse	R/W		0x0
6	reg_dpll_ssc_syn_fix_div	R/W		0x0
31:7	Reserved			

8.7.2.8 dpll_ssc_syn_set

Table 8.40: dpll_ssc_syn_set, Offset Address: 0x054

Bits	Name	Access	Description	Reset
31:0	reg_dpll_ssc_syn_set	R/W	pll synthesizer fraction setting: [31:26] integer 6 bits [26:0] decimal 26 bits	0x0

8.7.2.9 dpll_ssc_syn_span

Table 8.41: dpll_ssc_syn_span, Offset Address: 0x058

Bits	Name	Access	Description	Reset
15:0	reg_dpll_ssc_syn_span	R/W		0x0
31:16	Reserved			

8.7.2.10 dpll_ssc_syn_step

Table 8.42: dpll_ssc_syn_step, Offset Address: 0x05c

Bits	Name	Access	Description	Reset
23:0	reg_dpll_ssc_syn_step	R/W		0x0
31:24	Reserved			

8.7.2.11 mpll_ssc_syn_ctrl

Table 8.43: mpll_ssc_syn_ctrl, Offset Address: 0x060

Bits	Name	Access	Description	Reset
0	reg_mpll_ssc_syn_sw_up	W1T	pll synthesizer software update	
1	reg_mpll_ssc_syn_en_ssc	R/W	pll synthesizer ssc enable	0x0
3:2	reg_mpll_ssc_syn_ssc_mode	R/W		0x0
4	reg_mpll_ssc_syn_bypass	R/W		0x1
5	reg_mpll_ssc_syn_extpulse	R/W		0x0
6	reg_mpll_ssc_syn_fix_div	R/W		0x0
31:7	Reserved			

8.7.2.12 mpll_ssc_syn_set

Table 8.44: mpll_ssc_syn_set, Offset Address: 0x064

Bits	Name	Access	Description	Reset
31:0	reg_mpll_ssc_syn_set	R/W	pll synthesizer fraction setting: [31:26] integer 6 bits [26:0] decimal 26 bits	0x0

8.7.2.13 mpll_ssc_syn_span

Table 8.45: mpll_ssc_syn_span, Offset Address: 0x068

Bits	Name	Access	Description	Reset
15:0	reg_mpll_ssc_syn_span	R/W		0x0
31:16	Reserved			

8.7.2.14 mpll_ssc_syn_step

Table 8.46: mpll_ssc_syn_step, Offset Address: 0x06c

Bits	Name	Access	Description	Reset
23:0	reg_mpll_ssc_syn_step	R/W		0x0
31:24	Reserved			

8.7.2.15 tpll_ssc_syn_ctrl

Table 8.47: tpll_ssc_syn_ctrl, Offset Address: 0x070

Bits	Name	Access	Description	Reset
0	reg_tpll_ssc_syn_sw_up	W1T	pll synthesizer software update	
1	reg_tpll_ssc_syn_en_ssc	R/W	pll synthesizer ssc enable	0x0
3:2	reg_tpll_ssc_syn_ssc_mode	R/W		0x0
4	reg_tpll_ssc_syn_bypass	R/W		0x1
5	reg_tpll_ssc_syn_extpulse	R/W		0x0
6	reg_tpll_ssc_syn_fix_div	R/W		0x0
31:7	Reserved			

8.7.2.16 tpll_ssc_syn_set

Table 8.48: tpll_ssc_syn_set, Offset Address: 0x074

Bits	Name	Access	Description	Reset
31:0	reg_tpll_ssc_syn_set	R/W	pll synthesizer fraction setting: [31:26] integer 6 bits [26:0] decimal 26 bits	0x0

8.7.2.17 tpll_ssc_syn_span

Table 8.49: tpll_ssc_syn_span, Offset Address: 0x078

Bits	Name	Access	Description	Reset
15:0	reg_tpll_ssc_syn_span	R/W		0x0
31:16	Reserved			

8.7.2.18 tpll_ssc_syn_step

Table 8.50: tpll_ssc_syn_step, Offset Address: 0x07c

Bits	Name	Access	Description	Reset
23:0	reg_tpll_ssc_syn_step	R/W		0x0
31:24	Reserved			

8.8 IP/Subsystem Clcok Control Registers Overview

Table 8.51: Clock Gen base address : 0x03002000

Name	Address Offset	Description
clk_en_0	0x000	clock enable register 0
clk_en_1	0x004	clock enable register 1
clk_en_2	0x008	clock enable register 2
clk_en_3	0x00c	clock enable register 3
clk_en_4	0x010	clock enable register 4
clk_sel_0	0x020	clock source selection register 0
clk_byp_0	0x030	clock bypass to xtal register 0
clk_byp_1	0x034	clock bypass to xtal register 1
div_clk_a53_0	0x040	divider register of clk_a53_0
div_clk_a53_1	0x044	divider register of clk_a53_1
div_clk_cpu_axi0	0x048	divider register of clk_cpu_axi0
div_clk_cpu_gic	0x050	divider register of clk_cpu_gic
div_clk_tpu	0x054	divider register of clk_tpu
div_clk_emmc	0x064	divider register of clk_emmc
div_clk_100k_emmc	0x06c	divider register of clk_100k_emmc
div_clk_sd0	0x070	divider register of clk_sd0
div_clk_100k_sd0	0x078	divider register of clk_100k_sd0
div_clk_sd1	0x07c	divider register of clk_sd1
div_clk_100k_sd1	0x084	divider register of clk_100k_sd1
div_clk_spi_nand	0x088	divider register of clk_spi_nand
div_clk_500m_eth0	0x08c	divider register of clk_500m_eth0
div_clk_gpio_db	0x094	divider register of clk_gpio_db
div_clk_sdma_aud0	0x098	divider register of clk_sdma_aud0
div_clk_sdma_aud1	0x09c	divider register of clk_sdma_aud1
div_clk_sdma_aud2	0x0a0	divider register of clk_sdma_aud2
div_clk_sdma_aud3	0x0a4	divider register of clk_sdma_aud3
div_clk_cam0_200	0x0a8	divider register of clk_cam0_200
div_clk_axi4	0x0b8	divider register of clk_axi4
div_clk_axi6	0x0bc	divider register of clk_axi6
div_clk_dsi_esc	0x0c4	divider register of clk_dsi_esc
div_clk_axi_vip	0x0c8	divider register of clk_axi_vip
div_clk_src_vip_sys_0	0x0d0	divider register of clk_src_vip_sys_0
div_clk_src_vip_sys_1	0x0d8	divider register of clk_src_vip_sys_1
div_clk_disp_src_vip	0x0e0	divider register of clk_disp_src_vip
di v_clk_axi_video_codec	0x0e4	divider register of clk_axi_video_codec
div_clk_vc_src0	0x0ec	divider register of clk_vc_src0
div_clk_1m	0x0fc	divider register of clk_1m
div_clk_spi	0x100	divider register of clk_spi
div_clk_i2c	0x104	divider register of clk_i2c
div_clk_src_vip_sys_2	0x110	divider register of clk_src_vip_sys_2
div_clk_audsrc	0x118	divider register of clk_audsrc
div_clk_pwm_src_0	0x120	divider register of clk_pwm_src_0
div_clk_ap_debug	0x128	divider register of clk_ap_debug
div_clk_rtcsys_src_0	0x12c	divider register of clk_rtcsys_src_0
div_clk_c906_0_0	0x130	divider register of clk_c906_0_0
div_clk_c906_0_1	0x134	divider register of clk_c906_0_1

continues on next page

Table 8.51 – continued from previous page

Name	Address Offset	Description
div_clk_c906_1_0	0x138	divider register of clk_c906_1_0
div_clk_c906_1_1	0x13c	divider register of clk_c906_1_1
div_clk_src_vip_sys_3	0x140	divider register of clk_src_vip_sys_3
div_clk_src_vip_sys_4	0x144	divider register of clk_src_vip_sys_4

8.9 IP/Subsystem Clcok Control Registers Description

8.9.1 clk_en_0

Table 8.52: clk_en_0, Offset Address: 0x000

Bits	Name	Access	Description	Reset
0	clk_en_0_0	R/W	Clock Enable for clk_a53 (1: Enable; 0: Gate)	0x1
1	clk_en_0_1	R/W	Clock Enable for clk_cpu_axi0 (1: Enable; 0: Gate)	0x1
2	clk_en_0_2	R/W	Clock Enable for clk_cpu_gic (1: Enable; 0: Gate)	0x1
3	clk_en_0_3	R/W	Clock Enable for clk_xtal_a53 (1: Enable; 0: Gate)	0x1
4	clk_en_0_4	R/W	Clock Enable for clk_tpu (1: Enable; 0: Gate)	0x1
5	Reserved			
6	clk_en_0_6	R/W	Clock Enable for clk_ahb_rom (1: Enable; 0: Gate)	0x1
7	clk_en_0_7	R/W	Clock Enable for clk_ddr_axi_reg (1: Enable; 0: Gate)	0x1
8	clk_en_0_8	R/W	Clock Enable for clk_rtc_25m (1: Enable; 0: Gate)	0x1
9	clk_en_0_9	R/W	Clock Enable for clk_tempsen (1: Enable; 0: Gate)	0x1
10	clk_en_0_10	R/W	Clock Enable for clk_saradc (1: Enable; 0: Gate)	0x1
11	clk_en_0_11	R/W	Clock Enable for clk_efuse (1: Enable; 0: Gate)	0x1
12	clk_en_0_12	R/W	Clock Enable for clk_apb_efuse (1: Enable; 0: Gate)	0x1
13	Reserved			
14	clk_en_0_14	R/W	Clock Enable for clk_xtal_misc (1: Enable; 0: Gate)	0x1
15	clk_en_0_15	R/W	Clock Enable for clk_axi4_emmc (1: Enable; 0: Gate)	0x1
16	clk_en_0_16	R/W	Clock Enable for clk_emmc (1: Enable; 0: Gate)	0x1
17	clk_en_0_17	R/W	Clock Enable for clk_100k_emmc (1: Enable; 0: Gate)	0x1
18	clk_en_0_18	R/W	Clock Enable for clk_axi4_sd0 (1: Enable; 0: Gate)	0x1
19	clk_en_0_19	R/W	Clock Enable for clk_sd0 (1: Enable; 0: Gate)	0x1
20	clk_en_0_20	R/W	Clock Enable for clk_100k_sd0 (1: Enable; 0: Gate)	0x1
21	clk_en_0_21	R/W	Clock Enable for clk_axi4_sd1 (1: Enable; 0: Gate)	0x1
22	clk_en_0_22	R/W	Clock Enable for clk_sd1 (1: Enable; 0: Gate)	0x1

continues on next page

Table 8.52 – continued from previous page

Bits	Name	Access	Description	Reset
23	clk_en_0_23	R/W	Clock Enable for clk_100k_sd1 (1: Enable; 0: Gate)	0x1
24	clk_en_0_24	R/W	Clock Enable for clk_spi_nand (1: Enable; 0: Gate)	0x1
25	clk_en_0_25	R/W	Clock Enable for clk_500m_eth0 (1: Enable; 0: Gate)	0x1
26	clk_en_0_26	R/W	Clock Enable for clk_axi4_eth0 (1: Enable; 0: Gate)	0x1
28:27	Reserved			
29	clk_en_0_29	R/W	Clock Enable for clk_apb_gpio (1: Enable; 0: Gate)	0x1
30	clk_en_0_30	R/W	Clock Enable for clk_apb_gpio_intr (1: Enable; 0: Gate)	0x1
31	clk_en_0_31	R/W	Clock Enable for clk_gpio_db (1: Enable; 0: Gate)	0x1

8.9.2 clk_en_1

Table 8.53: clk_en_1, Offset Address: 0x004

Bits	Name	Access	Description	Reset
0	clk_en_1_0	R/W	Clock Enable for clk_ahb_sf (1: Enable; 0: Gate)	0x1
1	clk_en_1_1	R/W	Clock Enable for clk_sdma_axi (1: Enable; 0: Gate)	0x1
2	clk_en_1_2	R/W	Clock Enable for clk_sdma_aud0 (1: Enable; 0: Gate)	0x1
3	clk_en_1_3	R/W	Clock Enable for clk_sdma_aud1 (1: Enable; 0: Gate)	0x1
4	clk_en_1_4	R/W	Clock Enable for clk_sdma_aud2 (1: Enable; 0: Gate)	0x1
5	clk_en_1_5	R/W	Clock Enable for clk_sdma_aud3 (1: Enable; 0: Gate)	0x1
6	clk_en_1_6	R/W	Clock Enable for clk_apb_i2c (1: Enable; 0: Gate)	0x1
7	clk_en_1_7	R/W	Clock Enable for clk_apb_wdt (1: Enable; 0: Gate)	0x1
8	clk_en_1_8	R/W	Clock Enable for clk_apb_pwm (1: Enable; 0: Gate)	0x1
9	clk_en_1_9	R/W	Clock Enable for clk_apb_spi0 (1: Enable; 0: Gate)	0x1
10	clk_en_1_10	R/W	Clock Enable for clk_apb_spi1 (1: Enable; 0: Gate)	0x1
11	clk_en_1_11	R/W	Clock Enable for clk_apb_spi2 (1: Enable; 0: Gate)	0x1
12	clk_en_1_12	R/W	Clock Enable for clk_apb_spi3 (1: Enable; 0: Gate)	0x1
13	clk_en_1_13	R/W	Clock Enable for clk_187p5m (1: Enable; 0: Gate)	0x1

continues on next page

Table 8.53 – continued from previous page

Bits	Name	Access	Description	Reset
14	clk_en_1_14	R/W	Clock Enable for clk_uart0 (1: Enable; 0: Gate)	0x1
15	clk_en_1_15	R/W	Clock Enable for clk_apb_uart0 (1: Enable; 0: Gate)	0x1
16	clk_en_1_16	R/W	Clock Enable for clk_uart1 (1: Enable; 0: Gate)	0x1
17	clk_en_1_17	R/W	Clock Enable for clk_apb_uart1 (1: Enable; 0: Gate)	0x1
18	clk_en_1_18	R/W	Clock Enable for clk_uart2 (1: Enable; 0: Gate)	0x1
19	clk_en_1_19	R/W	Clock Enable for clk_apb_uart2 (1: Enable; 0: Gate)	0x1
20	clk_en_1_20	R/W	Clock Enable for clk_uart3 (1: Enable; 0: Gate)	0x1
21	clk_en_1_21	R/W	Clock Enable for clk_apb_uart3 (1: Enable; 0: Gate)	0x1
22	clk_en_1_22	R/W	Clock Enable for clk_uart4 (1: Enable; 0: Gate)	0x1
23	clk_en_1_23	R/W	Clock Enable for clk_apb_uart4 (1: Enable; 0: Gate)	0x1
24	clk_en_1_24	R/W	Clock Enable for clk_apb_i2s0 (1: Enable; 0: Gate)	0x1
25	clk_en_1_25	R/W	Clock Enable for clk_apb_i2s1 (1: Enable; 0: Gate)	0x1
26	clk_en_1_26	R/W	Clock Enable for clk_apb_i2s2 (1: Enable; 0: Gate)	0x1
27	clk_en_1_27	R/W	Clock Enable for clk_apb_i2s3 (1: Enable; 0: Gate)	0x1
28	clk_en_1_28	R/W	Clock Enable for clk_axi4_usb (1: Enable; 0: Gate)	0x1
29	clk_en_1_29	R/W	Clock Enable for clk_apb_usb (1: Enable; 0: Gate)	0x1
31:30	Reserved			

8.9.3 clk_en_2

Table 8.54: clk_en_2, Offset Address: 0x008

Bits	Name	Access	Description	Reset
0	Reserved			
1	clk_en_2_1	R/W	Clock Enable for clk_axi4 (1: Enable; 0: Gate)	0x1
2	clk_en_2_2	R/W	Clock Enable for clk_axi6 (1: Enable; 0: Gate)	0x1
3	clk_en_2_3	R/W	Clock Enable for clk_dsi_esc (1: Enable; 0: Gate)	0x1
4	clk_en_2_4	R/W	Clock Enable for clk_axi_vip (1: Enable; 0: Gate)	0x1

continues on next page

Table 8.54 – continued from previous page

Bits	Name	Access	Description	Reset
5	clk_en_2_5	R/W	Clock Enable for clk_src_vip_sys_0 (1: Enable; 0: Gate)	0x1
6	clk_en_2_6	R/W	Clock Enable for clk_src_vip_sys_1 (1: Enable; 0: Gate)	0x1
7	clk_en_2_7	R/W	Clock Enable for clk_disp_src_vip (1: Enable; 0: Gate)	0x1
8	clk_en_2_8	R/W	Clock Enable for clk_axi_video_codec (1: Enable; 0: Gate)	0x1
9	clk_en_2_9	R/W	Clock Enable for clk_vc_src0 (1: Enable; 0: Gate)	0x1
10	clk_en_2_10	R/W	Clock Enable for clk_h264c (1: Enable; 0: Gate)	0x1
11	clk_en_2_11	R/W	Clock Enable for clk_h265c (1: Enable; 0: Gate)	0x1
12	clk_en_2_12	R/W	Clock Enable for clk_jpeg (1: Enable; 0: Gate)	0x1
13	clk_en_2_13	R/W	Clock Enable for clk_apb_jpeg (1: Enable; 0: Gate)	0x1
14	clk_en_2_14	R/W	Clock Enable for clk_apb_h264c (1: Enable; 0: Gate)	0x1
15	clk_en_2_15	R/W	Clock Enable for clk_apb_h265c (1: Enable; 0: Gate)	0x1
16	clk_en_2_16	R/W	Clock Enable for clk_cam0 (1: Enable; 0: Gate)	0x1
17	clk_en_2_17	R/W	Clock Enable for clk_cam1 (1: Enable; 0: Gate)	0x1
18	clk_en_2_18	R/W	Clock Enable for clk_csi_mac0_vip (1: Enable; 0: Gate)	0x1
19	clk_en_2_19	R/W	Clock Enable for clk_csi_mac1_vip (1: Enable; 0: Gate)	0x1
20	clk_en_2_20	R/W	Clock Enable for clk_isp_top_vip (1: Enable; 0: Gate)	0x1
21	clk_en_2_21	R/W	Clock Enable for clk_img_d_vip (1: Enable; 0: Gate)	0x1
22	clk_en_2_22	R/W	Clock Enable for clk_img_v_vip (1: Enable; 0: Gate)	0x1
23	clk_en_2_23	R/W	Clock Enable for clk_sc_top_vip (1: Enable; 0: Gate)	0x1
24	clk_en_2_24	R/W	Clock Enable for clk_sc_d_vip (1: Enable; 0: Gate)	0x1
25	clk_en_2_25	R/W	Clock Enable for clk_sc_v1_vip (1: Enable; 0: Gate)	0x1
26	clk_en_2_26	R/W	Clock Enable for clk_sc_v2_vip (1: Enable; 0: Gate)	0x1
27	clk_en_2_27	R/W	Clock Enable for clk_sc_v3_vip (1: Enable; 0: Gate)	0x1

continues on next page

Table 8.54 – continued from previous page

Bits	Name	Access	Description	Reset
28	clk_en_2_28	R/W	Clock Enable for clk_ldc_vip (1: Enable; 0: Gate)	0x1
29	clk_en_2_29	R/W	Clock Enable for clk_bt_vip (1: Enable; 0: Gate)	0x1
30	clk_en_2_30	R/W	Clock Enable for clk_disp_vip (1: Enable; 0: Gate)	0x1
31	clk_en_2_31	R/W	Clock Enable for clk_dsi_mac_vip (1: Enable; 0: Gate)	0x1

8.9.4 clk_en_3

Table 8.55: clk_en_3, Offset Address: 0x00c

Bits	Name	Access	Description	Reset
0	clk_en_3_0	R/W	Clock Enable for clk_lvds0_vip (1: Enable; 0: Gate)	0x1
1	clk_en_3_1	R/W	Clock Enable for clk_lvds1_vip (1: Enable; 0: Gate)	0x1
2	clk_en_3_2	R/W	Clock Enable for clk_csi0_rx_vip (1: Enable; 0: Gate)	0x1
3	clk_en_3_3	R/W	Clock Enable for clk_csi1_rx_vip (1: Enable; 0: Gate)	0x1
4	clk_en_3_4	R/W	Clock Enable for clk_pad_vi_vip (1: Enable; 0: Gate)	0x1
5	clk_en_3_5	R/W	Clock Enable for clk_1m (1: Enable; 0: Gate)	0x1
6	clk_en_3_6	R/W	Clock Enable for clk_spi (1: Enable; 0: Gate)	0x1
7	clk_en_3_7	R/W	Clock Enable for clk_i2c (1: Enable; 0: Gate)	0x1
8	clk_en_3_8	R/W	Clock Enable for clk_pm (1: Enable; 0: Gate)	0x1
9	clk_en_3_9	R/W	Clock Enable for clk_timer0 (1: Enable; 0: Gate)	0x1
10	clk_en_3_10	R/W	Clock Enable for clk_timer1 (1: Enable; 0: Gate)	0x1
11	clk_en_3_11	R/W	Clock Enable for clk_timer2 (1: Enable; 0: Gate)	0x1
12	clk_en_3_12	R/W	Clock Enable for clk_timer3 (1: Enable; 0: Gate)	0x1
13	clk_en_3_13	R/W	Clock Enable for clk_timer4 (1: Enable; 0: Gate)	0x1
14	clk_en_3_14	R/W	Clock Enable for clk_timer5 (1: Enable; 0: Gate)	0x1
15	clk_en_3_15	R/W	Clock Enable for clk_timer6 (1: Enable; 0: Gate)	0x1
16	clk_en_3_16	R/W	Clock Enable for clk_timer7 (1: Enable; 0: Gate)	0x1

continues on next page

Table 8.55 – continued from previous page

Bits	Name	Access	Description	Reset
17	clk_en_3_17	R/W	Clock Enable for clk_apb_i2c0 (1: Enable; 0: Gate)	0x1
18	clk_en_3_18	R/W	Clock Enable for clk_apb_i2c1 (1: Enable; 0: Gate)	0x1
19	clk_en_3_19	R/W	Clock Enable for clk_apb_i2c2 (1: Enable; 0: Gate)	0x1
20	clk_en_3_20	R/W	Clock Enable for clk_apb_i2c3 (1: Enable; 0: Gate)	0x1
21	clk_en_3_21	R/W	Clock Enable for clk_apb_i2c4 (1: Enable; 0: Gate)	0x1
22	clk_en_3_22	R/W	Clock Enable for clk_wgn (1: Enable; 0: Gate)	0x1
23	clk_en_3_23	R/W	Clock Enable for clk_wgn0 (1: Enable; 0: Gate)	0x1
24	clk_en_3_24	R/W	Clock Enable for clk_wgn1 (1: Enable; 0: Gate)	0x1
25	clk_en_3_25	R/W	Clock Enable for clk_wgn2 (1: Enable; 0: Gate)	0x1
26	clk_en_3_26	R/W	Clock Enable for clk_keyscan (1: Enable; 0: Gate)	0x1
27	clk_en_3_27	R/W	Clock Enable for clk_ahb_sf1 (1: Enable; 0: Gate)	0x1
28	Reserved			
29	clk_en_3_29	R/W	Clock Enable for clk_src_vip_sys_2 (1: Enable; 0: Gate)	0x1
30	clk_en_3_30	R/W	Clock Enable for clk_pad_vi1_vip (1: Enable; 0: Gate)	0x1
31	clk_en_3_31	R/W	Clock Enable for clk_cfg_reg_vip (1: Enable; 0: Gate)	0x1

8.9.5 clk_en_4

Table 8.56: clk_en_4, Offset Address: 0x010

Bits	Name	Access	Description	Reset
0	clk_en_4_0	R/W	Clock Enable for clk_cfg_reg_vc (1: Enable; 0: Gate)	0x1
1	clk_en_4_1	R/W	Clock Enable for clk_audsrc (1: Enable; 0: Gate)	0x1
2	clk_en_4_2	R/W	Clock Enable for clk_apb_audsrc (1: Enable; 0: Gate)	0x1
3	Reserved			
4	clk_en_4_4	R/W	Clock Enable for clk_pwm_src (1: Enable; 0: Gate)	0x1
5	clk_en_4_5	R/W	Clock Enable for clk_ap_debug(1: Enable; 0: Gate)	0x1
6	clk_en_4_6	R/W	Clock Enable for clk_rtcsys_src_0 (1: Enable; 0: Gate)	0x1
7	clk_en_4_7	R/W	Clock Enable for clk_pad_vi2_vip (1: Enable; 0: Gate)	0x1
8	clk_en_4_8	R/W	Clock Enable for clk_csi_be_vip (1: Enable; 0: Gate)	0x1
9	clk_en_4_9	R/W	Clock Enable for clk_vip_ip0_en	0x1
10	clk_en_4_10	R/W	Clock Enable for clk_vip_ip1_en	0x1
11	clk_en_4_11	R/W	Clock Enable for clk_vip_ip2_en	0x1
12	clk_en_4_12	R/W	Clock Enable for clk_vip_ip3_en	0x1
13	clk_en_4_13	R/W	Clock Enable for clk_c906_0_en	0x1
14	clk_en_4_14	R/W	Clock Enable for clk_c906_1_en	0x1
15	clk_en_4_15	R/W	Clock Enable for clk_src_vip_sys_3_en	0x1
16	clk_en_4_16	R/W	Clock Enable for clk_src_vip_sys_4_en	0x1
17	clk_en_4_17	R/W	Clock Enable for clk_ive_vip_en	0x1
18	clk_en_4_18	R/W	Clock Enable for clk_raw_vip_en	0x1
19	clk_en_4_19	R/W	Clock Enable for clk_osdc_vip_en	0x1
20	clk_en_4_20	R/W	Clock Enable for clk_fbc_vip_en	0x1
21	clk_en_4_21	R/W	Clock Enable for clk_cam0_vip_en	0x1
31:22	Reserved			

8.9.6 clk_sel_0

Table 8.57: clk_sel_0, Offset Address: 0x020

Bits	Name	Access	Description	Reset
0	clk_sel_0_0	R/W	Clock Select for A53's clock clk_a53 1: Select div_clk_a53_0 as clock source 0: Select div_clk_a53_1 as clock source	0x0
22:1	Reserved			
23	clk_sel_0_23	R/W	Clock Select for C906's clock clk_c906_0 1: Select div_clk_c906_0_0 as clock source 0: Select div_clk_c906_0_1 as clock source	0x0
24	clk_sel_0_24	R/W	Clock Select for C906's clock clk_c906_1 1: Select div_clk_c906_1_0 as clock source 0: Select div_clk_c906_1_1 as clock source	0x0
31:25	Reserved			

8.9.7 clk_byp_0

Table 8.58: clk_byp_0, Offset Address: 0x030

Bits	Name	Access	Description	Reset
0	clk_byp_0_0	R/W	Clock Bypass to xtal for A53's clock clk_a53	0x1
1	clk_byp_0_1	R/W	Clock Bypass to xtal for A53's clock clk_cpu_axi0	0x1
2	clk_byp_0_2	R/W	Clock Bypass to xtal for A53's clock clk_cpu_gic	0x1
3	clk_byp_0_3	R/W	Clock Bypass to xtal for TPU's clock clk_tpu	0x1
4	Reserved			
5	clk_byp_0_5	R/W	Clock Bypass to xtal for EMMC's clock clk_emmc	0x1
6	clk_byp_0_6	R/W	Clock Bypass to xtal for SD's clock clk_sd0	0x1
7	clk_byp_0_7	R/W	Clock Bypass to xtal for SD's clock clk_sd1	0x1
8	clk_byp_0_8	R/W	Clock Bypass to xtal for SPI_NAND's clock clk_spi_nand	0x1
9	clk_byp_0_9	R/W	Clock Bypass to xtal for ETH0's clock clk_500m_eth0	0x1
10	Reserved			

continues on next page

Table 8.58 – continued from previous page

Bits	Name	Access	Description	Reset
11	clk_byp_0_11	R/W	Clock Bypass to xtal for AUDIO's clock clk_aud0	0x1
12	clk_byp_0_12	R/W	Clock Bypass to xtal for AUDIO's clock clk_aud1	0x1
13	clk_byp_0_13	R/W	Clock Bypass to xtal for AUDIO's clock clk_aud2	0x1
14	clk_byp_0_14	R/W	Clock Bypass to xtal for AUDIO's clock clk_aud3	0x1
15	clk_byp_0_15	R/W	Clock Bypass to xtal for PWM's clock clk_pwm_src	0x1
16	clk_byp_0_16	R/W	Clock Bypass to xtal for TOP's clock clk_cam0_200	0x1
18:17	Reserved			
19	clk_byp_0_19	R/W	Clock Bypass to xtal for FABRIC_AXI4's clock clk_axi4	0x1
20	clk_byp_0_20	R/W	Clock Bypass to xtal for FABRIC_AXI6's clock clk_axi6	0x1
21	clk_byp_0_21	R/W	Clock Bypass to xtal for VIP_SYS's clock clk_dsi_esc	0x1
22	clk_byp_0_22	R/W	Clock Bypass to xtal for VIP_SYS's clock clk_axi_vip	0x1
23	clk_byp_0_23	R/W	Clock Bypass to xtal for VIP_SYS's clock clk_src_vip_sys_0	0x1
24	clk_byp_0_24	R/W	Clock Bypass to xtal for VIP_SYS's clock clk_src_vip_sys_1	0x1
25	clk_byp_0_25	R/W	Clock Bypass to xtal for VIP_SYS's clock clk_disp_src_vip	0x1
26	clk_byp_0_26	R/W	Clock Bypass to xtal for Video_subsys's clock clk_axi_video_codec	0x1
27	clk_byp_0_27	R/W	Clock Bypass to xtal for Video_subsys's clock clk_vc_src0	0x1
29:28	Reserved			
30	clk_byp_0_30	R/W	Clock Bypass to xtal for SPI's clock clk_spi	0x1
31	clk_byp_0_31	R/W	Clock Bypass to xtal for IIC's clock clk_i2c	0x1

8.9.8 clk_byp_1

Table 8.59: clk_byp_1, Offset Address: 0x034

Bits	Name	Access	Description	Reset
0	Reserved			
1	clk_byp_1_1	R/W	Clock Bypass to xtal for VIP_SYS's clock clk_src_vip_sys_2	0x1
2	clk_byp_1_2	R/W	Clock Bypass to xtal for AUDSRC's clock clk_audsrc	0x1
3	clk_byp_1_3	R/W	Clock Bypass to xtal for Video_subsys's clock clk_vc_src2	0x1
4	clk_byp_1_4	R/W	Clock Bypass to xtal for clk_ap_debug	0x1
5	clk_byp_1_5	R/W	Clock Bypass to xtal for clk_src_rtc_sys_0	0x1
6	clk_byp_1_6	R/W	Clock Bypass to xtal for c906_0	0x1
7	clk_byp_1_7	R/W	Clock Bypass to xtal for c906_1	0x1
8	clk_byp_1_8	R/W	Clock Bypass to xtal for VIP_SYS's clock clk_src_vip_sys_3	0x1
9	clk_byp_1_9	R/W	Clock Bypass to xtal for VIP_SYS's clock clk_src_vip_sys_4	0x1
31:10	Reserved			

8.9.9 div_clk_a53_0

Table 8.60: div_clk_a53_0, Offset Address: 0x040

Bits	Name	Access	Description	Reset
31:0	div_clk_a53_0	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : tpll 1 : apll 2 : mipimpll 3 : mpll	0x 0000 0201

8.9.10 div_clk_a53_1

Table 8.61: div_clk_a53_1, Offset Address: 0x044

Bits	Name	Access	Description	Reset
31:0	div_clk_a53_1	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : fpll	0x 0000 0001

8.9.11 div_clk_cpu_axi0

Table 8.62: div_clk_cpu_axi0, Offset Address: 0x048

Bits	Name	Access	Description	Reset
31:0	div_clk_cpu_axi0	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : fpll 1 : disppll	0x 0000 0001

8.9.12 div_clk_cpu_gic

Table 8.63: div_clk_cpu_gic, Offset Address: 0x050

Bits	Name	Access	Description	Reset
31:0	div_clk_cpu_gic	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor	0x 0000 0001

8.9.13 div_clk_tpu

Table 8.64: div_clk_tpu, Offset Address: 0x054

Bits	Name	Access	Description	Reset
31:0	div_clk_tpu	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : tpll 1 : apll 2 : mipimpll 3 : fpll	0x 0000 0301

8.9.14 div_clk_emmc

Table 8.65: div_clk_emmc, Offset Address: 0x064

Bits	Name	Access	Description	Reset
31:0	div_clk_emmc	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : fpll 1 : disppll	0x 0000 0001

8.9.15 div_clk_100k_emmc

Table 8.66: div_clk_100k_emmc, Offset Address: 0x06c

Bits	Name	Access	Description	Reset
31:0	div_clk_100k_emmc	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor	0x 0000 0001

8.9.16 div_clk_sd0

Table 8.67: div_clk_sd0, Offset Address: 0x070

Bits	Name	Access	Description	Reset
31:0	div_clk_sd0	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : fpll 1 : disppll	0x 0000 0001

8.9.17 div_clk_100k_sd0

Table 8.68: div_clk_100k_sd0, Offset Address: 0x078

Bits	Name	Access	Description	Reset
31:0	div_clk_100k_sd0	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor	0x 0000 0001

8.9.18 div_clk_sd1

Table 8.69: div_clk_sd1, Offset Address: 0x07c

Bits	Name	Access	Description	Reset
31:0	div_clk_sd1	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : fpll 1 : disppll	0x 0000 0001

8.9.19 div_clk_100k_sd1

Table 8.70: div_clk_100k_sd1, Offset Address: 0x084

Bits	Name	Access	Description	Reset
31:0	div_clk_100k_sd1	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor	0x 0000 0001

8.9.20 div_clk_spi_nand

Table 8.71: div_clk_spi_nand, Offset Address: 0x088

Bits	Name	Access	Description	Reset
31:0	div_clk_spi_nand	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : fpll 1 : disppll	0x 0000 0001

8.9.21 div_clk_500m_eth0

Table 8.72: div_clk_500m_eth0, Offset Address: 0x08c

Bits	Name	Access	Description	Reset
31:0	div_clk_500m_eth0	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor	0x 0000 0001

8.9.22 div_clk_gpio_db

Table 8.73: div_clk_gpio_db, Offset Address: 0x094

Bits	Name	Access	Description	Reset
31:0	div_clk_gpio_db	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor	0x 0000 0001

8.9.23 div_clk_sdma_aud0

Table 8.74: div_clk_sdma_aud0, Offset Address: 0x098

Bits	Name	Access	Description	Reset
31:0	div_clk_sdma_aud0	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : apll 1 : a24k	0x 0000 0001

8.9.24 div_clk_sdma_aud1

Table 8.75: div_clk_sdma_aud1, Offset Address: 0x09c

Bits	Name	Access	Description	Reset
31:0	div_clk_sdma_aud1	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : apll 1 : a24k	0x 0000 0001

8.9.25 div_clk_sdma_aud2

Table 8.76: div_clk_sdma_aud2, Offset Address: 0x0a0

Bits	Name	Access	Description	Reset
31:0	div_clk_sdma_aud2	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : apll 1 : a24k	0x 0000 0001

8.9.26 div_clk_sdma_aud3

Table 8.77: div_clk_sdma_aud3, Offset Address: 0x0a4

Bits	Name	Access	Description	Reset
31:0	div_clk_sdma_aud3	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : apll 1 : a24k	0x 0000 0001

8.9.27 div_clk_cam0_200

Table 8.78: div_clk_cam0_200, Offset Address: 0x0a8

Bits	Name	Access	Description	Reset
31:0	div_clk_cam0_200	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : xtal 1 : disppll	0x 0000 0001

8.9.28 div_clk_axi4

Table 8.79: div_clk_axi4, Offset Address: 0x0b8

Bits	Name	Access	Description	Reset
31:0	div_clk_axi4	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : fpll 1 : disppll	0x 0000 0001

8.9.29 div_clk_axi6

Table 8.80: div_clk_axi6, Offset Address: 0x0bc

Bits	Name	Access	Description	Reset
31:0	div_clk_axi6	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor	0x 0000 0001

8.9.30 div_clk_dsi_esc

Table 8.81: div_clk_dsi_esc, Offset Address: 0x0c4

Bits	Name	Access	Description	Reset
31:0	div_clk_dsi_esc	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor	0x 0000 0001

8.9.31 div_clk_axi_vip

Table 8.82: div_clk_axi_vip, Offset Address: 0x0c8

Bits	Name	Access	Description	Reset
31:0	div_clk_axi_vip	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : mipimpll 1 : cam0pll 2 : disppll 3 : fpll	0x 0000 0001

8.9.32 div_clk_src_vip_sys_0

Table 8.83: div_clk_src_vip_sys_0, Offset Address: 0x0d0

Bits	Name	Access	Description	Reset
31:0	div_clk_src_vip_sys_0	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : mipimpll 1 : cam0pll 2 : disppll 3 : fpll	0x 0000 0301

8.9.33 div_clk_src_vip_sys_1

Table 8.84: div_clk_src_vip_sys_1, Offset Address: 0x0d8

Bits	Name	Access	Description	Reset
31:0	div_clk_src_vip_sys_1	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : mipimpll 1 : cam0pll 2 : disppll 3 : fpll	0x 0000 0301

8.9.34 div_clk_disp_src_vip

Table 8.85: div_clk_disp_src_vip, Offset Address: 0x0e0

Bits	Name	Access	Description	Reset
31:0	div_clk_disp_src_vip	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor	0x 0000 0001

8.9.35 div_clk_axi_video_codec

Table 8.86: div_clk_axi_video_codec, Offset Address: 0x0e4

Bits	Name	Access	Description	Reset
31:0	div_clk_axi_video_codec	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : apll 1 : mipimpll 2 : cam1pll 3 : fpll	0x 0000 0101

8.9.36 div_clk_vc_src0

Table 8.87: div_clk_vc_src0, Offset Address: 0x0ec

Bits	Name	Access	Description	Reset
31:0	div_clk_vc_src0	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : apll 1 : mipimpll 2 : cam1pll 3 : fpll	0x 0000 0101

8.9.37 div_clk_1m

Table 8.88: div_clk_1m, Offset Address: 0x0fc

Bits	Name	Access	Description	Reset
31:0	div_clk_1m	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor	0x 0000 0001

8.9.38 div_clk_spi

Table 8.89: div_clk_spi, Offset Address: 0x100

Bits	Name	Access	Description	Reset
31:0	div_clk_spi	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor	0x 0000 0001

8.9.39 div_clk_i2c

Table 8.90: div_clk_i2c, Offset Address: 0x104

Bits	Name	Access	Description	Reset
31:0	div_clk_i2c	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor	0x 0000 0001

8.9.40 div_clk_src_vip_sys_2

Table 8.91: div_clk_src_vip_sys_2, Offset Address: 0x110

Bits	Name	Access	Description	Reset
31:0	div_clk_src_vip_sys_2	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : mipimpll 1 : cam0pll 2 : disppll 3 : fpll	0x 0000 0201

8.9.41 div_clk_audsrc

Table 8.92: div_clk_audsrc, Offset Address: 0x118

Bits	Name	Access	Description	Reset
31:0	div_clk_audsrc	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : apll 1 : a24k	0x 0000 0001

8.9.42 div_clk_pwm_src_0

Table 8.93: div_clk_pwm_src_0, Offset Address: 0x120

Bits	Name	Access	Description	Reset
31:0	div_clk_pwm_src	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : fpll 1 : disppll	0x 0000 0001

8.9.43 div_clk_ap_debug

Table 8.94: div_clk_ap_debug, Offset Address: 0x128

Bits	Name	Access	Description	Reset
31:0	div_clk_ap_debug	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor	0x 0000 0001

8.9.44 div_clk_rtcsys_src_0

Table 8.95: div_clk_rtcsys_src_0, Offset Address: 0x12c

Bits	Name	Access	Description	Reset
31:0	div_clk_src_rtc_sys_0	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor	0x 0000 0001

8.9.45 div_clk_c906_0_0

Table 8.96: div_clk_c906_0_0, Offset Address: 0x130

Bits	Name	Access	Description	Reset
31:0	div_clk_c906_0_0	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [1] High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider [2] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [3] Select High Wide Control from Register 0: Select initial value 1: Select High Wide from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : tpll 1 : apll 2 : mipimpll 3 : mpll	0x 0000 0201

8.9.46 div_clk_c906_0_1

Table 8.97: div_clk_c906_0_1, Offset Address: 0x134

Bits	Name	Access	Description	Reset
31:0	div_clk_c906_0_1	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : fpll	0x 0000 0001

8.9.47 div_clk_c906_1_0

Table 8.98: div_clk_c906_1_0l, Offset Address: 0x138

Bits	Name	Access	Description	Reset
31:0	div_clk_c906_1_0	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [1] High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider [2] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [3] Select High Wide Control from Register 0: Select initial value 1: Select High Wide from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : tpll 1 : apll 2 : mipimpll 3 : mpll	0x 0000 0001

8.9.48 div_clk_c906_1_1

Table 8.99: div_clk_c906_1_1, Offset Address: 0x13c

Bits	Name	Access	Description	Reset
31:0	div_clk_c906_1_1	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : fpll	0x 0000 0001

8.9.49 div_clk_src_vip_sys_3

Table 8.100: div_clk_src_vip_sys_3, Offset Address: 0x140

Bits	Name	Access	Description	Reset
31:0	div_clk_src_vip_sys_3	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : mipimpll 1 : cam0pll 2 : disppll 3 : fpll	0x 0000 0001

8.9.50 div_clk_src_vip_sys_4

Table 8.101: div_clk_src_vip_sys_4, Offset Address: 0x144

Bits	Name	Access	Description	Reset
31:0	div_clk_src_vip_sys_4	R/W	[0] Divider Reset Control 0: Assert Reset 1: De-assert Reset [3] Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register [20:16] Clock Divider Factor [9:8] clk_src 0 : mipimpll 1 : cam0pll 2 : disppll 3 : fpll	0x 0000 0201

CHAPTER

NINE

SYSTEM CONTROLLER

9.1 Overview

The system controller implements some system control and enablement of the chip through registers, including system soft reset, clock control, etc. The reset clock has been described in other chapters. This chapter describes the configuration and status registers of some other system function modules.

9.2 Function description

9.2.1 Global reset enable

System global soft reset, debug reset and watchdog reset need to be enabled by configuring the reg_sw_root_reset_en register. Details of each bit are described in reg_sw_root_reset_en.

9.2.2 System DMA channel mapping

This chip system has 8 built-in DMA channels, each configured with 0 ~ 7 channel request interfaces. Channel requests from 0 to 7 are mapped to one of the peripheral interfaces in the table below by the system control registers sdma_dma_ch_remap0 and sdma_dma_ch_remap1. Note that multiple channels cannot be configured as the same peripheral interface.

Configuration steps:

Configure the DMA channel image register sdma_dma_ch_remap0, sdma_dma_ch_remap1, update_dma_remp_0_3, update_dma_remp_4_7 and write 1 to make the mapping effective.

The system DMA channel mapping is as follows:

Table 9.1: System DMA Channel Mapping

No.	DMA Interface	No.	DMA Interface
0	dma_rx_req_i2s0	24	dma_rx_req_i2c0
1	dma_tx_req_i2s0	25	dma_tx_req_i2c0
2	dma_rx_req_i2s1	26	dma_rx_req_i2c1
3	dma_tx_req_i2s1	27	dma_tx_req_i2c1
4	dma_rx_req_i2s2	28	dma_rx_req_i2c2
5	dma_tx_req_i2s2	29	dma_tx_req_i2c2
6	dma_rx_req_i2s3	30	dma_rx_req_i2c3
7	dma_tx_req_i2s3	31	dma_tx_req_i2c3
8	dma_rx_req_n_uart0	32	dma_rx_req_i2c4
9	dma_tx_req_n_uart0	33	dma_tx_req_i2c4
10	dma_rx_req_n_uart1	34	dma_rx_req_tdm0
11	dma_tx_req_n_uart1	35	dma_tx_req_tdm0
12	dma_rx_req_n_uart2	36	dma_rx_req_tdm1
13	dma_tx_req_n_uart2	37	dma_req_audsrc
14	dma_rx_req_n_uart3	38	dma_req_spi_nand
15	dma_tx_req_n_uart3	39	dma_req_spi_nor
16	dma_rx_req_spi0	40	dma_rx_req_n_uart4
17	dma_tx_req_spi0	41	dma_tx_req_n_uart4
18	dma_rx_req_spi1	42	dma_req_spi_nor1
19	dma_tx_req_spi1		
20	dma_rx_req_spi2		
21	dma_tx_req_spi2		
22	dma_rx_req_spi3		
23	dma_tx_req_spi3		

9.2.3 DDR AXI Urgent/Qos configuration

The AXI transmission priority from the subsystem can be configured by ddr_axi_urgent_ow, ddr_axi_urgent, ddr_axi_qos_0, ddr_axi_qos_1 of the control system controller. For details, please refer to the DDR controller chapter *DDR Controller*.

9.3 System Control Register

9.3.1 System Control Register Overview

Bass address 0x03000000

Table 9.2: System Control Registers Overview

Name	Address Offset	Description
conf_info	0x004	conf_info
sys_ctrl_reg	0x008	sys_ctrl_reg
usb_phy_ctrl_reg	0x048	usb_phy_ctrl_reg
sdma_dma_ch_remap0	0x154	sdma_dma_ch_remap0
sdma_dma_ch_remap1	0x158	sdma_dma_ch_remap1
top_timer_clk_sel	0x1a0	top_timer_clk_sel
top_wdt_ctrl	0x1a8	top_timer_clk_sel
ddr_axi_urgent_ow	0x1b8	ddr_axi_urgent_ow
ddr_axi_urgent	0x1bc	ddr_axi_urgent
ddr_axi_qos_0	0x1d8	ddr_axi_qos_0
ddr_axi_qos_1	0x1dc	ddr_axi_qos_1
sd_pwrsw_ctrl	0x1f4	sd_pwrsw_ctrl
sd_pwrsw_time	0x1f8	sd_pwrsw_time
ddr_axi_qos_ow	0x23c	ddr_axi_qos_ow
sd_ctrl_opt	0x294	additional control register for sd
sdma_dma_int_mux	0x298	Mux sdma channel interrupt to different processors

9.3.2 System Control Register Description

9.3.2.1 conf_info

Table 9.3: conf_info, Offset Address: 0x004

Bits	Name	Access	Description	Reset
7:0	boot_sel	RO	[2:0] boot device selection 0: SPI_NAND 1: reserved 2: SPI_NOR 3: EMMC [7:3] : resreved	
8	io_sta_usbid	RO	IO status from USBID PAD	
9	io_sta_usbvbus	RO	IO status from USB_VBUS_DET PAD	
23:10	Reserved			
31:24	io_sta_trap	RO	io_sta_trap[0] : io_boot_rom_din io_sta_trap[1] : io_boot_dev0_din io_sta_trap[2] : io_boot_dev1_din io_sta_trap[3] : io_trap_sd0_pwr_din io_sta_trap[4] : io_pkg_type0_din io_sta_trap[5] : io_pkg_type1_din io_sta_trap[6] : io_pkg_type2_din io_sta_trap[7] : io_trap_zq_din	

9.3.2.2 sys_ctrl_reg

Table 9.4: sys_ctrl_reg, Offset Address: 0x008

Bits	Name	Access	Description	Reset
1:0	Reserved			
5:2	reg_sw_root_reset_en	R/W	bit0 : wdt reset enable bit1 : cdbgrstreq enable bit2 : reserved bit3 : reg_soft_reset_x_system enable	0x0
31:6	Reserved			

9.3.2.3 usb_phy_ctrl_reg

Table 9.5: usb_phy_ctrl_reg, Offset Address: 0x048

Bits	Name	Access	Description	Reset
0	reg_usb_ph y_external_vbusvalid	R/W	external vbus status	0x0
1	reg_usb_drive_vbus	R/W	drive vbus power	0x0
4:2	Reserved			
5	toreg_usb_id_en	RO	usb id pullup status	
6	re g_usb_phy_idpad_c_ow	R/W	usb id overwrite enable	0x0
7	re g_usb_phy_idpad_c_sw	R/W	usb id overwrite value	0x0
8	io_usb_phy_idpad_c	RO	usb id external IO pin status	
9	t oreg_usb_phy_idpad_c	RO	usb id pin status	
31:10	Reserved			

9.3.2.4 sdma_dma_ch_remap0

Table 9.6: sdma_dma_ch_remap0, Offset Address: 0x154

Bits	Name	Access	Description	Reset
5:0	reg_dma_remap_ch0	R/W	dma channel 0 mapping	0x0
7:6	Reserved			
13:8	reg_dma_remap_ch1	R/W	dma channel 1 mapping	0x0
15:14	Reserved			
21:16	reg_dma_remap_ch2	R/W	dma channel 2 mapping	0x0
23:22	Reserved			
29:24	reg_dma_remap_ch3	R/W	dma channel 3 mapping	0x0
30	Reserved			
31	update_dma_remp_0_3	W1T	write 1 to update dma channel0~3 mapping	

9.3.2.5 sdma_dma_ch_remap1

Table 9.7: sdma_dma_ch_remap1, Offset Address: 0x158

Bits	Name	Access	Description	Reset
5:0	reg_dma_remap_ch4	R/W	dma channel 4 mapping	0x0
7:6	Reserved			
13:8	reg_dma_remap_ch5	R/W	dma channel 5 mapping	0x0
15:14	Reserved			
21:16	reg_dma_remap_ch6	R/W	dma channel 6 mapping	0x0
23:22	Reserved			
29:24	reg_dma_remap_ch7	R/W	dma channel 7 mapping	0x0
30	Reserved			
31	update_dma_remp_4_7	W1T	write 1 to update dma channel4~7 mapping	

9.3.2.6 top_timer_clk_sel

Table 9.8: top_timer_clk_sel, Offset Address: 0x1a0

Bits	Name	Access	Description	Reset
7:0	reg_timer_clk_sel	R/W	timer0~7 clock selection. 0: xtal clock,1:32k clock	0x0
31:8	Reserved			

9.3.2.7 top_wdt_ctrl

Table 9.9: top_wdt_ctrl, Offset Address: 0x1a8

Bits	Name	Access	Description	Reset
2:0	reg_wdt_rst_sys_en	R/W	enable wdt0~wdt2 to reset system	0x7
3	Reserved			
6:4	reg_wdt_rst_cpu_en	R/W	enable wdt0~wdt2 to reset cpu	0x0
7	Reserved			
10:8	reg_wdt_clk_sel	R/W	top_wdt clock selection. 0: xtal clock,1:32k clock	0x0
31:11	Reserved			

9.3.2.8 ddr_axi_urgent_ow

Table 9.10: ddr_axi_urgent_ow, Offset Address: 0x1b8

Bits	Name	Access	Description	Reset
0	reg_awurgent_m1_ow	R/W	ddr axi port1 awurgent overwrite enable	0x1
1	reg_arurgent_m1_ow	R/W	ddr axi port1 arurgent overwrite enable	0x1
2	reg_awurgent_m2_ow	R/W	ddr axi port2 awurgent overwrite enable	0x1
3	reg_arurgent_m2_ow	R/W	ddr axi port2 arurgent overwrite enable	0x1
4	reg_awurgent_m3_ow	R/W	ddr axi port3 awurgent overwrite enable	0x1
5	reg_arurgent_m3_ow	R/W	ddr axi port3 arurgent overwrite enable	0x1
6	reg_awurgent_m4_ow	R/W	ddr axi port4 awurgent overwrite enable	0x1
7	reg_arurgent_m4_ow	R/W	ddr axi port4 arurgent overwrite enable	0x1
8	reg_awurgent_m5_ow	R/W	ddr axi port5 awurgent overwrite enable	0x1
9	reg_arurgent_m5_ow	R/W	ddr axi port5 arurgent overwrite enable	0x1
10	reg_awurgent_m6_ow	R/W	ddr axi port6 awurgent overwrite enable	0x1
11	reg_arurgent_m6_ow	R/W	ddr axi port6 arurgent overwrite enable	0x1
31:12	Reserved			

9.3.2.9 ddr_axi_urgent

Table 9.11: ddr_axi_urgent, Offset Address: 0x1bc

Bits	Name	Access	Description	Reset
0	reg_awurgent_m1	R/W	ddr axi port1 awurgent overwrite value	0x0
1	reg_arurgent_m1	R/W	ddr axi port1 arurgent overwrite value	0x0
2	reg_awurgent_m2	R/W	ddr axi port2 awurgent overwrite value	0x0
3	reg_arurgent_m2	R/W	ddr axi port2 arurgent overwrite value	0x0
4	reg_awurgent_m3	R/W	ddr axi port3 awurgent overwrite value	0x0
5	reg_arurgent_m3	R/W	ddr axi port3 arurgent overwrite value	0x0
6	reg_awurgent_m4	R/W	ddr axi port4 awurgent overwrite value	0x0
7	reg_arurgent_m4	R/W	ddr axi port4 arurgent overwrite value	0x0
8	reg_awurgent_m5	R/W	ddr axi port5 awurgent overwrite value	0x0
9	reg_arurgent_m5	R/W	ddr axi port5 arurgent overwrite value	0x0
10	reg_awurgent_m6	R/W	ddr axi port6 awurgent overwrite value	0x0
11	reg_arurgent_m6	R/W	ddr axi port6 arurgent overwrite value	0x0
31:12	Reserved			

9.3.2.10 ddr_axi_qos_0

Table 9.12: ddr_axi_qos_0, Offset Address: 0x1d8

Bits	Name	Access	Description	Reset
3:0	reg_awqos_m1	R/W	ddr axi port1 awqos setting	0x0
7:4	reg_arqos_m1	R/W	ddr axi port1 arqos setting	0x0
11:8	reg_awqos_m2	R/W	ddr axi port2 awqos setting	0x0
15:12	reg_arqos_m2	R/W	ddr axi port2 arqos setting	0x0
19:16	reg_awqos_m3	R/W	ddr axi port3 awqos setting	0x0
23:20	reg_arqos_m3	R/W	ddr axi port3 arqos setting	0x0
27:24	reg_awqos_m4	R/W	ddr axi port4 awqos setting	0x0
31:28	reg_arqos_m4	R/W	ddr axi port4 arqos setting	0x0

9.3.2.11 ddr_axi_qos_1

Table 9.13: ddr_axi_qos_1, Offset Address: 0x1dc

Bits	Name	Access	Description	Reset
3:0	reg_awqos_m5	R/W	ddr axi port5 awqos setting	0x0
7:4	reg_arqos_m5	R/W	ddr axi port5 arqos setting	0x0
11:8	reg_awqos_m6	R/W	ddr axi port6 awqos setting	0x0
15:12	reg_arqos_m6	R/W	ddr axi port6 arqos setting	0x0
31:16	Reserved			

9.3.2.12 sd_pwrsw_ctrl

Table 9.14: sd_pwrsw_ctrl, Offset Address: 0x1f4

Bits	Name	Access	Description	Reset
0	reg_en_pwrsw	R/W	18/33 IO power switch enable	0x0
1	reg_pwrsw_vsel	R/W	18/33 IO power switch enable 0: 3.3v 1: 1.8v	0x1
2	reg_pwrsw_disc	R/W	18/33 IO power switch discharge enable	0x0
3	reg_pwrsw_auto	R/W	18/33 IO power switch auto protect enable	0x1
31:4	Reserved			

9.3.2.13 sd_pwrsw_time

Table 9.15: sd_pwrsw_time, Offset Address: 0x1f8

Bits	Name	Access	Description	Reset
15:0	reg_tpwrup	R/W	18/33 IO power switch, power up protection time is 500x40ns = 20us	0x1f4
31:16	reg_tpwrdn	R/W	18/33 IO power switch, power down protection time is 500x40ns = 20us	0x1f4

9.3.2.14 ddr_axi_qos_ow

Table 9.16: ddr_axi_qos_ow, Offset Address: 0x23c

Bits	Name	Access	Description	Reset
0	reg_awqos_m1_ow	R/W	ddr axi port1 awqos overwrite enable	0x1
1	reg_arqos_m1_ow	R/W	ddr axi port1 arqos overwrite enable	0x1
2	reg_awqos_m2_ow	R/W	ddr axi port2 awqos overwrite enable	0x1
3	reg_arqos_m2_ow	R/W	ddr axi port2 arqos overwrite enable	0x1
4	reg_awqos_m3_ow	R/W	ddr axi port3 awqos overwrite enable	0x1
5	reg_arqos_m3_ow	R/W	ddr axi port3 arqos overwrite enable	0x1
6	reg_awqos_m4_ow	R/W	ddr axi port4 awqos overwrite enable	0x1
7	reg_arqos_m4_ow	R/W	ddr axi port4 arqos overwrite enable	0x1
8	reg_awqos_m5_ow	R/W	ddr axi port5 awqos overwrite enable	0x1
9	reg_arqos_m5_ow	R/W	ddr axi port5 arqos overwrite enable	0x1
10	reg_awqos_m6_ow	R/W	ddr axi port6 awqos overwrite enable	0x1
11	reg_arqos_m6_ow	R/W	ddr axi port6 arqos overwrite enable	0x1
31:12	Reserved			

9.3.2.15 sd_ctrl_opt

Table 9.17: sd_ctrl_opt, Offset Address: 0x294

Bits	Name	Access	Description	Reset
0	reg_sd0_carddet_ow	R/W	sd0 card detect over write enable	0x0
1	reg_sd0_carddet_sw	R/W	sd0 card detect over write value	0x0
7:2	Reserved			
8	reg_sd1_carddet_ow	R/W	sd1 card detect over write enable	0x0
9	reg_sd1_carddet_sw	R/W	sd1 card detect over write value	0x0
10	reg_sd1_phy_sel	R/W	sd1 phy sel	0x0
15:11	Reserved			
16	reg_sd0_pwr_en_polarity	R/W	off chip sd0 pwr en polarity 0: SD_LDO power ctrl high is power on , low is power off 1: SD_LDO power ctrl high is power off, low is power on	0x0
31:17	Reserved			

9.3.2.16 sdma_dma_int_mux

Table 9.18: sdma_dma_int_mux, Offset Address: 0x298

Bits	Name	Access	Description	Reset
8:0	reg_dma_int_mux_cpu0	R/W	This register is used to mux separate sdma channel interrupts to CPU0. These are enable bits corresponding to {intr_cmnreg,intr_ch[7:0]}	0x1FF
9	Reserved			
18:10	reg_dma_int_mux_cpu1	R/W	This register is used to mux separate sdma channel interrupts to CPU1. These are enable bits corresponding to {intr_cmnreg,intr_ch[7:0]}	0x0
19	Reserved			
28:20	reg_dma_int_mux_cpu2	R/W	This register is used to mux separate sdma channel interrupts to CPU2. These are enable bits corresponding to {intr_cmnreg,intr_ch[7:0]}	0x0
31:29	Reserved			

CHAPTER

TEN

PINMUX AND PINCTRL

10.1 Pin Multiplexing (PINMUX)

10.1.1 Mapping between interface functions and signals/pins/FMUX registers

10.1.1.1 ADC

Table 10.1: ADC

Signal Name	Direction	PinName	Function Number	Function select register
(ADC1)	I	ADC1	0	FMUX_GPIO_REG_IOCTRL_ADC1 0x0300_10F8

10.1.1.2 No-die domain ADC

Table 10.2: No-die domain ADC

Signal Name	Direction	PinName	Function Number	Function select register
PWR_GPIO[2](PWRSA R.VIN1)	I	PW R_GPIO2	0	FMUX_GPIO _REG_IOCTRL_PWR_GPIO2 0x0300_10AC
PWR_GPIO[1](PWRSA R1.VIN2)	I	PW R_GPIO1	0	FMUX_GPIO _REG_IOCTRL_PWR_GPIO1 0x0300_10A8
PWR_VBAT_DE T(PWRSA R.VIN3)	I	PWR_V BAT_DET	0	FMUX_GPIO_RE G_IOCTRL_PWR_VBAT_DET 0x0300_107C

10.1.1.3 Audio

Table 10.3: Audio

Signal Name	Direction	PinName	Function Number	Function select register
(PAD_AUD_AINL_MIC)	I	PAD_AUD_AINL_MIC	0	FMUX_GPIO_REG_IOCTRL_PAD_AUD_AINL_MIC 0x0300_11BC
(PAD_AUD_AOUTR)	O	PAD_AUD_AOUTR	0	FMUX_GPIO_REG_IOCTRL_PAD_AUD_AOUTR 0x0300_11C8

10.1.1.4 Ethernet

Table 10.4: Ethernet

Signal Name	Direction	PinName	Function Number	Function select register
(PAD_ETH_RXM)	I/O	PAD _ETH_RX M___E PHY_TXP	0	FMUX_GPIO_R EG_IOCTRL_PAD_ETH_RXM 0x0300_1130
(PAD_ETH_RXP)	I/O	PAD _ETH_RX P___E PHY_TXN	0	FMUX_GPIO_R EG_IOCTRL_PAD_ETH_RXP 0x0300_112C
(PAD_ETH_TXM)	I/O	PAD _ETH_TX M___E PHY_RXP	0	FMUX_GPIO_R EG_IOCTRL_PAD_ETH_TXM 0x0300_1128
(PAD_ETH_TXP)	I/O	PAD _ETH_TX P___E PHY_RXN	0	FMUX_GPIO_R EG_IOCTRL_PAD_ETH_TXP 0x0300_1124
EPHY_LNK_LED	O	SD0_CLK	6	FMUX_GP IO_REG_IOCTRL_SD0_CLK 0x0300_101C
EPHY_LNK_LED	O	PWR _WAKEUP0	6	FMUX_GPIO_R EG_IOCTRL_PWR_WAKEUP0 0x0300_1090
EPHY_LNK_LED	O	SD1_CMD	5	FMUX_GP IO_REG_IOCTRL_SD1_CMD 0x0300_10E0
EPHY_LNK_LED	O	PW R_GPIO1	3	FMUX_GPIO _REG_IOCTRL_PWR_GPIO1 0x0300_10A8
EPHY_SPD_LED	O	SD0_CMD	6	FMUX_GP IO_REG_IOCTRL_SD0_CMD 0x0300_1020
EPHY_SPD_LED	O	PWR _BUT- TON1	6	FMUX_GPIO_R EG_IOCTRL_PWR_BUTTON1 0x0300_1098
EPHY_SPD_LED	O	SD1_CLK	5	FMUX_GP IO_REG_IOCTRL_SD1_CLK 0x0300_10E4
EPHY_SPD_LED	O	PW R_GPIO2	3	FMUX_GPIO _REG_IOCTRL_PWR_GPIO2 0x0300_10AC

10.1.1.5 DSI/LVDS

Table 10.5: DSI/LVDS

Signal Name	Direction	PinName	Function Number	Function select register
(PAD_MIPI_TXM0)	I/O	PAD_MIPI_TXM0	0	FMUX_GPIO_REG_IOCTRL_PAD_MIPI_TXM0 0x0300_11B4
(PAD_MIPI_TXP0)	I/O	PAD_MIPI_TXP0	0	FMUX_GPIO_REG_IOCTRL_PAD_MIPI_TXP0 0x0300_11B8
(PAD_MIPI_TXM1)	I/O	PAD_MIPI_TXM1	3	FMUX_GPIO_REG_IOCTRL_PAD_MIPI_TXM1 0x0300_11AC
(PAD_MIPI_TXP1)	I/O	PAD_MIPI_TXP1	3	FMUX_GPIO_REG_IOCTRL_PAD_MIPI_TXP1 0x0300_11B0
(PAD_MIPI_TXM2)	I/O	PAD_MIPI_TXM2	3	FMUX_GPIO_REG_IOCTRL_PAD_MIPI_TXM2 0x0300_11A4
(PAD_MIPI_TXP2)	I/O	PAD_MIPI_TXP2	3	FMUX_GPIO_REG_IOCTRL_PAD_MIPI_TXP2 0x0300_11A8

10.1.1.6 CSI/sLVDS/HiSPI

Table 10.6: CSI/sLVDS/HiSPI

Signal Name	Direction	PinName	Function Number	Function select register
(PAD_MIPIRX0N)	I/O	PAD_MIPIRX0N	0	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX0N 0x0300_118C
(PAD_MIPIRX0P)	I/O	PAD_MIPIRX0P	0	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX0P 0x0300_1190
(PAD_MIPIRX1N)	I/O	PAD_MIPIRX1N	0	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX1N 0x0300_1184
(PAD_MIPIRX1P)	I/O	PAD_MIPIRX1P	0	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX1P 0x0300_1188
(PAD_MIPIRX2N)	I/O	PAD_MIPIRX2N	0	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX2N 0x0300_117C
(PAD_MIPIRX2P)	I/O	PAD_MIPIRX2P	0	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX2P 0x0300_1180
(PAD_MIPIRX3N)	I/O	PAD_MIPIRX3N	3	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX3N 0x0300_1174
(PAD_MIPIRX3P)	I/O	PAD_MIPIRX3P	3	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX3P 0x0300_1178
(PAD_MIPIRX4N)	I/O	PAD_MIPIRX4N	3	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX4N 0x0300_116C
(PAD_MIPIRX4P)	I/O	PAD_MIPIRX4P	3	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX4P 0x0300_1170

10.1.1.7 Aux clockout

Table 10.7: Aux clockout

Signal Name	Direction	PinName	Function Number	Function select register
AUX0	O	AUX0	0	FMUX _GPIO_REG_IOCTRL_AUX0 0x0300_1078
AUX0	O	SD0_D1	2	FMUX_G PIO_REG_IOCTRL_SD0_D1 0x0300_1028
AUX0	O	U ART0_RX	5	FMUX_GPI O_REG_IOCTRL_UART0_RX 0x0300_1044
AUX0	O	JTAG _CPU_TMS	5	FMUX_GPIO_RE G_IOCTRL_JTAG_CPU_TMS 0x0300_1064
AUX1	O	SD0_D2	2	FMUX_G PIO_REG_IOCTRL_SD0_D2 0x0300_102C
AUX1	O	U ART0_TX	5	FMUX_GPI O_REG_IOCTRL_UART0_TX 0x0300_1040
AUX1	O	JTAG _CPU_TCK	5	FMUX_GPIO_RE G_IOCTRL_JTAG_CPU_TCK 0x0300_1068

10.1.1.8 Camera Interface

Table 10.8: Camera Interface

Signal Name	Direction	PinName	Function Number	Function select register
CAM_HS0	O	SD1_CLK	4	FMUX_GP IO_REG_IOCTRL_SD1_CLK 0x0300_10E4
CAM_HS0	O	PAD_MI PI_TXP0	6	FMUX_GPIO_REG _IOC-TRL_PAD_MIPI_TXP0 0x0300_11B8
CAM_HS0	O	PAD _ETH_RX P___E PHY_TXN	5	FMUX_GPIO_R EG_IOCTRL_PAD_ETH_RXP 0x0300_112C
CAM_MCLK0	O	PAD_MI PI_TXP0	4	FMUX_GPIO_REG _IOC-TRL_PAD_MIPI_TXP0 0x0300_11B8
CAM_MCLK0	O	SD0_D3	1	FMUX_G PIO_REG_IOCTRL_SD0_D3 0x0300_1030
CAM_MCLK0	O	U ART0_RX	1	FMUX_GPI O_REG_IOCTRL_UART0_RX 0x0300_1044

continues on next page

Table 10.8 – continued from previous page

Signal Name	Direction	PinName	Function Number	Function select register
CAM_MCLK0	O	JTAG _CPU_TMS	1	FMUX_GPIO_RE G_IOCTRL_JTAG_CPU_TMS 0x0300_1064
CAM_MCLK0	O	SD1_D3	4	FMUX_G PIO_REG_IOCTRL_SD1_D3 0x0300_10D0
CAM_MCLK0	O	SD1_D2	4	FMUX_G PIO_REG_IOCTRL_SD1_D2 0x0300_10D4
CAM_MCLK0	O	USB_V BUS_DET	4	FMUX_GPIO_RE G_IOCTRL_USB_VBUS_DET 0x0300_1108
CAM_MCLK0	O	PAD_M IPIRX3N	4	FMUX_GPIO_RE G_IOCTRL_PAD_MIPIRX3N 0x0300_1174
CAM_MCLK0	O	PAD_M IPIRX0P	4	FMUX_GPIO_RE G_IOCTRL_PAD_MIPIRX0P 0x0300_1190
CAM_MCLK0	O	MUX _SPI1_CS	2	FMUX_GPIO_R EG_IOCTRL_MUX_SPI1_CS 0x0300_111C
CAM_MCLK0	O	PAD _ETH_RX M___E PHY_TXP	2	FMUX_GPIO_R EG_IOCTRL_PAD_ETH_RXM 0x0300_1130
CAM_MCLK0	O	PAD _ETH_TX P___E PHY_RXN	5	FMUX_GPIO_R EG_IOCTRL_PAD_ETH_TXP 0x0300_1124
CAM_MCLK0	O	PAD_M IPIRX4N	5	FMUX_GPIO_RE G_IOCTRL_PAD_MIPIRX4N 0x0300_116C
CAM_MCLK0	O	GPIO_RT X___E PHY_RTX	5	FMUX_GPI O_REG_IOCTRL_GPIO_RTX 0x0300_11CC
CAM_MCLK1	O	SD1_D1	4	FMUX_G PIO_REG_IOCTRL_SD1_D1 0x0300_10D8
CAM_MCLK1	O	SD1_D0	4	FMUX_G PIO_REG_IOCTRL_SD1_D0 0x0300_10DC
CAM_MCLK1	O	PAD_MI PI_TXM0	4	FMUX_GPIO_REG _IOC- TRL_PAD_MIPI_TXM0 0x0300_11B4
CAM_MCLK1	O	MUX_S PI1_SCK	2	FMUX_GPIO_RE G_IOCTRL_MUX_SPI1_SCK 0x0300_1120

continues on next page

Table 10.8 – continued from previous page

Signal Name	Direction	PinName	Function Number	Function select register
CAM_MCLK1	O	PAD _ETH_RX P___E PHY_TXN	2	FMUX_GPIO_R EG_IOCTRL_PAD_ETH_RXP 0x0300_112C
CAM_MCLK1	O	USB_V BUS_DET	5	FMUX_GPIO_RE G_IOCTRL_USB_VBUS_DET 0x0300_1108
CAM_MCLK1	O	PAD _ETH_TX M___E PHY_RXP	5	FMUX_GPIO_R EG_IOCTRL_PAD_ETH_TXM 0x0300_1128
CAM_MCLK1	O	PAD_M IPIRX4P	5	FMUX_GPIO_RE G_IOCTRL_PAD_MIPIRX4P 0x0300_1170
CAM_MCLK1	O	PAD_M IPIRX0N	5	FMUX_GPIO_RE G_IOCTRL_PAD_MIPIRX0N 0x0300_118C
CAM_MCLK1	O	SD0_D0	1	FMUX_G PIO_REG_IOCTRL_SD0_D0 0x0300_1024
CAM_MCLK1	O	U ART0_TX	1	FMUX_GPI O_REG_IOCTRL_UART0_TX 0x0300_1040
CAM_MCLK1	O	JTAG _CPU_TCK	1	FMUX_GPIO_RE G_IOCTRL_JTAG_CPU_TCK 0x0300_1068
CAM_VS0	O	SD1_CMD	4	FMUX_GP IO_REG_IOCTRL_SD1_CMD 0x0300_10E0
CAM_VS0	O	PAD_MI PI_TXM0	6	FMUX_GPIO_REG_IO CTRL_PAD_MIPI_TXM0 0x0300_11B4
CAM_VS0	O	PAD _ETH_RX M___E PHY_TXP	5	FMUX_GPIO_R EG_IOCTRL_PAD_ETH_RXM 0x0300_1130

10.1.1.9 Parallel Video Out

Table 10.9: Parallel Video Out

Signal Name	Direc-tion	PinName	Func-tion Num-ber	Function select register
VO_CLK0	O	PAD_MI PI_TXP2	2	FMUX_GPIO_REG _IOCTRL_PAD_MIPI_TXP2 0x0300_11A8
VO_D[0]	O	PAD_MI PI_TXM2	2	FMUX_GPIO_REG _IOCTRL_PAD_MIPI_TXM2 0x0300_11A4
VO_D[1]	O	PAD_MI PI_TXP1	2	FMUX_GPIO_REG _IOCTRL_PAD_MIPI_TXP1 0x0300_11B0
VO_D[2]	O	PAD_MI PI_TXM1	2	FMUX_GPIO_REG _IOCTRL_PAD_MIPI_TXM1 0x0300_11AC
VO_D[3]	O	PAD_MI PI_TXP0	2	FMUX_GPIO_REG _IOCTRL_PAD_MIPI_TXP0 0x0300_11B8
VO_D[4]	O	PAD_MI PI_TXM0	2	FMUX_GPIO_REG _IOCTRL_PAD_MIPI_TXM0 0x0300_11B4
VO_D[5]	O	PAD_M IPIRX0P	2	FMUX_GPIO_RE G_IOCTRL_PAD_MIPIRX0P 0x0300_1190
VO_D[6]	O	PAD_M IPIRX0N	2	FMUX_GPIO_RE G_IOCTRL_PAD_MIPIRX0N 0x0300_118C
VO_D[7]	O	PAD_M IPIRX1P	2	FMUX_GPIO_RE G_IOCTRL_PAD_MIPIRX1P 0x0300_1188
VO_D[8]	O	PAD_M IPIRX1N	2	FMUX_GPIO_RE G_IOCTRL_PAD_MIPIRX1N 0x0300_1184
VO_D[9]	O	PAD_M IPIRX2P	2	FMUX_GPIO_RE G_IOCTRL_PAD_MIPIRX2P 0x0300_1180
V O_D[10]	O	PAD_M IPIRX2N	2	FMUX_GPIO_RE G_IOCTRL_PAD_MIPIRX2N 0x0300_117C
V O_D[28]	O	JTAG _CPU_TMS	7	FMUX_GPIO_RE G_IOCTRL_JTAG_CPU_TMS 0x0300_1064
V O_D[29]	O	JTAG _CPU_TCK	7	FMUX_GPIO_RE G_IOCTRL_JTAG_CPU_TCK 0x0300_1068
V O_D[31]	O	AUX0	5	FMUX _GPIO_REG_IOCTRL_AUX0 0x0300_1078
V O_D[32]	O	SD1_D3	0	FMUX_G PIO_REG_IOCTRL_SD1_D3 0x0300_10D0
V O_D[33]	O	SD1_D2	0	FMUX_G PIO_REG_IOCTRL_SD1_D2 0x0300_10D4
V O_D[34]	O	SD1_D1	0	FMUX_G PIO_REG_IOCTRL_SD1_D1 0x0300_10D8
V O_D[35]	O	SD1_D0	0	FMUX_G PIO_REG_IOCTRL_SD1_D0 0x0300_10DC
V O_D[36]	O	SD1_CMD	0	FMUX_GP IO_REG_IOCTRL_SD1_CMD 0x0300_10E0
V O_D[37]	O	SD1_CLK	0	FMUX_GP IO_REG_IOCTRL_SD1_CLK 0x0300_10E4

10.1.1.10 Parallel Video In

Table 10.10: Parallel Video In

Signal Name	Direction	Pin-Name	Function Number	Function select register
VI0_CLK	I	PAD_MIPIRX4N	1	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX4N 0x0300_116C
VI0_D[0]	I	PAD_MIPIRX4P	1	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX4P 0x0300_1170
VI0_D[1]	I	PAD_MIPIRX3N	1	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX3N 0x0300_1174
VI0_D[10]	I	PAD_MIPI_TXP0	1	FMUX_GPIO_REG_IOCTRL_PAD_MIPI_TXP0 0x0300_11B8
VI0_D[11]	I	PAD_MIPI_TXM1	1	FMUX_GPIO_REG_IOCTRL_PAD_MIPI_TXM1 0x0300_11AC
VI0_D[12]	I	PAD_MIPI_TXP1	1	FMUX_GPIO_REG_IOCTRL_PAD_MIPI_TXP1 0x0300_11B0
VI0_D[13]	I	PAD_MIPI_TXM2	1	FMUX_GPIO_REG_IOCTRL_PAD_MIPI_TXM2 0x0300_11A4
VI0_D[14]	I	PAD_MIPI_TXP2	1	FMUX_GPIO_REG_IOCTRL_PAD_MIPI_TXP2 0x0300_11A8
VI0_D[2]	I	PAD_MIPIRX3P	1	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX3P 0x0300_1178
VI0_D[3]	I	PAD_MIPIRX2N	1	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX2N 0x0300_117C
VI0_D[4]	I	PAD_MIPIRX2P	1	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX2P 0x0300_1180
VI0_D[5]	I	PAD_MIPIRX1N	1	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX1N 0x0300_1184
VI0_D[6]	I	PAD_MIPIRX1P	1	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX1P 0x0300_1188
VI0_D[7]	I	PAD_MIPIRX0N	1	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX0N 0x0300_118C
VI0_D[8]	I	PAD_MIPIRX0P	1	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX0P 0x0300_1190
VI0_D[9]	I	PAD_MIPI_TXM0	1	FMUX_GPIO_REG_IOCTRL_PAD_MIPI_TXM0 0x0300_11B4
VI1_D[13]	I	PAD_MIPIRX4N	2	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX4N 0x0300_116C
VI1_D[14]	I	PAD_MIPIRX4P	2	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX4P 0x0300_1170
VI1_D[15]	I	PAD_MIPIRX3N	2	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX3N 0x0300_1174
VI1_D[16]	I	PAD_MIPIRX3P	2	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX3P 0x0300_1178
VI1_D[17]	I	PAD_MIPIRX2N	4	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX2N 0x0300_117C
VI1_D[18]	I	PAD_MIPIRX2P	4	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX2P 0x0300_1180

10.1.1.11 eMMC

Table 10.11: eMMC

Signal Name	Direction	PinName	Function Number	Function select register
EMMC_CLK	O	E MMC_CLK	0	FMUX_GPI O_REG_IOCTRL_EMMC_CLK 0x0300_1050
EMMC_CMD	I/O	E MMC_CMD	0	FMUX_GPI O_REG_IOCTRL_EMMC_CMD 0x0300_105C
EMMC_DAT[0]	I/O	EM MC_DAT0	0	FMUX_GPIO _REG_IOCTRL_EMMC_DAT0 0x0300_1054
EMMC_DAT[1]	I/O	EM MC_DAT1	0	FMUX_GPIO _REG_IOCTRL_EMMC_DAT1 0x0300_1060
EMMC_DAT[2]	I/O	EM MC_DAT2	0	FMUX_GPIO _REG_IOCTRL_EMMC_DAT2 0x0300_104C
EMMC_DAT[3]	I/O	EM MC_DAT3	0	FMUX_GPIO _REG_IOCTRL_EMMC_DAT3 0x0300_1058

10.1.1.12 SPI_NAND

Table 10.12: SPI_NAND

Signal Name	Direction	PinName	Function Number	Function select register
SPINAND_CLK	O	E MMC_CLK	2	FMUX_GPI O_REG_IOCTRL_EMMC_CLK 0x0300_1050
SPINAND_CS	O	EM MC_DAT1	2	FMUX_GPIO _REG_IOCTRL_EMMC_DAT1 0x0300_1060
SPINAND_HOLD	I/O	EM MC_DAT2	2	FMUX_GPIO _REG_IOCTRL_EMMC_DAT2 0x0300_104C
SPINAND_MISO	I/O	E MMC_CMD	2	FMUX_GPI O_REG_IOCTRL_EMMC_CMD 0x0300_105C
SPINAND_MOSI	I/O	EM MC_DAT0	2	FMUX_GPIO _REG_IOCTRL_EMMC_DAT0 0x0300_1054
SPINAND_WP	I/O	EM MC_DAT3	2	FMUX_GPIO _REG_IOCTRL_EMMC_DAT3 0x0300_1058

10.1.1.13 SPI_NOR

Table 10.13: SPI_NOR

Signal Name	Direction	PinName	Function Number	Function select register
SPINOR_CS_X	O	EM MC_DAT1	1	FMUX_GPIO _REG_IOCTRL_EMMC_DAT1 0x0300_1060
SPINOR_HOLD_X	I/O	EM MC_DAT2	1	FMUX_GPIO _REG_IOCTRL_EMMC_DAT2 0x0300_104C
SPINOR_MISO	I/O	E MMC_CMD	1	FMUX_GPI O_REG_IOCTRL_EMMC_CMD 0x0300_105C
SPINOR_MOSI	I/O	EM MC_DAT0	1	FMUX_GPIO _REG_IOCTRL_EMMC_DAT0 0x0300_1054
SPINOR_SCK	O	E MMC_CLK	1	FMUX_GPI O_REG_IOCTRL_EMMC_CLK 0x0300_1050
SPINOR_WP_X	I/O	EM MC_DAT3	1	FMUX_GPIO _REG_IOCTRL_EMMC_DAT3 0x0300_1058

10.1.1.14 I2C

Table 10.14: I2C

Signal Name	Direction	PinName	Function Number	Function select register
IIC1_SCL	I/O	PAD_MIPIRX4P	4	FMUX_GPIO_REG _IOCTRL_PAD_MIPIRX4P 0x0300_1170
IIC1_SCL	I/O	PAD_MIPIRX0N	4	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX0N 0x0300_118C
IIC1_SCL	I/O	SD0_CMD	1	FMUX_GPIO_REG_IOCTRL_SD0_CMD 0x0300_1020
IIC1_SCL	I/O	SD0_D2	1	FMUX_GPIO_REG_IOCTRL_SD0_D2 0x0300_102C
IIC1_SCL	I/O	PAD_MIPI_TXP2	4	FMUX_GPIO_REG _IOCTRL_PAD_MIPI_TXP2 0x0300_11A8
IIC1_SCL	I/O	SD1_D3	2	FMUX_GPIO_REG_IOCTRL_SD1_D3 0x0300_10D0
IIC1_SCL	I/O	MUX_SPI1_MOSI	2	FMUX_GPIO_REG _IOCTRL_MUX_SPI1_MOSI 0x0300_1118
IIC1_SCL	I/O	PAD _ETH_TXP___EPHY_RXN	2	FMUX_GPIO_REG _IOCTRL_PAD_ETH_TXP 0x0300_1124
IIC1_SCL	I/O	SD1_D2	1	FMUX_GPIO_REG_IOCTRL_SD1_D2 0x0300_10D4
IIC1_SDA	I/O	PAD_MIPIRX4N	4	FMUX_GPIO_REG _IOCTRL_PAD_MIPIRX4N 0x0300_116C

To be continued ……

Table 10.15: I2C (continued)

Signal Name	Direction	PinName	Function Number	Function select register
IIC1_SDA	I/O	PAD_MIPIRX1P	4	FMUX_GPIO_REG _IOCTRL_PAD_MIPIRX1P 0x0300_1188
IIC1_SDA	I/O	SD0_CLK	1	FMUX_GPIO_REG _IOCTRL_SD0_CLK 0x0300_101C
IIC1_SDA	I/O	SD0_D1	1	FMUX_GPIO_REG _IOCTRL_SD0_D1 0x0300_1028
IIC1_SDA	I/O	PAD_MIPI_TXM2	4	FMUX_GPIO_REG _IOCTRL_PAD_MIPI_TXM2 0x0300_11A4
IIC1_SDA	I/O	SD1_D0	2	FMUX_GPIO_REG _IOCTRL_SD1_D0 0x0300_10DC
IIC1_SDA	I/O	MUX_SPI1_MISO	2	FMUX_GPIO_REG _IOCTRL_MUX_SPI1_MISO 0x0300_1114
IIC1_SDA	I/O	PAD _ETH_TXM___EPHY_RXP	2	FMUX_GPIO_REG _IOCTRL_PAD_ETH_TXM 0x0300_1128
IIC1_SDA	I/O	SD1_D1	1	FMUX_GPIO_REG _IOCTRL_SD1_D1 0x0300_10D8
IIC2_SCL	I/O	PAD_MIPI_TXP1	4	FMUX_GPIO_REG _IOCTRL_PAD_MIPI_TXP1 0x0300_11B0
IIC2_SCL	I/O	PWR_GPIO1	6	FMUX_GPIO_REG _IOCTRL_PWR_GPIO1 0x0300_10A8
IIC2_SDA	I/O	PAD_MIPI_TXM1	4	FMUX_GPIO_REG _IOCTRL_PAD_MIPI_TXM1 0x0300_11AC
IIC2_SDA	I/O	PWR_GPIO2	6	FMUX_GPIO_REG _IOCTRL_PWR_GPIO2 0x0300_10AC
IIC3_SCL	I/O	SD1_CMD	2	FMUX_GPIO_REG _IOCTRL_SD1_CMD 0x0300_10E0
IIC3_SDA	I/O	SD1_CLK	2	FMUX_GPIO_REG _IOCTRL_SD1_CLK 0x0300_10E4
IIC4_SCL	I/O	PWR_WAKEUP0	5	FMUX_GPIO_REG _IOCTRL_PWR_WAKEUP0 0x0300_1090
IIC4_SCL	I/O	PAD_MIPIRX2N	5	FMUX_GPIO_REG _IOCTRL_PAD_MIPIRX2N 0x0300_117C
IIC4_SDA	I/O	PWR _BUTTON1	5	FMUX_GPIO_REG _IOCTRL_PWR_BUTTON1 0x0300_1098
IIC4_SDA	I/O	PAD_MIPIRX2P	5	FMUX_GPIO_REG _IOCTRL_PAD_MIPIRX2P 0x0300_1180

10.1.1.15 No-die domain I2C

Table 10.16: No-die domain I2C

Signal Name	Direction	PinName	Function Number	Function select register
PWR_IIC_SCL	I/O	PWR_GPIO1	5	FMUX_GPIO_REG_IOCTRL_PWR_GPIO1 0x0300_10A8
PWR_IIC_SDA	I/O	PWR_GPIO2	5	FMUX_GPIO_REG_IOCTRL_PWR_GPIO2 0x0300_10AC

10.1.1.16 IIS

Table 10.17: IIS

Signal Name	Direction	PinName	Function Number	Function select register
IIS1_BCLK	I/O	PA D_AUD_A INL_MIC	4	FMUX_GPIO_REG_IO CTRL_PAD_AUD_AINL_MIC 0x0300_11BC
IIS1_DI	I	PAD_AU D_AOUTR	4	FMUX_GPIO_REG_IO CTRL_PAD_AUD_AOUTR 0x0300_11C8
IIS1_DO	O	PAD_AU D_AOUTR	6	FMUX_GPIO_REG _IOC- TRL_PAD_AUD_AOUTR 0x0300_11C8
IIS1_MCLK	I/O	AUX0	4	FMUX _GPIO_REG_IOCTRL_AUX0 0x0300_1078
IIS2_BCLK	I/O	PA D_AUD_A INL_MIC	5	FMUX_GPIO_REG_IO CTRL_PAD_AUD_AINL_MIC 0x0300_11BC
IIS2_BCLK	I/O	PAD _ETH_TX M___E PHY_RXP	7	FMUX_GPIO_R EG_IOCTRL_PAD_ETH_TXM 0x0300_1128
IIS2_DI	I	PAD _ETH_RX M___E PHY_TXP	7	FMUX_GPIO_R EG_IOCTRL_PAD_ETH_RXM 0x0300_1130
IIS2_DO	O	PAD_AU D_AOUTR	5	FMUX_GPIO_REG _IOC- TRL_PAD_AUD_AOUTR 0x0300_11C8
IIS2_DO	O	PAD _ETH_RX P___E PHY_TXN	7	FMUX_GPIO_R EG_IOCTRL_PAD_ETH_RXP 0x0300_112C
IIS2_LRCK	I/O	PAD _ETH_TX P___E PHY_RXN	7	FMUX_GPIO_R EG_IOCTRL_PAD_ETH_TXP 0x0300_1124
KEY_COL0	I/O	MUX_SP I1_MOSI	5	FMUX_GPIO_REG _IOC- TRL_MUX_SPI1_MOSI 0x0300_1118
KEY_COL1	I/O	MUX_SP I1_MISO	5	FMUX_GPIO_REG _IOC- TRL_MUX_SPI1_MISO 0x0300_1114
KEY_COL2	I/O	ADC1	4	FMUX _GPIO_REG_IOCTRL_ADC1 0x0300_10F8
KEY_ROW0	I/O	PAD_M IPIRX4N	6	FMUX_GPIO_RE G_IOCTRL_PAD_MIPIRX4N 0x0300_116C
KEY_ROW1	I/O	PAD_M IPIRX4P	6	FMUX_GPIO_RE G_IOCTRL_PAD_MIPIRX4P 0x0300_1170

To be continued ……

Table 10.18: IIS (continued)

Signal Name	Direction	PinName	Function Number	Function select register
KEY_ROW2	I/O	PAD_MIPIRX1P	6	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX1P 0x0300_1188
KEY_ROW2	I/O	MUX_SPI1_SCK	5	FMUX_GPIO_REG_IOCTRL_MUX_SPI1_SCK 0x0300_1120
KEY_ROW3	I/O	PAD_MIPIRX1N	6	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX1N 0x0300_1184
KEY_ROW3	I/O	MUX_SPI1_CS	5	FMUX_GPIO_REG_IOCTRL_MUX_SPI1_CS 0x0300_111C

10.1.1.17 PWM

Table 10.19: PWM

Signal Name	Direction	PinName	Function Number	Function select register
PWM[0]	I/O	PWM0_BUCK	0	FMUX_GPIO_REG_IOCTRL_PWM0_BUCK 0x0300_10EC
PWM[1]	I/O	GPIO_RTX___EPHY_RTX	4	FMUX_GPIO_REG_IOCTRL_GPIO_RTX 0x0300_11CC
PWM[2]	I/O	GPIO_ZQ__PAD_ZQ	4	FMUX_GPIO_REG_IOCTRL_GPIO_ZQ 0x0300_11D0
PWM[4]	I/O	UART0_TX	2	FMUX_GPIO_REG_IOCTRL_UART0_TX 0x0300_1040
PWM[4]	I/O	SD1_D3	7	FMUX_GPIO_REG_IOCTRL_SD1_D3 0x0300_10D0
PWM[5]	I/O	UART0_RX	2	FMUX_GPIO_REG_IOCTRL_UART0_RX 0x0300_1044
PWM[5]	I/O	SD1_D2	7	FMUX_GPIO_REG_IOCTRL_SD1_D2 0x0300_10D4
PWM[6]	I/O	JTAG_CPU_TCK	2	FMUX_GPIO_REG_IOCTRL_JTAG_CPU_TCK 0x0300_1068
PWM[6]	I/O	SD1_D1	7	FMUX_GPIO_REG_IOCTRL_SD1_D1 0x0300_10D8

continues on next page

Table 10.19 – continued from previous page

Signal Name	Direction	PinName	Function Number	Function select register
PWM[7]	I/O	JTAG _CPU_TMS	2	FMUX_GPIO_RE G_IOCTRL_JTAG_CPU_TMS 0x0300_1064
PWM[7]	I/O	SD1_D0	7	FMUX_G PIO_REG_IOCTRL_SD1_D0 0x0300_10DC
PWM[8]	I/O	PW R_GPIO0	4	FMUX_GPIO _REG_IOCTRL_PWR_GPIO0 0x0300_10A4
PWM[8]	I/O	MUX_SP I1_MOSI	4	FMUX_GPIO_REG_IO CTRL_MUX_SPI1_MOSI 0x0300_1118
PWM[8]	I/O	SD1_CMD	7	FMUX_GP IO_REG_IOCTRL_SD1_CMD 0x0300_10E0
PWM[8]	I/O	PAD_MI PI_TXM2	5	FMUX_GPIO_REG_IO CTRL_PAD_MIPI_TXM2 0x0300_11A4
PWM[9]	I/O	PW R_GPIO1	4	FMUX_GPIO _REG_IOCTRL_PWR_GPIO1 0x0300_10A8
PWM[9]	I/O	MUX_SP I1_MISO	4	FMUX_GPIO_REG_IO CTRL_MUX_SPI1_MISO 0x0300_1114
PWM[9]	I/O	SD1_CLK	7	FMUX_GP IO_REG_IOCTRL_SD1_CLK 0x0300_10E4
PWM[9]	I/O	PAD_MI PI_TXP2	5	FMUX_GPIO_REG_IO CTRL_PAD_MIPI_TXP2 0x0300_11A8
PWM[10]	I/O	PW R_GPIO2	4	FMUX_GPIO _REG_IOCTRL_PWR_GPIO2 0x0300_10AC
PWM[10]	I/O	MUX_S PI1_SCK	4	FMUX_GPIO_RE G_IOCTRL_MUX_SPI1_SCK 0x0300_1120
PWM[10]	I/O	SD0_D3	5	FMUX_G PIO_REG_IOCTRL_SD0_D3 0x0300_1030
PWM[10]	I/O	PAD_MI PI_TXM1	5	FMUX_GPIO_REG_IO CTRL_PAD_MIPI_TXM1 0x0300_11AC
PWM[11]	I/O	MUX _SPI1_CS	4	FMUX_GPIO_R EG_IOCTRL_MUX_SPI1_CS 0x0300_111C
PWM[11]	I/O	SD0_D2	5	FMUX_G PIO_REG_IOCTRL_SD0_D2 0x0300_102C

continues on next page

Table 10.19 – continued from previous page

Signal Name	Direction	PinName	Function Number	Function select register
PWM[11]	I/O	PAD_MIPI_TXP1	5	FMUX_GPIO_REG_IOCTRL_PAD_MIPI_TXP1 0x0300_11B0
PWM[12]	I/O	PAD_ETH_TXM___EPHY_RXP	4	FMUX_GPIO_REG_IOCTRL_PAD_ETH_TXM 0x0300_1128
PWM[12]	I/O	SD0_D1	5	FMUX_GPIO_REG_IOCTRL_SD0_D1 0x0300_1028
PWM[13]	I/O	PAD_ETH_TXP___EPHY_RXN	4	FMUX_GPIO_REG_IOCTRL_PAD_ETH_TXP 0x0300_1124
PWM[13]	I/O	SD0_D0	5	FMUX_GPIO_REG_IOCTRL_SD0_D0 0x0300_1024
PWM[14]	I/O	PAD_ETH_RXM___EPHY_TXP	4	FMUX_GPIO_REG_IOCTRL_PAD_ETH_RXM 0x0300_1130
PWM[14]	I/O	SD0_CMD	5	FMUX_GPIO_REG_IOCTRL_SD0_CMD 0x0300_1020
PWM[14]	I/O	PAD_MIPI_TXM0	5	FMUX_GPIO_REG_IOCTRL_PAD_MIPI_TXM0 0x0300_11B4
PWM[15]	I/O	PAD_ETH_RXP___EPHY_TXN	4	FMUX_GPIO_REG_IOCTRL_PAD_ETH_RXP 0x0300_112C
PWM[15]	I/O	SD0_CLK	5	FMUX_GPIO_REG_IOCTRL_SD0_CLK 0x0300_101C
PWM[15]	I/O	PAD_MIPI_TXP0	5	FMUX_GPIO_REG_IOCTRL_PAD_MIPI_TXP0 0x0300_11B8

10.1.1.18 CA53 JTAG(2W) RISCV JTAG(4W) I2C0

Table 10.20: CA53 JTAG(2W) RISCV JTAG(4W) I2C0

Signal Name	Direction	PinName	Function Number	Function select register
CR_4WTCK	I/O	JTAG _CPU_TCK	0	FMUX_GPIO_RE G_IOCTRL_JTAG_CPU_TCK 0x0300_1068
CR_4WTMS	I/O	JTAG _CPU_TMS	0	FMUX_GPIO_RE G_IOCTRL_JTAG_CPU_TMS 0x0300_1064
CR_2WTCK	I/O	PAD_M IPIRX3P	0	FMUX_GPIO_RE G_IOCTRL_PAD_MIPIRX3P 0x0300_1178
CR_2WTCK	I/O	PAD_MI PI_TXP1	0	FMUX_GPIO_REG_IO CTRL_PAD_MIPI_TXP1 0x0300_11B0
CR_2WTCK	I/O	PW R_GPIO2	7	FMUX_GPIO _REG_IOCTRL_PWR_GPIO2 0x0300_10AC
CR_SCL0	I/O	PAD_M IPIRX4N	0	FMUX_GPIO_RE G_IOCTRL_PAD_MIPIRX4N 0x0300_116C
CR_SCL0	I/O	PAD_MI PI_TXP2	0	FMUX_GPIO_REG_IO CTRL_PAD_MIPI_TXP2 0x0300_11A8
CR_2WTMS	I/O	PAD_M IPIRX3N	0	FMUX_GPIO_RE G_IOCTRL_PAD_MIPIRX3N 0x0300_1174
CR_2WTMS	I/O	PAD_MI PI_TXM1	0	FMUX_GPIO_REG_IO CTRL_PAD_MIPI_TXM1 0x0300_11AC
CR_SDA0	I/O	PAD_M IPIRX4P	0	FMUX_GPIO_RE G_IOCTRL_PAD_MIPIRX4P 0x0300_1170
CR_SDA0	I/O	PAD_MI PI_TXM2	0	FMUX_GPIO_REG_IO CTRL_PAD_MIPI_TXM2 0x0300_11A4
CR_SDA0	I/O	PW R_GPIO1	7	FMUX_GPIO _REG_IOCTRL_PWR_GPIO1 0x0300_10A8
CR_4WTDI	I/O	I IC0_SCL	0	FMUX_GPI O_REG_IOCTRL_IIC0_SCL 0x0300_1070
CR_4WTDO	I/O	I IC0_SDA	0	FMUX_GPI O_REG_IOCTRL_IIC0_SDA 0x0300_1074

10.1.1.19 System

Table 10.21: System

Signal Name	Direction	PinName	Function Number	Function select register
PWR_BUTTON1	I	PWR_BUTTON1	0	FMUX_GPIO_REG_IOCTRL_PWR_BUTTON1 0x0300_1098
PWR_RSTN	I	PWR_RSTN	0	FMUX_GPIO_REG_IOCTRL_PWR_RSTN 0x0300_1080
PWR_SEQ1	O	PWR_SEQ1	0	FMUX_GPIO_REG_IOCTRL_PWR_SEQ1 0x0300_1084
PWR_SEQ2	O	PWR_SEQ2	0	FMUX_GPIO_REG_IOCTRL_PWR_SEQ2 0x0300_1088
PWR_WAKEUP0	I	PWR_WAKEUP0	0	FMUX_GPIO_REG_IOCTRL_PWR_WAKEUP0 0x0300_1090
USB_VBUS_DET	I	USB_VBUS_DET	0	FMUX_GPIO_REG_IOCTRL_USB_VBUS_DET 0x0300_1108

10.1.1.20 No-die domain IR

Table 10.22: No-die domain IR

Signal Name	Direction	PinName	Function Number	Function select register
PWR_IR0	I	PWR_WAKEUP0	1	FMUX_GPIO_REG_IOCTRL_PWR_WAKEUP0 0x0300_1090

10.1.1.21 SPI_NOR1

Table 10.23: SPI_NOR1

Signal Name	Direction	PinName	Function Number	Function select register
PWR_SPINOR1_CS_X	O	SD1_D3	6	FMUX_GPIO_REG_IOCTRL_SD1_D3 0x0300_10D0
PWR_SPINOR1_HOLD_X	I/O	SD1_D2	6	FMUX_GPIO_REG_IOCTRL_SD1_D2 0x0300_10D4
PWR_SPINOR1_MISO	I/O	SD1_D0	6	FMUX_GPIO_REG_IOCTRL_SD1_D0 0x0300_10DC
PWR_SPINOR1_MOSI	I/O	SD1_CMD	6	FMUX_GPIO_REG_IOCTRL_SD1_CMD 0x0300_10E0
PWR_SPINOR1_SCK	O	SD1_CLK	6	FMUX_GPIO_REG_IOCTRL_SD1_CLK 0x0300_10E4
PWR_SPINOR1_WP_X	I/O	SD1_D1	6	FMUX_GPIO_REG_IOCTRL_SD1_D1 0x0300_10D8

10.1.1.22 SD1

Table 10.24: SD1

Signal Name	Direction	PinName	Function Number	Function select register
PWR_SD1_CLK	O	SD1_CLK	0	FMUX_GPIO_REG_IOCTRL_SD1_CLK 0x0300_10E4
PWR_SD1_CMD	I/O	SD1_CMD	0	FMUX_GPIO_REG_IOCTRL_SD1_CMD 0x0300_10E0
PWR_SD1_D0	I/O	SD1_D0	0	FMUX_GPIO_REG_IOCTRL_SD1_D0 0x0300_10DC
PWR_SD1_D1	I/O	SD1_D1	0	FMUX_GPIO_REG_IOCTRL_SD1_D1 0x0300_10D8
PWR_SD1_D2	I/O	SD1_D2	0	FMUX_GPIO_REG_IOCTRL_SD1_D2 0x0300_10D4
PWR_SD1_D3	I/O	SD1_D3	0	FMUX_GPIO_REG_IOCTRL_SD1_D3 0x0300_10D0

10.1.1.23 SD0

Table 10.25: SD0

Signal Name	Direction	PinName	Function Number	Function select register
SDIO0_CD	I	SD0_CD	0	FMUX_G PIO_REG_IOCTRL_SD0_CD 0x0300_1034
SDIO0_CLK	O	SD0_CLK	0	FMUX_GP IO_REG_IOCTRL_SD0_CLK 0x0300_101C
SDIO0_CMD	I/O	SD0_CMD	0	FMUX_GP IO_REG_IOCTRL_SD0_CMD 0x0300_1020
SDIO0_D[0]	I/O	SD0_D0	0	FMUX_G PIO_REG_IOCTRL_SD0_D0 0x0300_1024
SDIO0_D[1]	I/O	SD0_D1	0	FMUX_G PIO_REG_IOCTRL_SD0_D1 0x0300_1028
SDIO0_D[2]	I/O	SD0_D2	0	FMUX_G PIO_REG_IOCTRL_SD0_D2 0x0300_102C
SDIO0_D[3]	I/O	SD0_D3	0	FMUX_G PIO_REG_IOCTRL_SD0_D3 0x0300_1030
SDIO0_PWR_EN	O	SD0 _PWR_EN	0	FMUX_GPIO _REG_IOCTRL_SD0_PWR_EN 0x0300_1038

10.1.1.24 SPI

Table 10.26: SPI

Signal Name	Direction	PinName	Function Number	Function select register
SPI0_CS_X	O	SD0_D3	2	FMUX_GPIO_REG_IOCTRL_SD0_D3 0x0300_1030
SPI0_CS_X	O	PAD_MIPI_TXP2	6	FMUX_GPIO_REG_IOCTRL_PAD_MIPI_TXP2 0x0300_11A8
SPI0_SCK	O	SD0_CLK	2	FMUX_GPIO_REG_IOCTRL_SD0_CLK 0x0300_101C
SPI0_SCK	O	PAD_MIPI_TXM2	6	FMUX_GPIO_REG_IOCTRL_PAD_MIPI_TXM2 0x0300_11A4
SPI0_SDI	I	SD0_D0	2	FMUX_GPIO_REG_IOCTRL_SD0_D0 0x0300_1024
SPI0_SDI	I	PAD_MIPI_TXP1	6	FMUX_GPIO_REG_IOCTRL_PAD_MIPI_TXP1 0x0300_11B0
SPI0_SDO	I/O	SD0_CMD	2	FMUX_GPIO_REG_IOCTRL_SD0_CMD 0x0300_1020
SPI0_SDO	I/O	PAD_MIPI_TXM1	6	FMUX_GPIO_REG_IOCTRL_PAD_MIPI_TXM1 0x0300_11AC
SPI1_CS_X	O	MUX_SPI1_CS	6	FMUX_GPIO_REG_IOCTRL_MUX_SPI1_CS 0x0300_111C
SPI1_CS_X	O	PAD_ETH_RXM___EPHY_TXP	6	FMUX_GPIO_REG_IOCTRL_PAD_ETH_RXM 0x0300_1130
SPI1_SCK	O	MUX_SPI1_SCK	6	FMUX_GPIO_REG_IOCTRL_MUX_SPI1_SCK 0x0300_1120
SPI1_SCK	O	PAD_ETH_RXP___EPHY_TXN	6	FMUX_GPIO_REG_IOCTRL_PAD_ETH_RXP 0x0300_112C
SPI1_SDI	I	MUX_SPI1_MISO	6	FMUX_GPIO_REG_IOCTRL_MUX_SPI1_MISO 0x0300_1114
SPI1_SDI	I	PAD_ETH_TXM___EPHY_RXP	6	FMUX_GPIO_REG_IOCTRL_PAD_ETH_TXM 0x0300_1128
SPI1_SDO	I/O	MUX_SPI1_MOSI	6	FMUX_GPIO_REG_IOCTRL_MUX_SPI1_MOSI 0x0300_1118

To be continued ……

Table 10.27: SPI (continued)

Signal Name	Direction	PinName	Function Number	Function select register
SPI1_SDO	I/O	PAD _ETH_TX P___E PHY_RXN	6	FMUX_GPIO_R EG_IOCTRL_PAD_ETH_TXP 0x0300_1124
SPI2_CS_X	O	SD1_D3	1	FMUX_G PIO_REG_IOCTRL_SD1_D3 0x0300_10D0
SPI2_SCK	O	SD1_CLK	1	FMUX_GP IO_REG_IOCTRL_SD1_CLK 0x0300_10E4
SPI2_SDI	I	SD1_D0	1	FMUX_G PIO_REG_IOCTRL_SD1_D0 0x0300_10DC
SPI2_SDO	I/O	SD1_CMD	1	FMUX_GP IO_REG_IOCTRL_SD1_CMD 0x0300_10E0

10.1.1.25 UART

Table 10.28: UART

Signal Name	Direction	PinName	Function Number	Function select register
UART0_RX	I	U ART0_RX	0	FMUX_GPI O_REG_IOCTRL_UART0_RX 0x0300_1044
UART0_TX	O	U ART0_TX	0	FMUX_GPI O_REG_IOCTRL_UART0_TX 0x0300_1040
UART1_CTS	I	JTAG _CPU_TCK	4	FMUX_GPIO_RE G_IOCTRL_JTAG_CPU_TCK 0x0300_1068
UART1_RTS	O	JTAG _CPU_TMS	4	FMUX_GPIO_RE G_IOCTRL_JTAG_CPU_TMS 0x0300_1064
UART1_RX	I	SD0_D2	4	FMUX_G PIO_REG_IOCTRL_SD0_D2 0x0300_102C
UART1_RX	I	U ART0_RX	4	FMUX_GPI O_REG_IOCTRL_UART0_RX 0x0300_1044
UART1_RX	I	PWR _BUT-TON1	4	FMUX_GPIO_R EG_IOCTRL_PWR_BUTTON1 0x0300_1098

continues on next page

Table 10.28 – continued from previous page

Signal Name	Direction	PinName	Function Number	Function select register
UART1_RX	I	JTAG _CPU_TCK	6	FMUX_GPIO_RE G_IOCTRL_JTAG_CPU_TCK 0x0300_1068
UART1_RX	I	I IC0_SDA	1	FMUX_GPI O_REG_IOCTRL_IIC0_SDA 0x0300_1074
UART1_TX	O	SD0_D1	4	FMUX_G PIO_REG_IOCTRL_SD0_D1 0x0300_1028
UART1_TX	O	U ART0_TX	4	FMUX_GPI O_REG_IOCTRL_UART0_TX 0x0300_1040
UART1_TX	O	PWR _WAKEUP0	4	FMUX_GPIO_R EG_IOCTRL_PWR_WAKEUP0 0x0300_1090
UART1_TX	O	JTAG _CPU_TMS	6	FMUX_GPIO_RE G_IOCTRL_JTAG_CPU_TMS 0x0300_1064
UART1_TX	O	I IC0_SCL	1	FMUX_GPI O_REG_IOCTRL_IIC0_SCL 0x0300_1070
UART2_RX	I	I IC0_SDA	2	FMUX_GPI O_REG_IOCTRL_IIC0_SDA 0x0300_1074
UART2_RX	I	SD1_D1	2	FMUX_G PIO_REG_IOCTRL_SD1_D1 0x0300_10D8
UART2_RX	I	PW R_GPIO1	1	FMUX_GPIO _REG_IOCTRL_PWR_GPIO1 0x0300_10A8
UART2_TX	O	I IC0_SCL	2	FMUX_GPI O_REG_IOCTRL_IIC0_SCL 0x0300_1070
UART2_TX	O	SD1_D2	2	FMUX_G PIO_REG_IOCTRL_SD1_D2 0x0300_10D4
UART2_TX	O	PW R_GPIO0	1	FMUX_GPIO _REG_IOCTRL_PWR_GPIO0 0x0300_10A4
UART3_CTS	I	MUX _SPI1_CS	1	FMUX_GPIO_R EG_IOCTRL_MUX_SPI1_CS 0x0300_111C
UART3_CTS	I	PAD _ETH_RX M___E PHY_TXP	1	FMUX_GPIO_R EG_IOCTRL_PAD_ETH_RXM 0x0300_1130
UART3_CTS	I	SD1_D3	5	FMUX_G PIO_REG_IOCTRL_SD1_D3 0x0300_10D0

continues on next page

Table 10.28 – continued from previous page

Signal Name	Direction	PinName	Function Number	Function select register
UART3_RTS	O	MUX_SPI1_MISO	1	FMUX_GPIO_REG_IOCTRL_MUX_SPI1_MISO 0x0300_1114
UART3_RTS	O	PAD_ETH_TXM___EPHY_RXP	1	FMUX_GPIO_REG_IOCTRL_PAD_ETH_TXM 0x0300_1128
UART3_RTS	O	SD1_D0	5	FMUX_GPIO_REG_IOCTRL_SD1_D0 0x0300_10DC
UART3_RX	I	SD0_D3	4	FMUX_GPIO_REG_IOCTRL_SD0_D3 0x0300_1030
UART3_RX	I	MUX_SPI1_MOSI	1	FMUX_GPIO_REG_IOCTRL_MUX_SPI1_MOSI 0x0300_1118
UART3_RX	I	PAD_ETH_TXP___EPHY_RXN	1	FMUX_GPIO_REG_IOCTRL_PAD_ETH_TXP 0x0300_1124
UART3_RX	I	SD1_D1	5	FMUX_GPIO_REG_IOCTRL_SD1_D1 0x0300_10D8
UART3_TX	O	SD0_D0	4	FMUX_GPIO_REG_IOCTRL_SD0_D0 0x0300_1024
UART3_TX	O	MUX_SPI1_SCK	1	FMUX_GPIO_REG_IOCTRL_MUX_SPI1_SCK 0x0300_1120
UART3_TX	O	PAD_ETH_RXP___EPHY_TXN	1	FMUX_GPIO_REG_IOCTRL_PAD_ETH_RXP 0x0300_112C
UART3_TX	O	SD1_D2	5	FMUX_GPIO_REG_IOCTRL_SD1_D2 0x0300_10D4

10.1.1.26 No-die domain UART

Table 10.29: No-die domain UART

Signal Name	Direction	PinName	Function Number	Function select register
PWR_UART0_RX	I	PWR_GPIO0	2	FMUX_GPIO_REG_IOCTRL_PWR_GPIO0 0x0300_10A4
PWR_UART0_TX	O	PWR_WAKEUP0	2	FMUX_GPIO_REG_IOCTRL_PWR_WAKEUP0 0x0300_1090

10.1.1.27 Wiegand

Table 10.30: Wiegand

Signal Name	Direction	PinName	Function Number	Function select register
WG0_D0	I/O	SD0_D0	6	FMUX_G PIO_REG_IOCTRL_SD0_D0 0x0300_1024
WG0_D0	I/O	I IC0_SCL	5	FMUX_GPI O_REG_IOCTRL_IIC0_SCL 0x0300_1070
WG0_D1	I/O	SD0_D1	6	FMUX_G PIO_REG_IOCTRL_SD0_D1 0x0300_1028
WG0_D1	I/O	I IC0_SDA	5	FMUX_GPI O_REG_IOCTRL_IIC0_SDA 0x0300_1074
WG1_D0	I/O	SD0_D2	6	FMUX_G PIO_REG_IOCTRL_SD0_D2 0x0300_102C
WG1_D0	I/O	I IC0_SDA	6	FMUX_GPI O_REG_IOCTRL_IIC0_SDA 0x0300_1074
WG1_D1	I/O	SD0_D3	6	FMUX_G PIO_REG_IOCTRL_SD0_D3 0x0300_1030
WG1_D1	I/O	AUX0	6	FMUX _GPIO_REG_IOCTRL_AUX0 0x0300_1078
WG2_D0	I/O	PWR _WAKEUP0	7	FMUX_GPIO_R EG_IOCTRL_PWR_WAKEUP0 0x0300_1090
WG2_D1	I/O	PWR _BUT-TON1	7	FMUX_GPIO_R EG_IOCTRL_PWR_BUTTON1 0x0300_1098

10.1.1.28 GPIO

Table 10.31: GPIO

Signal Name	Direction	PinName	Function Number	Function select register
XGPIOA[7]	I/O	SD0_CLK	3	FMUX_GP IO_REG_IOCTRL_SD0_CLK 0x0300_101C
XGPIOA[8]	I/O	SD0_CMD	3	FMUX_GP IO_REG_IOCTRL_SD0_CMD 0x0300_1020
XGPIOA[9]	I/O	SD0_D0	3	FMUX_G PIO_REG_IOCTRL_SD0_D0 0x0300_1024

continues on next page

Table 10.31 – continued from previous page

Signal Name	Direction	PinName	Function Number	Function select register
XGPIOA[10]	I/O	SD0_D1	3	FMUX_G PIO_REG_IOCTRL_SD0_D1 0x0300_1028
XGPIOA[11]	I/O	SD0_D2	3	FMUX_G PIO_REG_IOCTRL_SD0_D2 0x0300_102C
XGPIOA[12]	I/O	SD0_D3	3	FMUX_G PIO_REG_IOCTRL_SD0_D3 0x0300_1030
XGPIOA[13]	I/O	SD0_CD	3	FMUX_G PIO_REG_IOCTRL_SD0_CD 0x0300_1034
XGPIOA[14]	I/O	SD0 _PWR_EN	3	FMUX_GPIO _REG_IOCTRL_SD0_PWR_EN 0x0300_1038
XGPIOA[15]	I/O	SPK_EN	3	FMUX_G PIO_REG_IOCTRL_SPK_EN 0x0300_103C
XGPIOA[16]	I/O	U ART0_TX	3	FMUX_GPI O_REG_IOCTRL_UART0_TX 0x0300_1040
XGPIOA[17]	I/O	U ART0_RX	3	FMUX_GPI O_REG_IOCTRL_UART0_RX 0x0300_1044
XGPIOA[18]	I/O	JTAG _CPU_TCK	3	FMUX_GPIO_RE G_IOCTRL_JTAG_CPU_TCK 0x0300_1068
XGPIOA[19]	I/O	JTAG _CPU_TMS	3	FMUX_GPIO_RE G_IOCTRL_JTAG_CPU_TMS 0x0300_1064
XGPIOA[22]	I/O	E MMC_CLK	3	FMUX_GPI O_REG_IOCTRL_EMMC_CLK 0x0300_1050
XGPIOA[23]	I/O	E MMC_CMD	3	FMUX_GPI O_REG_IOCTRL_EMMC_CMD 0x0300_105C
XGPIOA[24]	I/O	EM MC_DAT1	3	FMUX_GPIO _REG_IOCTRL_EMMC_DAT1 0x0300_1060
XGPIOA[25]	I/O	EM MC_DAT0	3	FMUX_GPIO _REG_IOCTRL_EMMC_DAT0 0x0300_1054
XGPIOA[26]	I/O	EM MC_DAT2	3	FMUX_GPIO _REG_IOCTRL_EMMC_DAT2 0x0300_104C
XGPIOA[27]	I/O	EM MC_DAT3	3	FMUX_GPIO _REG_IOCTRL_EMMC_DAT3 0x0300_1058

continues on next page

Table 10.31 – continued from previous page

Signal Name	Direction	PinName	Function Number	Function select register
XGPIOA[28]	I/O	I IC0_SCL	3	FMUX_GPI O_REG_IOCTRL_IIC0_SCL 0x0300_1070
XGPIOA[29]	I/O	I IC0_SDA	3	FMUX_GPI O_REG_IOCTRL_IIC0_SDA 0x0300_1074
XGPIOA[30]	I/O	AUX0	3	FMUX _GPIO_REG_IOCTRL_AUX0 0x0300_1078
XGPIOB[0]	I/O	PW M0_BUCK	3	FMUX_GPIO _REG_IOCTRL_PWM0_BUCK 0x0300_10EC
XGPIOB[10]	I/O	MUX _SPI1_CS	3	FMUX_GPIO_R EG_IOCTRL_MUX_SPI1_CS 0x0300_111C
XGPIOB[23]	I/O	GPIO_RT X___E PHY_RTX	3	FMUX_GPI O_REG_IOCTRL_GPIO_RTX 0x0300_11CC
XGPIOB[24]	I/O	PAD _ETH_TX M___E PHY_RXP	3	FMUX_GPIO_R EG_IOCTRL_PAD_ETH_TXM 0x0300_1128
XGPIOB[25]	I/O	PAD _ETH_TX P___E PHY_RXN	3	FMUX_GPIO_R EG_IOCTRL_PAD_ETH_TXP 0x0300_1124
XGPIOB[26]	I/O	PAD _ETH_RX M___E PHY_TXP	3	FMUX_GPIO_R EG_IOCTRL_PAD_ETH_RXM 0x0300_1130
XGPIOB[27]	I/O	PAD _ETH_RX P___E PHY_TXN	3	FMUX_GPIO_R EG_IOCTRL_PAD_ETH_RXP 0x0300_112C
XGPIOB[3]	I/O	ADC1	3	FMUX _GPIO_REG_IOCTRL_ADC1 0x0300_10F8
XGPIOB[6]	I/O	USB_V BUS_DET	3	FMUX_GPIO_RE G_IOCTRL_USB_VBUS_DET 0x0300_1108
XGPIOB[7]	I/O	MUX_SP I1_MOSI	3	FMUX_GPIO_REG _IOC-TRL_MUX_SPI1_MOSI 0x0300_1118
XGPIOB[8]	I/O	MUX_SP I1_MISO	3	FMUX_GPIO_REG _IOC-TRL_MUX_SPI1_MISO 0x0300_1114
XGPIOB[9]	I/O	MUX_S PI1_SCK	3	FMUX_GPIO_RE G_IOCTRL_MUX_SPI1_SCK 0x0300_1120
XGPIOC[2]	I/O	PAD_M IPIRX4N	3	FMUX_GPIO_RE G_IOCTRL_PAD_MIPIRX4N 0x0300_116C

continues on next page

Table 10.31 – continued from previous page

Signal Name	Direction	PinName	Function Number	Function select register
XGPIOC[3]	I/O	PAD_MIPIRX4P	3	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX4P 0x0300_1170
XGPIOC[4]	I/O	PAD_MIPIRX3N	3	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX3N 0x0300_1174
XGPIOC[5]	I/O	PAD_MIPIRX3P	3	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX3P 0x0300_1178
XGPIOC[6]	I/O	PAD_MIPIRX2N	3	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX2N 0x0300_117C
XGPIOC[7]	I/O	PAD_MIPIRX2P	3	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX2P 0x0300_1180
XGPIOC[8]	I/O	PAD_MIPIRX1N	3	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX1N 0x0300_1184
XGPIOC[9]	I/O	PAD_MIPIRX1P	3	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX1P 0x0300_1188
XGPIOC[10]	I/O	PAD_MIPIRX0N	3	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX0N 0x0300_118C
XGPIOC[11]	I/O	PAD_MIPIRX0P	3	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX0P 0x0300_1190
XGPIOC[12]	I/O	PAD_MIPI_TXM0	3	FMUX_GPIO_REG_IOCTRL_PAD_MIPI_TXM0 0x0300_11B4
XGPIOC[13]	I/O	PAD_MIPI_TXP0	3	FMUX_GPIO_REG_IOCTRL_PAD_MIPI_TXP0 0x0300_11B8
XGPIOC[14]	I/O	PAD_MIPI_TXM1	3	FMUX_GPIO_REG_IOCTRL_PAD_MIPI_TXM1 0x0300_11AC
XGPIOC[15]	I/O	PAD_MIPI_TXP1	3	FMUX_GPIO_REG_IOCTRL_PAD_MIPI_TXP1 0x0300_11B0
XGPIOC[16]	I/O	PAD_MIPI_TXM2	3	FMUX_GPIO_REG_IOCTRL_PAD_MIPI_TXM2 0x0300_11A4
XGPIOC[17]	I/O	PAD_MIPI_TXP2	3	FMUX_GPIO_REG_IOCTRL_PAD_MIPI_TXP2 0x0300_11A8
XGPIOC[23]	I/O	PAD_AUD_AINL_MIC	3	FMUX_GPIO_REG_IOCTRL_PAD_AUD_AINL_MIC 0x0300_11BC

continues on next page

Table 10.31 – continued from previous page

Signal Name	Direction	PinName	Function Number	Function select register
XGPIOC[24]	I/O	PAD_AUD_AOUTR	3	FMUX_GPIO_REG_IOCTRL_PAD_AUD_AOUTR 0x0300_11C8

10.1.1.29 No die domain GPIO

Table 10.32: No die domain GPIO

Signal Name	Direction	PinName	Function Number	Function select register
PWR_GPIO[0]	I/O	PWR_GPIO0	0	FMUX_GPIO_REG_IOCTRL_PWR_GPIO0 0x0300_10A4
PWR_GPIO[3]	I/O	PWR_SEQ1	3	FMUX_GPIO_REG_IOCTRL_PWR_SEQ1 0x0300_1084
PWR_GPIO[4]	I/O	PWR_SEQ2	3	FMUX_GPIO_REG_IOCTRL_PWR_SEQ2 0x0300_1088
PWR_GPIO[6]	I/O	PWR_WAKEUP0	3	FMUX_GPIO_REG_IOCTRL_PWR_WAKEUP0 0x0300_1090
PWR_GPIO[8]	I/O	PWR_BUTTON1	3	FMUX_GPIO_REG_IOCTRL_PWR_BUTTON1 0x0300_1098
PWR_GPIO[18]	I/O	SD1_D3	3	FMUX_GPIO_REG_IOCTRL_SD1_D3 0x0300_10D0
PWR_GPIO[19]	I/O	SD1_D2	3	FMUX_GPIO_REG_IOCTRL_SD1_D2 0x0300_10D4
PWR_GPIO[20]	I/O	SD1_D1	3	FMUX_GPIO_REG_IOCTRL_SD1_D1 0x0300_10D8
PWR_GPIO[21]	I/O	SD1_D0	3	FMUX_GPIO_REG_IOCTRL_SD1_D0 0x0300_10DC
PWR_GPIO[22]	I/O	SD1_CMD	3	FMUX_GPIO_REG_IOCTRL_SD1_CMD 0x0300_10E0
PWR_GPIO[23]	I/O	SD1_CLK	3	FMUX_GPIO_REG_IOCTRL_SD1_CLK 0x0300_10E4
PWR_GPIO[24]	I/O	GPIO_ZQ__PAD_ZQ	3	FMUX_GPIO_REG_IOCTRL_GPIO_ZQ 0x0300_11D0

10.1.1.30 Debug

Table 10.33: Debug

Signal Name	Direction	PinName	Function Number	Function select register
DBG[0]	O	SD0_CLK	7	FMUX_GP IO_REG_IOCTRL_SD0_CLK 0x0300_101C
DBG[1]	O	SD0_CMD	7	FMUX_GP IO_REG_IOCTRL_SD0_CMD 0x0300_1020
DBG[10]	O	I IC0_SCL	7	FMUX_GPI O_REG_IOCTRL_IIC0_SCL 0x0300_1070
DBG[10]	O	PAD_M IPIRX0N	7	FMUX_GPIO_RE G_IOCTRL_PAD_MIPIRX0N 0x0300_118C
DBG[11]	O	I IC0_SDA	7	FMUX_GPI O_REG_IOCTRL_IIC0_SDA 0x0300_1074
DBG[11]	O	PAD_M IPIRX0P	7	FMUX_GPIO_RE G_IOCTRL_PAD_MIPIRX0P 0x0300_1190
DBG[12]	O	AUX0	7	FMUX _GPIO_REG_IOCTRL_AUX0 0x0300_1078
DBG[12]	O	PAD_MI PI_TXM0	7	FMUX_GPIO_REG _IOCTRL_PAD_MIPI_TXM0 0x0300_11B4
DBG[13]	O	MUX_SP I1_MOSI	7	FMUX_GPIO_REG _IOCTRL_MUX_SPI1_MOSI 0x0300_1118
DBG[13]	O	PAD_MI PI_TXP0	7	FMUX_GPIO_REG _IOCTRL_PAD_MIPI_TXP0 0x0300_11B8
DBG[14]	O	MUX_SP I1_MISO	7	FMUX_GPIO_REG _IOCTRL_MUX_SPI1_MISO 0x0300_1114
DBG[14]	O	PAD_MI PI_TXM1	7	FMUX_GPIO_REG _IOCTRL_PAD_MIPI_TXM1 0x0300_11AC
DBG[15]	O	MUX_S PI1_SCK	7	FMUX_GPIO_RE G_IOCTRL_MUX_SPI1_SCK 0x0300_1120
DBG[15]	O	PAD_MI PI_TXP1	7	FMUX_GPIO_REG _IOCTRL_PAD_MIPI_TXP1 0x0300_11B0
DBG[16]	O	MUX _SPI1_CS	7	FMUX_GPIO_R EG_IOCTRL_MUX_SPI1_CS 0x0300_111C
DBG[2]	O	SD0_D0	7	FMUX_G PIO_REG_IOCTRL_SD0_D0 0x0300_1024
DBG[3]	O	SD0_D1	7	FMUX_G PIO_REG_IOCTRL_SD0_D1 0x0300_1028
DBG[4]	O	SD0_D2	7	FMUX_G PIO_REG_IOCTRL_SD0_D2 0x0300_102C
DBG[5]	O	SD0_D3	7	FMUX_G PIO_REG_IOCTRL_SD0_D3 0x0300_1030
DBG[6]	O	U ART0_TX	7	FMUX_GPI O_REG_IOCTRL_UART0_TX 0x0300_1040
DBG[6]	O	PAD_M IPIRX2N	7	FMUX_GPIO_RE G_IOCTRL_PAD_MIPIRX2N 0x0300_117C
DBG[7]	O	U ART0_RX	7	FMUX_GPI O_REG_IOCTRL_UART0_RX 0x0300_1044
DBG[7]	O	PAD_M IPIRX2P	7	FMUX_GPIO_RE G_IOCTRL_PAD_MIPIRX2P 0x0300_1180
DBG[8]	O	PAD_M IPIRX1N	7	FMUX_GPIO_RE G_IOCTRL_PAD_MIPIRX1N 0x0300_1184
DBG[9]	O	PAD_M IPIRX1P	7	FMUX_GPIO_RE G_IOCTRL_PAD_MIPIRX1P 0x0300_1188

10.1.1.31 Others

Table 10.34: Others

Signal Name	Direction	PinName	Function Number	Function select register
MUX_SPI1_CS	I/O	PAD_MIPIRX4P	7	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX4P 0x0300_1170
MUX_SPI1_MISO	I/O	PAD_MIPIRX3N	7	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX3N 0x0300_1174
MUX_SPI1_MOSI	I/O	PAD_MIPIRX3P	7	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX3P 0x0300_1178
MUX_SPI1_SCK	I/O	PAD_MIPIRX4N	7	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX4N 0x0300_116C
PKG_TYPE0	I	PKG_TYPE0	0	FMUX_GPIO_REG_IOCTRL_PKG_TYPE0 0x0300_1104
PKG_TYPE1	I	PKG_TYPE1	0	FMUX_GPIO_REG_IOCTRL_PKG_TYPE1 0x0300_110C
PKG_TYPE2	I	PKG_TYPE2	0	FMUX_GPIO_REG_IOCTRL_PKG_TYPE2 0x0300_1110

10.1.2 FMUX Registers Description

Table 10.35: FMUX Registers Configuration Information

Pin Num	Pin Name	Function_select_register	fmux_default	Description
6	SD0_CLK	FMUX_GPIO_REG_IOCTRL_SD0_CLK 0x0300_101C	0x0	IO SD0_CLK function select : • 0 : SDIO0_CLK (default) • 1 : IIC1_SDA • 2 : SPI0_SCK • 3 : XGPIOA[7] • 5 : PWM[15] • 6 : EPHY_LNK_LED • 7 : DBG[0] • Others : Reserved

continues on next page

Table 10.35 – continued from previous page

Pin Num	Pin Name	Function_select_register	fmux_default	Description
7	SD0_CMD	FMUX_GPIO_REG_IOCTRL_SD0_CMD 0x0300_1020	0x0	IO SD0_CMD function select : • 0 : SDIO0_CMD (default) • 1 : IIC1_SCL • 2 : SPI0_SDO • 3 : XGPIOA[8] • 5 : PWM[14] • 6 : EPHY_SPD_LED • 7 : DBG[1] • Others : Reserved
8	SD0_D0	FMUX_GPIO_REG_IOCTRL_SD0_D0 0x0300_1024	0x0	IO SD0_D0 function select: • 0 : SDIO0_D[0] (default) • 1 : CAM_MCLK1 • 2 : SPI0_SDI • 3 : XGPIOA[9] • 4 : UART3_TX • 5 : PWM[13] • 6 : WG0_D0 • 7 : DBG[2] • Others : Reserved
10	SD0_D1	FMUX_GPIO_REG_IOCTRL_SD0_D1 0x0300_1028	0x0	IO SD0_D1 function select: • 0 : SDIO0_D[1] (default) • 1 : IIC1_SDA • 2 : AUX0 • 3 : XGPIOA[10] • 4 : UART1_TX • 5 : PWM[12] • 6 : WG0_D1 • 7 : DBG[3] • Others : Reserved
11	SD0_D2	FMUX_GPIO_REG_IOCTRL_SD0_D2 0x0300_102C	0x0	IO SD0_D2 function select: • 0 : SDIO0_D[2] (default) • 1 : IIC1_SCL • 2 : AUX1 • 3 : XGPIOA[11] • 4 : UART1_RX • 5 : PWM[11] • 6 : WG1_D0 • 7 : DBG[4] • Others : Reserved

continues on next page

Table 10.35 – continued from previous page

Pin Num	Pin Name	Function_select_register	fmux_default	Description
12	SD0_D3	FMUX_GPIO_REG_IOCTRL_SD0_D3 0x0300_1030	0x0	IO SD0_D3 function select: • 0 : SDIO0_D[3] (default) • 1 : CAM_MCLK0 • 2 : SPI0_CS_X • 3 : XGPIOA[12] • 4 : UART3_RX • 5 : PWM[10] • 6 : WG1_D1 • 7 : DBG[5] • Others : Reserved
14	SD0_CD	FMUX_GPIO_REG_IOCTRL_SD0_CD 0x0300_1034	0x0	IO SD0_CD function select: • 0 : SDIO0_CD (default) • 3 : XGPIOA[13] • Others : Reserved
15	SD 0_PW R_EN	FMUX_GPIO _REG_IOCTRL_SD0_PWR_EN 0x0300_1038	0x3	IO SD0_PWR_EN function select : • 0 : SDIO0_PWR_EN • 3 : XGPIOA[14] (default) • Others : Reserved
17	SPK_EN	FMUX_GPIO_REG_IOCTRL_SPK_EN 0x0300_103C	0x3	IO SPK_EN function select: • 3 : XGPIOA[15] (default) • Others : Reserved
18	UART0_TX	FMUX_GP IO_REG_IOCTRL_UART0_TX 0x0300_1040	0x0	IO UART0_TX function select : • 0 : UART0_TX (default) • 1 : CAM_MCLK1 • 2 : PWM[4] • 3 : XGPIOA[16] • 4 : UART1_TX • 7 : DBG[6] • Others : Reserved
19	UART0_RX	FMUX_GPIO_REG_IOCTRL_UART0_RX 0x0300_1044	0x0	IO UART0_RX function select : • 0 : UART0_RX (default) • 1 : CAM_MCLK0 • 2 : PWM[5] • 3 : XGPIOA[17] • 4 : UART1_RX • 5 : AUX0 • 7 : DBG[7] • Others : Reserved

continues on next page

Table 10.35 – continued from previous page

Pin Num	Pin Name	Function_select_register	fmux_default	Description
20	EMMC_DAT2	FMUX_GPIO_REG_IOCTRL_EMMC_DAT2 0x0300_104C	0x1	IO EMMC_DAT2 function select : • 0 : EMMC_DAT[2] • 1 : SPINOR_HOLD_X(default) • 2 : SPINAND_HOLD • 3 : XGPIOA[26] • Others : Reserved
21	EMMC_CLK	FMUX_GPIO_REG_IOCTRL_EMMC_CLK 0x0300_1050	0x1	IO EMMC_CLK function select : • 0 : EMMC_CLK • 1 : SPINOR_SCK (default) • 2 : SPINAND_CLK • 3 : XGPIOA[22] • Others : Reserved
22	EMMC_DAT0	FMUX_GPIO_REG_IOCTRL_EMMC_DAT0 0x0300_1054	0x1	IO EMMC_DAT0 function select : • 0 : EMMC_DAT[0] • 1 : SPINOR_MOSI(default) • 2 : SPINAND_MOSI • 3 : XGPIOA[25] • Others : Reserved
23	EMMC_DAT3	FMUX_GPIO_REG_IOCTRL_EMMC_DAT3 0x0300_1058	0x1	IO EMMC_DAT3 function select : • 0 : EMMC_DAT[3] • 1 : SPINOR_WP_X (default) • 2 : SPINAND_WP • 3 : XGPIOA[27] • Others : Reserved
24	EMMC_CMD	FMUX_GPIO_REG_IOCTRL_EMMC_CMD 0x0300_105C	0x1	IO EMMC_CMD function select : • 0 : EMMC_CMD • 1 : SPINOR_MISO (default) • 2 : SPINAND_MISO • 3 : XGPIOA[23] • Others : Reserved
25	EMMC_DAT1	FMUX_GPIO_REG_IOCTRL_EMMC_DAT1 0x0300_1060	0x1	IO EMMC_DAT1 function select : • 0 : EMMC_DAT[1] • 1 : SPINOR_CS_X (default) • 2 : SPINAND_CS • 3 : XGPIOA[24] • Others : Reserved

continues on next page

Table 10.35 – continued from previous page

Pin Num	Pin Name	Function_select_register	fmux_default	Description
26	JTAG_CPU_TMS_TMS	FMUX_GPIO_REG_IOCTRL_JTAG_CPU_TMS 0x0300_1064	0x0	IO JTAG_CPU_TMS function select : • 0 : CR_4WTMS (default) • 1 : CAM_MCLK0 • 2 : PWM[7] • 3 : XGPIOA[19] • 4 : UART1_RTS • 5 : AUX0 • 6 : UART1_TX • 7 : VO_D[28] • Others : Reserved
27	JTAG_CPU_TCK_TCK	FMUX_GPIO_REG_IOCTRL_JTAG_CPU_TCK 0x0300_1068	0x0	IO JTAG_CPU_TCK function select : • 0 : CR_4WTCK (default) • 1 : CAM_MCLK1 • 2 : PWM[6] • 3 : XGPIOA[18] • 4 : UART1_CTS • 5 : AUX1 • 6 : UART1_RX • 7 : VO_D[29] • Others : Reserved
28	IIC0_SCL	FMUX_GPIO_REG_IOCTRL_IIC0_SCL 0x0300_1070	0x0	IO IIC0_SCL function select : • 0 : CR_4WTDI (default) • 1 : UART1_TX • 2 : UART2_TX • 3 : XGPIOA[28] • 5 : WG0_D0 • 7 : DBG[10] • Others : Reserved
29	IIC0_SDA	FMUX_GPIO_REG_IOCTRL_IIC0_SDA 0x0300_1074	0x0	IO IIC0_SDA function select : • 0 : CR_4WTDO (default) • 1 : UART1_RX • 2 : UART2_RX • 3 : XGPIOA[29] • 5 : WG0_D1 • 6 : WG1_D0 • 7 : DBG[11] • Others : Reserved

continues on next page

Table 10.35 – continued from previous page

Pin Num	Pin Name	Function_select_register	fmux_default	Description
30	AUX0	FMUX_GPIO_REG_IOCTRL_AUX0 0x0300_1078	0x3	IO AUX0 function select : • 0 : AUX0 • 3 : XGPIOA[30] (default) • 4 : IIS1_MCLK • 5 : VO_D[31] • 6 : WG1_D1 • 7 : DBG[12] • Others : Reserved
38	PWR_VBAT_DET	FMUX_GPIO_REG_IOCTRL_PWR_VBAT_DET 0x0300_107C	0x0	IO PWR_VBAT_DET function select : • 0 : PWR_VBAT_DET (default) • Others : Reserved
39	PWR_RSTN	FMUX_GPIO_REG_IOCTRL_PWR_RSTN 0x0300_1080	0x0	IO PWR_RSTN function select : • 0 : PWR_RSTN (default) • Others : Reserved
40	PWR_SEQ1	FMUX_GPIO_REG_IOCTRL_PWR_SEQ1 0x0300_1084	0x0	IO PWR_SEQ1 function select : • 0 : PWR_SEQ1 (default) • 3 : PWR_GPIO[3] • Others : Reserved
41	PWR_SEQ2	FMUX_GPIO_REG_IOCTRL_PWR_SEQ2 0x0300_1088	0x0	IO PWR_SEQ2 function select : • 0 : PWR_SEQ2 (default) • 3 : PWR_GPIO[4] • Others : Reserved
43	PWR_WAKEUP0	FMUX_GPIO_REG_IOCTRL_PWR_WAKEUP0 0x0300_1090	0x0	IO PWR_WAKEUP0 function select : • 0 : PWR_WAKEUP0(default) • 1 : PWR_IR0 • 2 : PWR_UART0_TX • 3 : PWR_GPIO[6] • 4 : UART1_TX • 5 : IIC4_SCL • 6 : EPHY_LNK_LED • 7 : WG2_D0 • Others : Reserved

continues on next page

Table 10.35 – continued from previous page

Pin Num	Pin Name	Function_select_register	fmux_default	Description
44	PWR_BUTTON1	FMUX_GPIO_REG_IOCTRL_PWR_BUTTON1 0x0300_1098	0x0	IO PWR_BUTTON1 function select : • 0 : PWR_BUTTON1(default) • 3 : PWR_GPIO[8] • 4 : UART1_RX • 5 : IIC4_SDA • 6 : EPHY_SPD_LED • 7 : WG2_D1 • Others : Reserved
45	XTAL_XIN	FMUX_GPIO_REG_IOCTRL_XTAL_XIN 0x0300_10A0	0x0	IO XTAL_XIN function select : • 0 : PWR_XTAL_CLKIN(default) • Others : Reserved
47	PWR_GPIO0	FMUX_GPIO_REG_IOCTRL_PWR_GPIO0 0x0300_10A4	0x0	IO PWR_GPIO0 function select : • 0 : PWR_GPIO[0](default) • 1 : UART2_TX • 2 : PWR_UART0_RX • 4 : PWM[8] • Others : Reserved
48	PWR_GPIO1	FMUX_GPIO_REG_IOCTRL_PWR_GPIO1 0x0300_10A8	0x0	IO PWR_GPIO1 function select : • 0 : PWR_GPIO[1](default) • 1 : UART2_RX • 3 : EPHY_LNK_LED • 4 : PWM[9] • 5 : PWR_IIC_SCL • 6 : IIC2_SCL • 7 : CR_SDA0 • Others : Reserved
49	PWR_GPIO2	FMUX_GPIO_REG_IOCTRL_PWR_GPIO2 0x0300_10AC	0x0	IO PWR_GPIO2 function select : • 0 : PWR_GPIO[2](default) • 2 : PWR_SECTICK • 3 : EPHY_SPD_LED • 4 : PWM[10] • 5 : PWR_IIC_SDA • 6 : IIC2_SDA • 7 : CR_2WTCK • Others : Reserved

continues on next page

Table 10.35 – continued from previous page

Pin Num	Pin Name	Function_select_register	fmux_default	Description
51	SD1_D3	FMUX_GPIO_REG_IOCTRL_SD1_D3 0x0300_10D0	0x6	IO SD1_D3 function select : • 0 : PWR_SD1_D3_VO32 • 1 : SPI2_CS_X • 2 : IIC1_SCL • 3 : PWR_GPIO[18] • 4 : CAM_MCLK0 • 5 : UART3_CTS • 6 : PWR_SPINOR1_CS_X • (default) • 7 : PWM[4] • Others : Reserved
52	SD1_D2	FMUX_GPIO_REG_IOCTRL_SD1_D2 0x0300_10D4	0x6	IO SD1_D2 function select : • 0 : PWR_SD1_D2_VO33 • 1 : IIC1_SCL • 2 : UART2_TX • 3 : PWR_GPIO[19] • 4 : CAM_MCLK0 • 5 : UART3_TX • 6 : PWR_SPINOR1_HOLD_X • (default) • 7 : PWM[5] • Others : Reserved
53	SD1_D1	FMUX_GPIO_REG_IOCTRL_SD1_D1 0x0300_10D8	0x6	IO SD1_D1 function select : • 0 : PWR_SD1_D1_VO34 • 1 : IIC1_SDA • 2 : UART2_RX • 3 : PWR_GPIO[20] • 4 : CAM_MCLK1 • 5 : UART3_RX • 6 : PWR_SPINOR1_WP_X • (default) • 7 : PWM[6] • Others : Reserved
54	SD1_D0	FMUX_GPIO_REG_IOCTRL_SD1_D0 0x0300_10DC	0x6	IO SD1_D0 function select : • 0 : PWR_SD1_D0_VO35 • 1 : SPI2_SDI • 2 : IIC1_SDA • 3 : PWR_GPIO[21] • 4 : CAM_MCLK1 • 5 : UART3_RTS • 6 : PWR_SPINOR1_MISO • (default) • 7 : PWM[7] • Others : Reserved

continues on next page

Table 10.35 – continued from previous page

Pin Num	Pin Name	Function_select_register	fmux_default	Description
55	SD1_CMD	FMUX_GPIO_REG_IOCTRL_SD1_CMD 0x0300_10E0	0x6	IO SD1_CMD function select : • 0 : PWR_SD1_CMD_VO36 • 1 : SPI2_SDO • 2 : IIC3_SCL • 3 : PWR_GPIO[22] • 4 : CAM_VS0 • 5 : EPHY_LNK_LED • 6 : PWR_SPINOR1_MOSI • (default) • 7 : PWM[8] • Others : Reserved
56	SD1_CLK	FMUX_GPIO_REG_IOCTRL_SD1_CLK 0x0300_10E4	0x6	IO SD1_CLK function select : • 0 : PWR_SD1_CLK_VO37 • 1 : SPI2_SCK • 2 : IIC3_SDA • 3 : PWR_GPIO[23] • 4 : CAM_HS0 • 5 : EPHY_SPD_LED • 6 : PWR_SPINOR1_SCK • (default) • 7 : PWM[9] • Others : Reserved
58	PWM0_BUCK	FMUX_GPIO_REG_IOCTRL_PWM0_BUCK 0x0300_10EC	0x3	IO PWM0_BUCK function select : • 0 : PWM[0] • 3 : XGPIOB[0] (default) • Others : Reserved
59	ADC1	FMUX_GPIO_REG_IOCTRL_ADC1 0x0300_10F8	0x3	IO ADC1 function select : • 3 : XGPIOB[3] (default) • 4 : KEY_COL2 • Others : Reserved
60	USB_VBUS_DET	FMUX_GPIO_REG_IOCTRL_USB_VBUS_DET 0x0300_1108	0x0	IO USB_VBUS_DET function select : • 0 : USB_VBUS_DET • (default) • 3 : XGPIOB[6] • 4 : CAM_MCLK0 • 5 : CAM_MCLK1 • Others : Reserved

continues on next page

Table 10.35 – continued from previous page

Pin Num	Pin Name	Function_select_register	fmux_default	Description
62	PAD _ETH _TXP ___ EPHY _RXN	FMUX_GPIO _REG_IOCTRL_PAD_ETH_TXP 0x0300_1124	0x3	IO PAD_ETH_TXP function select : • 1 : UART3_RX • 2 : IIC1_SCL • 3 : XGPIOB[25] (default) • 4 : PWM[13] • 5 : CAM_MCLK0 • 6 : SPI1_SDO • 7 : IIS2_LRCK • Others : Reserved
63	PAD _ETH _TXM ___ EPHY _RXP	FMUX_GPIO _REG_IOCTRL_PAD_ETH_TXM 0x0300_1128	0x3	IO PAD_ETH_TXM function select : • 1 : UART3_RTS • 2 : IIC1_SDA • 3 : XGPIOB[24] (default) • 4 : PWM[12] • 5 : CAM_MCLK1 • 6 : SPI1_SDI • 7 : IIS2_BCLK • Others : Reserved
64	PAD _ETH _RXP ___ EPHY _TXN	FMUX_GPIO _REG_IOCTRL_PAD_ETH_RXP 0x0300_112C	0x3	IO PAD_ETH_RXP function select : • 1 : UART3_TX • 2 : CAM_MCLK1 • 3 : XGPIOB[27] (default) • 4 : PWM[15] • 5 : CAM_HS0 • 6 : SPI1_SCK • 7 : IIS2_DO • Others : Reserved
65	PAD _ETH _RXM ___ EPHY _TXP	FMUX_GPIO _REG_IOCTRL_PAD_ETH_RXM 0x0300_1130	0x3	IO PAD_ETH_RXM function select : • 1 : UART3_CTS • 2 : CAM_MCLK0 • 3 : XGPIOB[26] (default) • 4 : PWM[14] • 5 : CAM_VS0 • 6 : SPI1_CS_X • 7 : IIS2_DI • Others : Reserved

continues on next page

Table 10.35 – continued from previous page

Pin Num	Pin Name	Function_select_register	fmux_default	Description
72	PAD _MIPIRX4N	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX4N 0x0300_116C	0x0	IO PAD_MIPIRX4N function select : • 0 : CR_SCL0 (default) • 1 : VI0_CLK • 2 : VI1_D[13] • 3 : XGPIOC[2] • 4 : IIC1_SDA • 5 : CAM_MCLK0 • 6 : KEY_ROW0 • 7 : MUX_SPI1_SCK • Others : Reserved
73	PAD _MIPIRX4P	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX4P 0x0300_1170	0x0	IO PAD_MIPIRX4P function select : • 0 : CR_SDA0 (default) • 1 : VI0_D[0] • 2 : VI1_D[14] • 3 : XGPIOC[3] • 4 : IIC1_SCL • 5 : CAM_MCLK1 • 6 : KEY_ROW1 • 7 : MUX_SPI1_CS • Others : Reserved
74	PAD _MIPIRX3N	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX3N 0x0300_1174	0x3	IO PAD_MIPIRX3N function select : • 0 : CR_2WTMS • 1 : VI0_D[1] • 2 : VI1_D[15] • 3 : XGPIOC[4] (default) • 4 : CAM_MCLK0 • 7 : MUX_SPI1_MISO • Others : Reserved
75	PAD _MIPIRX3P	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX3P 0x0300_1178	0x3	IO PAD_MIPIRX3P function select : • 0 : CR_2WTCK • 1 : VI0_D[2] • 2 : VI1_D[16] • 3 : XGPIOC[5] (default) • 7 : MUX_SPI1_MOSI • Others : Reserved

continues on next page

Table 10.35 – continued from previous page

Pin Num	Pin Name	Function_select_register	fmux_ default	Description
76	PAD _MIPI RX2N	FMUX_GPIO_R EG_IOCTRL_PAD_MIPIRX2N 0x0300_117C	0x3	IO PAD_MIPIRX2N function select : • 1 : VI0_D[3] • 2 : VO_D[10] • 3 : XGPIOC[6] (default) • 4 : VI1_D[17] • 5 : IIC4_SCL • 7 : DBG[6] • Others : Reserved
77	PAD _MIPI RX2P	FMUX_GPIO_R EG_IOCTRL_PAD_MIPIRX2P 0x0300_1180	0x3	IO PAD_MIPIRX2P function select : • 1 : VI0_D[4] • 2 : VO_D[9] • 3 : XGPIOC[7] (default) • 4 : VI1_D[18] • 5 : IIC4_SDA • 7 : DBG[7] • Others : Reserved
78	PAD _MIPI RX1N	FMUX_GPIO_R EG_IOCTRL_PAD_MIPIRX1N 0x0300_1184	0x3	IO PAD_MIPIRX1N function select : • 1 : VI0_D[5] • 2 : VO_D[8] • 3 : XGPIOC[8] (default) • 6 : KEY_ROW3 • 7 : DBG[8] • Others : Reserved
79	PAD _MIPI RX1P	FMUX_GPIO_R EG_IOCTRL_PAD_MIPIRX1P 0x0300_1188	0x3	IO PAD_MIPIRX1P function select : • 1 : VI0_D[6] • 2 : VO_D[7] • 3 : XGPIOC[9] (default) • 4 : IIC1_SDA • 6 : KEY_ROW2 • 7 : DBG[9] • Others : Reserved
80	PAD _MIPI RX0N	FMUX_GPIO_R EG_IOCTRL_PAD_MIPIRX0N 0x0300_118C	0x3	IO PAD_MIPIRX0N function select : • 1 : VI0_D[7] • 2 : VO_D[6] • 3 : XGPIOC[10] (default) • 4 : IIC1_SCL • 5 : CAM_MCLK1 • 7 : DBG[10] • Others : Reserved

continues on next page

Table 10.35 – continued from previous page

Pin Num	Pin Name	Function_select_register	fmux_default	Description
81	PAD _MIPI RX0P	FMUX_GPIO_REG_IOCTRL_PAD_MIPIRX0P 0x0300_1190	0x3	IO PAD_MIPIRX0P function select : • 1 : VI0_D[8] • 2 : VO_D[5] • 3 : XGPIOC[11] (default) • 4 : CAM_MCLK0 • 7 : DBG[11] • Others : Reserved
83	PAD _MIPI _TXM2	FMUX_GPIO_REG_IOCTRL_PAD_MIPI_TXM2 0x0300_11A4	0x3	IO PAD_MIPI_TXM2 function select : • 0 : CR_SDA0 • 1 : VI0_D[13] • 2 : VO_D[0] • 3 : XGPIOC[16] (default) • 4 : IIC1_SDA • 5 : PWM[8] • 6 : SPI0_SCK • 7 : SD1_D2 • Others : Reserved
84	PAD _MIPI _TXP2	FMUX_GPIO_REG_IOCTRL_PAD_MIPI_TXP2 0x0300_11A8	0x3	IO PAD_MIPI_TXP2 function select : • 0 : CR_SCL0 • 1 : VI0_D[14] • 2 : VO_CLK0 • 3 : XGPIOC[17] (default) • 4 : IIC1_SCL • 5 : PWM[9] • 6 : SPI0_CS_X • 7 : SD1_D3 • Others : Reserved
85	PAD _MIPI _TXM1	FMUX_GPIO_REG_IOCTRL_PAD_MIPI_TXM1 0x0300_11AC	0x3	IO PAD_MIPI_TXM1 function select : • 0 : CR_2WTMS • 1 : VI0_D[11] • 2 : VO_D[2] • 3 : XGPIOC[14] (default) • 4 : IIC2_SDA • 5 : PWM[10] • 6 : SPI0_SDO • 7 : DBG[14] • Others : Reserved

continues on next page

Table 10.35 – continued from previous page

Pin Num	Pin Name	Function_select_register	fmux_default	Description
86	PAD _MIPI _TXP1	FMUX_GPIO_REG_IOCTRL_PAD_MIPI_TXP1 0x0300_11B0	0x3	IO PAD_MIPI_TXP1 function select : • 0 : CR_2WTCK • 1 : VI0_D[12] • 2 : VO_D[1] • 3 : XGPIOC[15] (default) • 4 : IIC2_SCL • 5 : PWM[11] • 6 : SPI0_SDI • 7 : DBG[15] • Others : Reserved
87	PAD _MIPI _TXM0	FMUX_GPIO_REG_IOCTRL_PAD_MIPI_TXM0 0x0300_11B4	0x3	IO PAD_MIPI_TXM0 function select : • 1 : VI0_D[9] • 2 : VO_D[4] • 3 : XGPIOC[12] (default) • 4 : CAM_MCLK1 • 5 : PWM[14] • 6 : CAM_VS0 • 7 : DBG[12] • Others : Reserved
88	PAD _MIPI _TXP0	FMUX_GPIO_REG_IOCTRL_PAD_MIPI_TXP0 0x0300_11B8	0x3	IO PAD_MIPI_TXP0 function select : • 1 : VI0_D[10] • 2 : VO_D[3] • 3 : XGPIOC[13] (default) • 4 : CAM_MCLK0 • 5 : PWM[15] • 6 : CAM_HS0 • 7 : DBG[13] • Others : Reserved
2	PAD _AUD _AINL _MIC	FMUX_GPIO_REG_IOCTRL_PAD_AUD_AINL_MIC 0x0300_11BC	0x3	IO PAD_AUD_AINL_MIC function select : • 3 : XGPIOC[23] (default) • 4 : IIS1_BCLK • 5 : IIS2_BCLK • Others : Reserved
4	PAD _AUD _AOUTR	FMUX_GPIO_REG_IOCTRL_PAD_AUD_AOUTR 0x0300_11C8	0x3	IO PAD_AUD_AOUTR function select : • 3 : XGPIOC[24] (default) • 4 : IIS1_DI • 5 : IIS2_DO • 6 : IIS1_DO • Others : Reserved

continues on next page

Table 10.35 – continued from previous page

Pin Num	Pin Name	Function_select_register	fmux_default	Description
67	GPIO _RTX ___ EPHY _RTX	FMUX_GPIO_REG_IOCTRL_GPIO_RTX 0x0300_11CC	0x3	IO GPIO_RTX function select : • 3 : XGPIOB[23] (default) • 4 : PWM[1] • 5 : CAM_MCLK0 • Others : Reserved
35	GPIO _ZQ ___ PAD _ZQ	FMUX_GPIO_REG_IOCTRL_GPIO_ZQ 0x0300_11D0	0x3	IO GPIO_ZQ function select : • 3 : PWR_GPIO[24](default) • 4 : PWM[2] • Others : Reserved
#N/A	PKG _TYPE0	FMUX_GPIO_REG_IOCTRL_PKG_TYPE0 0x0300_1104	0x0	IO PKG_TYPE0 function select : • 0 : PKG_TYPE0 (default) • Others : Reserved
#N/A	PKG _TYPE1	FMUX_GPIO_REG_IOCTRL_PKG_TYPE1 0x0300_110C	0x0	IO PKG_TYPE1 function select : • 0 : PKG_TYPE1 (default) • Others : Reserved
#N/A	PKG _TYPE2	FMUX_GPIO_REG_IOCTRL_PKG_TYPE2 0x0300_1110	0x0	IO PKG_TYPE2 function select : • 0 : PKG_TYPE2 (default) • Others : Reserved
#N/A	MUX _SPI1 _MISO	FMUX_GPIO_REG_IOCTRL_MUX_SPI1_MISO 0x0300_1114	0x3	IO MUX_SPI1_MISO function select : • 1 : UART3_RTS • 2 : IIC1_SDA • 3 : XGPIOB[8] (default) • 4 : PWM[9] • 5 : KEY_COL1 • 6 : SPI1_SDI • 7 : DBG[14] • Others : Reserved
#N/A	MUX _SPI1 _MOSI	FMUX_GPIO_REG_IOCTRL_MUX_SPI1_MOSI 0x0300_1118	0x3	IO MUX_SPI1_MOSI function select : • 1 : UART3_RX • 2 : IIC1_SCL • 3 : XGPIOB[7] (default) • 4 : PWM[8] • 5 : KEY_COL0 • 6 : SPI1_SDO • 7 : DBG[13] • Others : Reserved

continues on next page

Table 10.35 – continued from previous page

Pin Num	Pin Name	Function_select_register	fmux_default	Description
#N/A	MUX _SPI1 _CS	FMUX_GPIO _REG_IOCTRL_MUX_SPI1_CS 0x0300_111C	0x3	IO MUX_SPI1_CS function select : • 1 : UART3_CTS • 2 : CAM_MCLK0 • 3 : XGPIOB[10] (default) • 4 : PWM[11] • 5 : KEY_ROW3 • 6 : SPI1_CS_X • 7 : DBG[16] • Others : Reserved
#N/A	MUX _SPI1 _SCK	FMUX_GPIO_R EG_IOCTRL_MUX_SPI1_SCK 0x0300_1120	0x3	IO MUX_SPI1_SCK function select : • 1 : UART3_TX • 2 : CAM_MCLK1 • 3 : XGPIOB[9] (default) • 4 : PWM[10] • 5 : KEY_ROW2 • 6 : SPI1_SCK • 7 : DBG[15] • Others : Reserved

10.2 Pin Control (PINCTRL)

10.2.1 Pin Control Register Overview

Table 10.36: Pine Control Registers Overview

Pin-Num	Pin Name	IO_cfg_register	Address
6	SD0_CLK	IOBLK_G10_REG_SD0_CLK	0x0300_1A00
7	SD0_CMD	IOBLK_G10_REG_SD0_CMD	0x0300_1A04
8	SD0_D0	IOBLK_G10_REG_SD0_D0	0x0300_1A08
10	SD0_D1	IOBLK_G10_REG_SD0_D1	0x0300_1A0C
11	SD0_D2	IOBLK_G10_REG_SD0_D2	0x0300_1A10
12	SD0_D3	IOBLK_G10_REG_SD0_D3	0x0300_1A14
14	SD0_CD	IOBLK_G7_REG_SD0_CD	0x0300_1900
15	SD0_PWR_EN	IOBLK_G7_REG_SD0_PWR_EN	0x0300_1904
17	SPK_EN	IOBLK_G7_REG_SPK_EN	0x0300_1908
18	UART0_TX	IOBLK_G7_REG_UART0_TX	0x0300_190C
19	UART0_RX	IOBLK_G7_REG_UART0_RX	0x0300_1910
20	EMMC_DAT2	IOBLK_G7_REG_EMMC_DAT2	0x0300_1918
21	EMMC_CLK	IOBLK_G7_REG_EMMC_CLK	0x0300_191C
22	EMMC_DAT0	IOBLK_G7_REG_EMMC_DAT0	0x0300_1920
23	EMMC_DAT3	IOBLK_G7_REG_EMMC_DAT3	0x0300_1924
24	EMMC_CMD	IOBLK_G7_REG_EMMC_CMD	0x0300_1928
25	EMMC_DAT1	IOBLK_G7_REG_EMMC_DAT1	0x0300_192C
26	JTAG_CPU_TMS	IOBLK_G7_REG_JTAG_CPU_TMS	0x0300_1930
27	JTAG_CPU_TCK	IOBLK_G7_REG_JTAG_CPU_TCK	0x0300_1934
28	IIC0_SCL	IOBLK_G7_REG_IIC0_SCL	0x0300_193C
29	IIC0_SDA	IOBLK_G7_REG_IIC0_SDA	0x0300_1940
30	AUX0	IOBLK_G7_REG_AUX0	0x0300_1944
38	PWR_VBAT_DET	IOBLK_GRTC_REG_PWR_VBAT_DET	0x0502_7000
39	PWR_RSTN	IOBLK_GRTC_REG_PWR_RSTN	0x0502_7004
40	PWR_SEQ1	IOBLK_GRTC_REG_PWR_SEQ1	0x0502_7008
41	PWR_SEQ2	IOBLK_GRTC_REG_PWR_SEQ2	0x0502_700C
43	PWR_WAKEUP0	IOBLK_GRTC_REG_PWR_WAKEUP0	0x0502_7018
44	PWR_BUTTON1	IOBLK_GRTC_REG_PWR_BUTTON1	0x0502_7020
45	XTAL_XIN	IOBLK_GRTC_REG_XTAL_XIN	0x0502_7028
47	PWR_GPIO0	IOBLK_GRTC_REG_PWR_GPIO0	0x0502_702C
48	PWR_GPIO1	IOBLK_GRTC_REG_PWR_GPIO1	0x0502_7030
49	PWR_GPIO2	IOBLK_GRTC_REG_PWR_GPIO2	0x0502_7034
51	SD1_D3	IOBLK_GRTC_REG_SD1_D3	0x0502_7058
52	SD1_D2	IOBLK_GRTC_REG_SD1_D2	0x0502_705C
53	SD1_D1	IOBLK_GRTC_REG_SD1_D1	0x0502_7060
54	SD1_D0	IOBLK_GRTC_REG_SD1_D0	0x0502_7064
55	SD1_CMD	IOBLK_GRTC_REG_SD1_CMD	0x0502_7068
56	SD1_CLK	IOBLK_GRTC_REG_SD1_CLK	0x0502_706C
58	PWM0_BUCK	IOBLK_G1_REG_PWM0_BUCK	0x0300_1804
59	ADC1	IOBLK_G1_REG_ADC1	0x0300_1810
60	USB_VBUS_DET	IOBLK_G1_REG_USB_VBUS_DET	0x0300_1820
62	PAD_ETH_TXP___EPHY_RXN	#N/A	#N/A
63	PAD_ETH_TXM___EPHY_RXP	#N/A	#N/A

continues on next page

Table 10.36 – continued from previous page

Pin-Num	Pin Name	IO_cfg_register	Address
64	PAD_ETH_RXP___EPHY_TXN	#N/A	#N/A
65	PAD_ETH_RXM___EPHY_TXP	#N/A	#N/A
72	PAD_MIPIRX4N	IOBLK_G12_REG_PAD_MIPIRX4N	0x0300_1C38
73	PAD_MIPIRX4P	IOBLK_G12_REG_PAD_MIPIRX4P	0x0300_1C3C
74	PAD_MIPIRX3N	IOBLK_G12_REG_PAD_MIPIRX3N	0x0300_1C40
75	PAD_MIPIRX3P	IOBLK_G12_REG_PAD_MIPIRX3P	0x0300_1C44
76	PAD_MIPIRX2N	IOBLK_G12_REG_PAD_MIPIRX2N	0x0300_1C48
77	PAD_MIPIRX2P	IOBLK_G12_REG_PAD_MIPIRX2P	0x0300_1C4C
78	PAD_MIPIRX1N	IOBLK_G12_REG_PAD_MIPIRX1N	0x0300_1C50
79	PAD_MIPIRX1P	IOBLK_G12_REG_PAD_MIPIRX1P	0x0300_1C54
80	PAD_MIPIRX0N	IOBLK_G12_REG_PAD_MIPIRX0N	0x0300_1C58
81	PAD_MIPIRX0P	IOBLK_G12_REG_PAD_MIPIRX0P	0x0300_1C5C
83	PAD_MIPI_TXM2	IOBLK_G12_REG_PAD_MIPI_TXM2	0x0300_1C70
84	PAD_MIPI_TXP2	IOBLK_G12_REG_PAD_MIPI_TXP2	0x0300_1C74
85	PAD_MIPI_TXM1	IOBLK_G12_REG_PAD_MIPI_TXM1	0x0300_1C78
86	PAD_MIPI_TXP1	IOBLK_G12_REG_PAD_MIPI_TXP1	0x0300_1C7C
87	PAD_MIPI_TXM0	IOBLK_G12_REG_PAD_MIPI_TXM0	0x0300_1C80
88	PAD_MIPI_TXP0	IOBLK_G12_REG_PAD_MIPI_TXP0	0x0300_1C84
2	PAD_AUD_AINL_MIC	#N/A	#N/A
4	PAD_AUD_AOUTR	#N/A	#N/A
67	GPIO_RTX___EPHY_RTX	IOBLK_G12_REG_GPIO_RTX	0x0300_1C8C
35	GPIO_ZQ___PAD_ZQ	IOBLK_GRTC_REG_GPIO_ZQ	0x0502_70E0
#N/A	PKG_TYPE0	IOBLK_G1_REG_PKG_TYPE0	0x0300_181C
#N/A	PKG_TYPE1	IOBLK_G1_REG_PKG_TYPE1	0x0300_1824
#N/A	PKG_TYPE2	IOBLK_G1_REG_PKG_TYPE2	0x0300_1828
#N/A	MUX_SPI1_MISO	#N/A	#N/A
#N/A	MUX_SPI1_MOSI	#N/A	#N/A
#N/A	MUX_SPI1_CS	#N/A	#N/A
#N/A	MUX_SPI1_SCK	#N/A	#N/A

10.2.2 Pin Control Register Description

10.2.2.1 IOBLK_G1_REG_PWM0_BUCK

Table 10.37: IOBLK_G1_REG_PWM0_BUCK

Field Name	Bit	Default	FieldDescription
IOBLK_G1_REG_PWM0_BUCK_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_G1_REG_PWM0_BUCK_PD	3	0x0	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_G1_REG_PWM0_BUCK_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_G1_REG_PWM0_BUCK_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_G1_REG_PWM0_BUCK_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_G1_REG_PWM0_BUCK_ST1	9	0x0	Input Schmitt trigger strength control, bit 1
IOBLK_G1_REG_PWM0_BUCK_HE	10	0x0	Weak level holder (Bus holder) enable. 0=Disabled; 1=Enabled
IOBLK_G1_REG_PWM0_BUCK_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.2 IOBLK_G1_REG_ADC1

Table 10.38: IOBLK_G1_REG_ADC1

Field Name	Bit	Default	FieldDescription
IOBLK_G1_REG_ADC1_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_G1_REG_ADC1_PD	3	0x1	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_G1_REG_ADC1_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_G1_REG_ADC1_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_G1_REG_ADC1_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_G1_REG_ADC1_ST1	9	0x0	Input Schmitt trigger strength control, bit 1
IOBLK_G1_REG_ADC1_HE	10	0x0	Weak level holder (Bus holder) enable. 0=Disabled; 1=Enabled
IOBLK_G1_REG_ADC1_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.3 IOBLK_G1_REG_PKG_TYPE0

Table 10.39: IOBLK_G1_REG_PKG_TYPE0

Field Name	Bit	Default	FieldDescription
IOBLK_G1_REG_PKG_TYPE0_PU	2	0x1	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_G1_REG_PKG_TYPE0_PD	3	0x0	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_G1_REG_PKG_TYPE0_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_G1_REG_PKG_TYPE0_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_G1_REG_PKG_TYPE0_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_G1_REG_PKG_TYPE0_ST1	9	0x0	Input Schmitt trigger strength control, bit 1
IOBLK_G1_REG_PKG_TYPE0_HE	10	0x0	Weak level holder (Bus holder) enable. 0=Disabled; 1=Enabled
IOBLK_G1_REG_PKG_TYPE0_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.4 IOBLK_G1_REG_USB_VBUS_DET

Table 10.40: IOBLK_G1_REG_USB_VBUS_DET

Field Name	Bit	Default	FieldDescription
IOBLK_G1_REG_USB_VBUS_DET_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_G1_REG_USB_VBUS_DET_PD	3	0x0	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_G1_REG_USB_VBUS_DET_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_G1_REG_USB_VBUS_DET_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_G1_REG_USB_VBUS_DET_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_G1_REG_USB_VBUS_DET_ST1	9	0x0	Input Schmitt trigger strength control, bit 1
IOBLK_G1_REG_USB_VBUS_DET_HE	10	0x0	Weak bus holder enable. 0=Disabled; 1=Enabled
IOBLK_G1_REG_USB_VBUS_DET_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.5 IOBLK_G1_REG_PKG_TYPE1

Table 10.41: IOBLK_G1_REG_PKG_TYPE1

Field Name	Bit	Default	FieldDescription
IOBLK_G1_REG_PKG_TYPE1_PU	2	0x1	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_G1_REG_PKG_TYPE1_PD	3	0x0	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_G1_REG_PKG_TYPE1_DS0	5	0x0	Output drive strength level bit 0
IOBLK_G1_REG_PKG_TYPE1_DS1	6	0x1	Output drive strength level bit 1
IOBLK_G1_REG_PKG_TYPE1_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_G1_REG_PKG_TYPE1_ST1	9	0x0	Input Schmitt trigger strength control, bit 1
IOBLK_G1_REG_PKG_TYPE1_HE	10	0x0	Weak level holder (Bus holder) enable. 0=Disabled; 1=Enabled
IOBLK_G1_REG_PKG_TYPE1_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.6 IOBLK_G1_REG_PKG_TYPE2

Table 10.42: IOBLK_G1_REG_PKG_TYPE2

Field Name	Bit	Default	FieldDescription
IOBLK_G1_REG_USB_VBUS_DET_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_G1_REG_USB_VBUS_DET_PD	3	0x0	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_G1_REG_USB_VBUS_DET_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_G1_REG_USB_VBUS_DET_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_G1_REG_USB_VBUS_DET_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_G1_REG_USB_VBUS_DET_ST1	9	0x0	Input Schmitt trigger strength control, bit 1
IOBLK_G1_REG_USB_VBUS_DET_HE	10	0x0	Weak level holder (Bus holder) enable. 0=Disabled; 1=Enabled
IOBLK_G1_REG_USB_VBUS_DET_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.7 IOBLK_G7_REG_SD0_CD

Table 10.43: IOBLK_G7_REG_SD0_CD

Field Name	Bit	Default	FieldDescription
IOBLK_G7_REG_SD0_CD_PU	2	0x1	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_G7_REG_SD0_CD_PD	3	0x0	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_G7_REG_SD0_CD_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_G7_REG_SD0_CD_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_G7_REG_SD0_CD_DS2	7	0x0	Output drive strength level, bit 2
IOBLK_G7_REG_SD0_CD_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_G7_REG_SD0_CD_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.8 IOBLK_G7_REG_SD0_PWR_EN

Table 10.44: IOBLK_G7_REG_SD0_PWR_EN

Field Name	Bit	Default	FieldDescription
IOBLK_G7_REG_SD0_PWR_EN_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_G7_REG_SD0_PWR_EN_PD	3	0x1	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_G7_REG_SD0_PWR_EN_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_G7_REG_SD0_PWR_EN_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_G7_REG_SD0_PWR_EN_DS2	7	0x0	Output drive strength level, bit 2
IOBLK_G7_REG_SD0_PWR_EN_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_G7_REG_SD0_PWR_EN_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.9 IOBLK_G7_REG_SPK_EN

Table 10.45: IOBLK_G7_REG_SPK_EN

Field Name	Bit	Default	FieldDescription
IOBLK_G7_REG_SPK_EN_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_G7_REG_SPK_EN_PD	3	0x1	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_G7_REG_SPK_EN_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_G7_REG_SPK_EN_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_G7_REG_SPK_EN_DS2	7	0x0	Output drive strength level, bit 2
IOBLK_G7_REG_SPK_EN_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_G7_REG_SPK_EN_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.10 IOBLK_G7_REG_UART0_TX

Table 10.46: IOBLK_G7_REG_UART0_TX

Field Name	Bit	Default	FieldDescription
IOBLK_G7_REG_UART0_TX_PU	2	0x1	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_G7_REG_UART0_TX_PD	3	0x0	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_G7_REG_UART0_TX_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_G7_REG_UART0_TX_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_G7_REG_UART0_TX_DS2	7	0x0	Output drive strength level, bit 2
IOBLK_G7_REG_UART0_TX_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_G7_REG_UART0_TX_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.11 IOBLK_G7_REG_UART0_RX

Table 10.47: IOBLK_G7_REG_UART0_RX

Field Name	Bit	Default	FieldDescription
IOBLK_G7_REG_UART0_RX_PU	2	0x1	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_G7_REG_UART0_RX_PD	3	0x0	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_G7_REG_UART0_RX_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_G7_REG_UART0_RX_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_G7_REG_UART0_RX_DS2	7	0x0	Output drive strength level, bit 2
IOBLK_G7_REG_UART0_RX_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_G7_REG_UART0_RX_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.12 IOBLK_G7_REG_EMMC_DAT2

Table 10.48: IOBLK_G7_REG_EMMC_DAT2

Field Name	Bit	Default	FieldDescription
IOBLK_G7_REG_EMMC_DAT2_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_G7_REG_EMMC_DAT2_PD	3	0x1	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_G7_REG_EMMC_DAT2_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_G7_REG_EMMC_DAT2_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_G7_REG_EMMC_DAT2_DS2	7	0x0	Output drive strength level, bit 2
IOBLK_G7_REG_EMMC_DAT2_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_G7_REG_EMMC_DAT2_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.13 IOBLK_G7_REG_EMMC_CLK

Table 10.49: IOBLK_G7_REG_EMMC_CLK

Field Name	Bit	Default	FieldDescription
IOBLK_G7_REG_EMMC_CLK_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_G7_REG_EMMC_CLK_PD	3	0x1	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_G7_REG_EMMC_CLK_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_G7_REG_EMMC_CLK_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_G7_REG_EMMC_CLK_DS2	7	0x0	Output drive strength level, bit 2
IOBLK_G7_REG_EMMC_CLK_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_G7_REG_EMMC_CLK_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.14 IOBLK_G7_REG_EMMC_DAT0

Table 10.50: IOBLK_G7_REG_EMMC_DAT0

Field Name	Bit	Default	FieldDescription
IOBLK_G7_REG_EMMC_DAT0_PU	2	0x1	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_G7_REG_EMMC_DAT0_PD	3	0x0	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_G7_REG_EMMC_DAT0_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_G7_REG_EMMC_DAT0_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_G7_REG_EMMC_DAT0_DS2	7	0x0	Output drive strength level, bit 2
IOBLK_G7_REG_EMMC_DAT0_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_G7_REG_EMMC_DAT0_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.15 IOBLK_G7_REG_EMMC_DAT3

Table 10.51: IOBLK_G7_REG_EMMC_DAT3

Field Name	Bit	Default	FieldDescription
IOBLK_G7_REG_EMMC_DAT3_PU	2	0x1	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_G7_REG_EMMC_DAT3_PD	3	0x0	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_G7_REG_EMMC_DAT3_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_G7_REG_EMMC_DAT3_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_G7_REG_EMMC_DAT3_DS2	7	0x0	Output drive strength level, bit 2
IOBLK_G7_REG_EMMC_DAT3_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_G7_REG_EMMC_DAT3_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.16 IOBLK_G7_REG_EMMC_CMD

Table 10.52: IOBLK_G7_REG_EMMC_CMD

Field Name	Bit	Default	FieldDescription
IOBLK_G7_REG_EMMC_CMD_PU	2	0x1	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_G7_REG_EMMC_CMD_PD	3	0x0	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_G7_REG_EMMC_CMD_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_G7_REG_EMMC_CMD_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_G7_REG_EMMC_CMD_DS2	7	0x0	Output drive strength level, bit 2
IOBLK_G7_REG_EMMC_CMD_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_G7_REG_EMMC_CMD_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.17 IOBLK_G7_REG_EMMC_DAT1

Table 10.53: IOBLK_G7_REG_EMMC_DAT1

Field Name	Bit	Default	FieldDescription
IOBLK_G7_REG_EMMC_DAT1_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_G7_REG_EMMC_DAT1_PD	3	0x1	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_G7_REG_EMMC_DAT1_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_G7_REG_EMMC_DAT1_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_G7_REG_EMMC_DAT1_DS2	7	0x0	Output drive strength level, bit 2
IOBLK_G7_REG_EMMC_DAT1_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_G7_REG_EMMC_DAT1_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.18 IOBLK_G7_REG_JTAG_CPU_TMS

Table 10.54: IOBLK_G7_REG_JTAG_CPU_TMS

Field Name	Bit	Default	FieldDescription
IOBLK_G7_REG_JTAG_CPU_TMS_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_G7_REG_JTAG_CPU_TMS_PD	3	0x1	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_G7_REG_JTAG_CPU_TMS_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_G7_REG_JTAG_CPU_TMS_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_G7_REG_JTAG_CPU_TMS_DS2	7	0x0	Output drive strength level, bit 2
IOBLK_G7_REG_JTAG_CPU_TMS_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_G7_REG_JTAG_CPU_TMS_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.19 IOBLK_G7_REG_JTAG_CPU_TCK

Table 10.55: IOBLK_G7_REG_JTAG_CPU_TCK

Field Name	Bit	Default	FieldDescription
IOBLK_G7_REG_JTAG_CPU_TCK_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_G7_REG_JTAG_CPU_TCK_PD	3	0x1	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_G7_REG_JTAG_CPU_TCK_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_G7_REG_JTAG_CPU_TCK_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_G7_REG_JTAG_CPU_TCK_DS2	7	0x0	Output drive strength level, bit 2
IOBLK_G7_REG_JTAG_CPU_TCK_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_G7_REG_JTAG_CPU_TCK_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.20 IOBLK_G7_REG_IIC0_SCL

Table 10.56: IOBLK_G7_REG_IIC0_SCL

Field Name	Bit	Default	FieldDescription
IOBLK_G7_REG_IIC0_SCL_PU	2	0x1	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_G7_REG_IIC0_SCL_PD	3	0x0	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_G7_REG_IIC0_SCL_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_G7_REG_IIC0_SCL_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_G7_REG_IIC0_SCL_DS2	7	0x0	Output drive strength level, bit 2
IOBLK_G7_REG_IIC0_SCL_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_G7_REG_IIC0_SCL_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.21 IOBLK_G7_REG_IIC0_SDA

Table 10.57: IOBLK_G7_REG_IIC0_SDA

Field Name	Bit	Default	FieldDescription
IOBLK_G7_REG_IIC0_SDA_PU	2	0x1	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_G7_REG_IIC0_SDA_PD	3	0x0	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_G7_REG_IIC0_SDA_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_G7_REG_IIC0_SDA_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_G7_REG_IIC0_SDA_DS2	7	0x0	Output drive strength level, bit 2
IOBLK_G7_REG_IIC0_SDA_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_G7_REG_IIC0_SDA_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.22 IOBLK_G7_REG_AUX0

Table 10.58: IOBLK_G7_REG_AUX0

Field Name	Bit	Default	FieldDescription
IOBLKG7_REG_AUX0_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLKG7_REG_AUX0_PD	3	0x1	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLKG7_REG_AUX0_DS0	5	0x0	Output drive strength level, bit 0
IOBLKG7_REG_AUX0_DS1	6	0x1	Output drive strength level, bit 1
IOBLKG7_REG_AUX0_DS2	7	0x0	Output drive strength level, bit 2
IOBLKG7_REG_AUX0_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLKG7_REG_AUX0_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.23 IOBLK_G10_REG_SD0_CLK

Table 10.59: IOBLK_G10_REG_SD0_CLK

Field Name	Bit	Default	FieldDescription
IOBLKG10_REG_SD0_CLK_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLKG10_REG_SD0_CLK_PD	3	0x1	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_G10_REG_SD0_CLK_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_G10_REG_SD0_CLK_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_G10_REG_SD0_CLK_DS2	7	0x0	Output drive strength level, bit 2
IOBLK_G10_REG_SD0_CLK_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLKG10_REG_SD0_CLK_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.24 IOBLK_G10_REG_SD0_CMD

Table 10.60: IOBLK_G10_REG_SD0_CMD

Field Name	Bit	Default	FieldDescription
IOBLKG10_REG_SD0_CMD_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLKG10_REG_SD0_CMD_PD	3	0x1	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_G10_REG_SD0_CMD_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_G10_REG_SD0_CMD_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_G10_REG_SD0_CMD_DS2	7	0x0	Output drive strength level, bit 2
IOBLK_G10_REG_SD0_CMD_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLKG10_REG_SD0_CMD_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.25 IOBLK_G10_REG_SD0_D0

Table 10.61: IOBLK_G10_REG_SD0_D0

Field Name	Bit	Default	FieldDescription
IOBLK_G10_REG_SD0_D0_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_G10_REG_SD0_D0_PD	3	0x1	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_G10_REG_SD0_D0_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_G10_REG_SD0_D0_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_G10_REG_SD0_D0_DS2	7	0x0	Output drive strength level, bit 2
IOBLK_G10_REG_SD0_D0_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_G10_REG_SD0_D0_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.26 IOBLK_G10_REG_SD0_D1

Table 10.62: IOBLK_G10_REG_SD0_D1

Field Name	Bit	Default	FieldDescription
IOBLK_G10_REG_SD0_D1_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_G10_REG_SD0_D1_PD	3	0x1	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_G10_REG_SD0_D1_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_G10_REG_SD0_D1_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_G10_REG_SD0_D1_DS2	7	0x0	Output drive strength level, bit 2
IOBLK_G10_REG_SD0_D1_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_G10_REG_SD0_D1_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.27 IOBLK_G10_REG_SD0_D2

Table 10.63: IOBLK_G10_REG_SD0_D2

Field Name	Bit	Default	FieldDescription
IOBLK_G10_REG_SD0_D2_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_G10_REG_SD0_D2_PD	3	0x1	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_G10_REG_SD0_D2_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_G10_REG_SD0_D2_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_G10_REG_SD0_D2_DS2	7	0x0	Output drive strength level, bit 2
IOBLK_G10_REG_SD0_D2_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_G10_REG_SD0_D2_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.28 IOBLK_G10_REG_SD0_D3

Table 10.64: IOBLK_G10_REG_SD0_D3

Field Name	Bit	Default	FieldDescription
IOBLK_G10_REG_SD0_D3_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_G10_REG_SD0_D3_PD	3	0x1	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_G10_REG_SD0_D3_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_G10_REG_SD0_D3_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_G10_REG_SD0_D3_DS2	7	0x0	Output drive strength level, bit 2
IOBLK_G10_REG_SD0_D3_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_G10_REG_SD0_D3_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.29 IOBLK_G12_REG_PAD_MIPIRX4N

Table 10.65: IOBLK_G12_REG_PAD_MIPIRX4N

Field Name	Bit	Default	FieldDescription
IOBLK_G12_REG_PAD_MIPIRX4N_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_PAD_MIPIRX4N_PD	3	0x1	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_PAD_MIPIRX4N_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_G12_REG_PAD_MIPIRX4N_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_G12_REG_PAD_MIPIRX4N_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_G12_REG_PAD_MIPIRX4N_ST1	9	0x0	Input Schmitt trigger strength control, bit 1
IOBLK_G12_REG_PAD_MIPIRX4N_HE	10	0x0	Weak bus holder enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_PAD_MIPIRX4N_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.30 IOBLK_G12_REG_PAD_MIPIRX4P

Table 10.66: IOBLK_G12_REG_PAD_MIPIRX4P

Field Name	Bit	Default	FieldDescription
IOBLK_G12_REG_PAD_MIPIRX4P_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_PAD_MIPIRX4P_PD	3	0x1	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_PAD_MIPIRX4P_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_G12_REG_PAD_MIPIRX4P_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_G12_REG_PAD_MIPIRX4P_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_G12_REG_PAD_MIPIRX4P_ST1	9	0x0	Input Schmitt trigger strength control, bit 1
IOBLK_G12_REG_PAD_MIPIRX4P_HE	10	0x0	Weak bus holder enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_PAD_MIPIRX4P_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.31 IOBLK_G12_REG_PAD_MIPIRX3N

Table 10.67: IOBLK_G12_REG_PAD_MIPIRX3N

Field Name	Bit	Default	FieldDescription
IOBLK_G12_REG_PAD_MIPIRX3N_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_PAD_MIPIRX3N_PD	3	0x1	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_PAD_MIPIRX3N_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_G12_REG_PAD_MIPIRX3N_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_G12_REG_PAD_MIPIRX3N_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_G12_REG_PAD_MIPIRX3N_ST1	9	0x0	Input Schmitt trigger strength control, bit 1
IOBLK_G12_REG_PAD_MIPIRX3N_HE	10	0x0	Weak bus holder enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_PAD_MIPIRX3N_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.32 IOBLK_G12_REG_PAD_MIPIRX3P

Table 10.68: IOBLK_G12_REG_PAD_MIPIRX3P

Field Name	Bit	Default	FieldDescription
IOBLK_G12_REG_PAD_MIPIRX3P_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_PAD_MIPIRX3P_PD	3	0x1	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_PAD_MIPIRX3P_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_G12_REG_PAD_MIPIRX3P_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_G12_REG_PAD_MIPIRX3P_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_G12_REG_PAD_MIPIRX3P_ST1	9	0x0	Input Schmitt trigger strength control, bit 1
IOBLK_G12_REG_PAD_MIPIRX3P_HE	10	0x0	Weak bus holder enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_PAD_MIPIRX3P_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.33 IOBLK_G12_REG_PAD_MIPIRX2N

Table 10.69: IOBLK_G12_REG_PAD_MIPIRX2N

Field Name	Bit	Default	FieldDescription
IOBLK_G12_REG_PAD_MIPIRX2N_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_PAD_MIPIRX2N_PD	3	0x1	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_PAD_MIPIRX2N_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_G12_REG_PAD_MIPIRX2N_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_G12_REG_PAD_MIPIRX2N_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_G12_REG_PAD_MIPIRX2N_ST1	9	0x0	Input Schmitt trigger strength control, bit 1
IOBLK_G12_REG_PAD_MIPIRX2N_HE	10	0x0	Weak bus holder enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_PAD_MIPIRX2N_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.34 IOBLK_G12_REG_PAD_MIPIRX2P

Table 10.70: IOBLK_G12_REG_PAD_MIPIRX2P

Field Name	Bit	Default	FieldDescription
IOBLK_G12_REG_PAD_MIPIRX2P_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_PAD_MIPIRX2P_PD	3	0x1	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_PAD_MIPIRX2P_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_G12_REG_PAD_MIPIRX2P_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_G12_REG_PAD_MIPIRX2P_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_G12_REG_PAD_MIPIRX2P_ST1	9	0x0	Input Schmitt trigger strength control, bit 1
IOBLK_G12_REG_PAD_MIPIRX2P_HE	10	0x0	Weak bus holder enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_PAD_MIPIRX2P_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.35 IOBLK_G12_REG_PAD_MIPIRX1N

Table 10.71: IOBLK_G12_REG_PAD_MIPIRX1N

Field Name	Bit	Default	FieldDescription
IOBLK_G12_REG_PAD_MIPIRX1N_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_PAD_MIPIRX1N_PD	3	0x1	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_PAD_MIPIRX1N_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_G12_REG_PAD_MIPIRX1N_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_G12_REG_PAD_MIPIRX1N_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_G12_REG_PAD_MIPIRX1N_ST1	9	0x0	Input Schmitt trigger strength control, bit 1
IOBLK_G12_REG_PAD_MIPIRX1N_HE	10	0x0	Weak bus holder enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_PAD_MIPIRX1N_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.36 IOBLK_G12_REG_PAD_MIPIRX1P

Table 10.72: IOBLK_G12_REG_PAD_MIPIRX1P

Field Name	Bit	Default	FieldDescription
IOBLK_G12_REG_PAD_MIPIRX1P_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_PAD_MIPIRX1P_PD	3	0x1	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_PAD_MIPIRX1P_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_G12_REG_PAD_MIPIRX1P_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_G12_REG_PAD_MIPIRX1P_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_G12_REG_PAD_MIPIRX1P_ST1	9	0x0	Input Schmitt trigger strength control, bit 1
IOBLK_G12_REG_PAD_MIPIRX1P_HE	10	0x0	Weak bus holder enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_PAD_MIPIRX1P_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.37 IOBLK_G12_REG_PAD_MIPIRX0N

Table 10.73: IOBLK_G12_REG_PAD_MIPIRX0N

Field Name	Bit	Default	FieldDescription
IOBLK_G12_REG_PAD_MIPIRX0N_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_PAD_MIPIRX0N_PD	3	0x1	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_PAD_MIPIRX0N_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_G12_REG_PAD_MIPIRX0N_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_G12_REG_PAD_MIPIRX0N_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_G12_REG_PAD_MIPIRX0N_ST1	9	0x0	Input Schmitt trigger strength control, bit 1
IOBLK_G12_REG_PAD_MIPIRX0N_HE	10	0x0	Weak bus holder enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_PAD_MIPIRX0N_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.38 IOBLK_G12_REG_PAD_MIPIRX0P

Table 10.74: IOBLK_G12_REG_PAD_MIPIRX0P

Field Name	Bit	Default	FieldDescription
IOBLK_G12_REG_PAD_MIPIRX0P_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_PAD_MIPIRX0P_PD	3	0x1	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_PAD_MIPIRX0P_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_G12_REG_PAD_MIPIRX0P_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_G12_REG_PAD_MIPIRX0P_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_G12_REG_PAD_MIPIRX0P_ST1	9	0x0	Input Schmitt trigger strength control, bit 1
IOBLK_G12_REG_PAD_MIPIRX0P_HE	10	0x0	Weak bus holder enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_PAD_MIPIRX0P_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.39 IOBLK_G12_REG_PAD_MIPI_TXM2

Table 10.75: IOBLK_G12_REG_PAD_MIPI_TXM2

Field Name	Bit	Default	FieldDescription
IOBLK_G12_REG_PAD_MIPI_TXM2_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_PAD_MIPI_TXM2_PD	3	0x1	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_PAD_MIPI_TXM2_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_G12_REG_PAD_MIPI_TXM2_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_G12_REG_PAD_MIPI_TXM2_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_G12_REG_PAD_MIPI_TXM2_ST1	9	0x0	Input Schmitt trigger strength control, bit 1
IOBLK_G12_REG_PAD_MIPI_TXM2_HE	10	0x0	Weak bus holder enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_PAD_MIPI_TXM2_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.40 IOBLK_G12_REG_PAD_MIPI_TXP2

Table 10.76: IOBLK_G12_REG_PAD_MIPI_TXP2

Field Name	Bit	Default	FieldDescription
IOBLK_G12_REG_PAD_MIPI_TXP2_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_PAD_MIPI_TXP2_PD	3	0x1	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_PAD_MIPI_TXP2_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_G12_REG_PAD_MIPI_TXP2_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_G12_REG_PAD_MIPI_TXP2_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_G12_REG_PAD_MIPI_TXP2_ST1	9	0x0	Input Schmitt trigger strength control, bit 1
IOBLK_G12_REG_PAD_MIPI_TXP2_HE	10	0x0	Weak bus holder enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_PAD_MIPI_TXP2_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.41 IOBLK_G12_REG_PAD_MIPI_TXM1

Table 10.77: IOBLK_G12_REG_PAD_MIPI_TXM1

Field Name	Bit	Default	FieldDescription
IOBLK_G12_REG_PAD_MIPI_TXM1_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_PAD_MIPI_TXM1_PD	3	0x1	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_PAD_MIPI_TXM1_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_G12_REG_PAD_MIPI_TXM1_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_G12_REG_PAD_MIPI_TXM1_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_G12_REG_PAD_MIPI_TXM1_ST1	9	0x0	Input Schmitt trigger strength control, bit 1
IOBLK_G12_REG_PAD_MIPI_TXM1_HE	10	0x0	Weak bus holder enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_PAD_MIPI_TXM1_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.42 IOBLK_G12_REG_PAD_MIPI_TXP1

Table 10.78: IOBLK_G12_REG_PAD_MIPI_TXP1

Field Name	Bit	Default	FieldDescription
IOBLK_G12_REG_PAD_MIPI_TXP1_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_PAD_MIPI_TXP1_PD	3	0x1	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_PAD_MIPI_TXP1_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_G12_REG_PAD_MIPI_TXP1_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_G12_REG_PAD_MIPI_TXP1_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_G12_REG_PAD_MIPI_TXP1_ST1	9	0x0	Input Schmitt trigger strength control, bit 1
IOBLK_G12_REG_PAD_MIPI_TXP1_HE	10	0x0	Weak bus holder enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_PAD_MIPI_TXP1_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.43 IOBLK_G12_REG_PAD_MIPI_TXM0

Table 10.79: IOBLK_G12_REG_PAD_MIPI_TXM0

Field Name	Bit	Default	FieldDescription
IOBLK_G12_REG_PAD_MIPI_TXM0_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_PAD_MIPI_TXM0_PD	3	0x1	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_PAD_MIPI_TXM0_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_G12_REG_PAD_MIPI_TXM0_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_G12_REG_PAD_MIPI_TXM0_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_G12_REG_PAD_MIPI_TXM0_ST1	9	0x0	Input Schmitt trigger strength control, bit 1
IOBLK_G12_REG_PAD_MIPI_TXM0_HE	10	0x0	Weak bus holder enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_PAD_MIPI_TXM0_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.44 IOBLK_G12_REG_PAD_MIPI_TXP0

Table 10.80: IOBLK_G12_REG_PAD_MIPI_TXP0

Field Name	Bit	Default	FieldDescription
IOBLK_G12_REG_PAD_MIPI_TXP0_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_PAD_MIPI_TXP0_PD	3	0x1	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_PAD_MIPI_TXP0_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_G12_REG_PAD_MIPI_TXP0_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_G12_REG_PAD_MIPI_TXP0_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_G12_REG_PAD_MIPI_TXP0_ST1	9	0x0	Input Schmitt trigger strength control, bit 1
IOBLK_G12_REG_PAD_MIPI_TXP0_HE	10	0x0	Weak bus holder enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_PAD_MIPI_TXP0_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.45 IOBLK_G12_REG_GPIO_RTX

Table 10.81: IOBLK_G12_REG_GPIO_RTX

Field Name	Bit	Default	FieldDescription
IOBLK_G12_REG_GPIO_RTX_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_GPIO_RTX_PD	3	0x1	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_GPIO_RTX_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_G12_REG_GPIO_RTX_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_G12_REG_GPIO_RTX_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_G12_REG_GPIO_RTX_ST1	9	0x0	Input Schmitt trigger strength control, bit 1
IOBLK_G12_REG_GPIO_RTX_HE	10	0x0	Weak bus holder enable. 0=Disabled; 1=Enabled
IOBLK_G12_REG_GPIO_RTX_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.46 IOBLK_GRTC_REG_PWR_VBAT_DET

Table 10.82: IOBLK_GRTC_REG_PWR_VBAT_DET

Field Name	Bit	Default	FieldDescription
IOBLK_GRTC_REG_PWR_VBAT_DET_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_GRTC_REG_PWR_VBAT_DET_PD	3	0x0	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_GRTC_REG_PWR_VBAT_DET_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_GRTC_REG_PWR_VBAT_DET_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_GRTC_REG_PWR_VBAT_DET_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_GRTC_REG_PWR_VBAT_DET_ST1	9	0x0	Input Schmitt trigger strength control, bit 1
IOBLK_GRTC_REG_PWR_VBAT_DET_HE	10	0x0	Weak bus holder enable. 0=Disabled; 1=Enabled
IOBLK_GRTC_REG_PWR_VBAT_DET_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.47 IOBLK_GRTC_REG_PWR_RSTN

Table 10.83: IOBLK_GRTC_REG_PWR_RSTN

Field Name	Bit	Default	FieldDescription
IOBLK_GRTC_REG_PWR_RSTN_PU	2	0x1	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_GRTC_REG_PWR_RSTN_PD	3	0x0	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_GRTC_REG_PWR_RSTN_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_GRTC_REG_PWR_RSTN_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_GRTC_REG_PWR_RSTN_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_GRTC_REG_PWR_RSTN_ST1	9	0x0	Input Schmitt trigger strength control, bit 1
IOBLK_GRTC_REG_PWR_RSTN_HE	10	0x0	Weak bus holder enable. 0=Disabled; 1=Enabled
IOBLK_GRTC_REG_PWR_RSTN_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.48 IOBLK_GRTC_REG_PWR_SEQ1

Table 10.84: IOBLK_GRTC_REG_PWR_SEQ1

Field Name	Bit	Default	FieldDescription
IOBLK_GRTC_REG_PWR_SEQ1_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_GRTC_REG_PWR_SEQ1_PD	3	0x1	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_GRTC_REG_PWR_SEQ1_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_GRTC_REG_PWR_SEQ1_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_GRTC_REG_PWR_SEQ1_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_GRTC_REG_PWR_SEQ1_ST1	9	0x0	Input Schmitt trigger strength control, bit 1
IOBLK_GRTC_REG_PWR_SEQ1_HE	10	0x0	Weak bus holder enable. 0=Disabled; 1=Enabled
IOBLK_GRTC_REG_PWR_SEQ1_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.49 IOBLK_GRTC_REG_PWR_SEQ2

Table 10.85: IOBLK_GRTC_REG_PWR_SEQ2

Field Name	Bit	Default	FieldDescription
IOBLK_GRTC_REG_PWR_SEQ2_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_GRTC_REG_PWR_SEQ2_PD	3	0x1	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_GRTC_REG_PWR_SEQ2_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_GRTC_REG_PWR_SEQ2_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_GRTC_REG_PWR_SEQ2_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_GRTC_REG_PWR_SEQ2_ST1	9	0x0	Input Schmitt trigger strength control, bit 1
IOBLK_GRTC_REG_PWR_SEQ2_HE	10	0x0	Weak bus holder enable. 0=Disabled; 1=Enabled
IOBLK_GRTC_REG_PWR_SEQ2_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.50 IOBLK_GRTC_REG_PTEST

Table 10.86: IOBLK_GRTC_REG_PTEST

Field Name	Bit	Default	FieldDescription
IOBLK_GRTC_REG_PTEST_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_GRTC_REG_PTEST_PD	3	0x1	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_GRTC_REG_PTEST_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_GRTC_REG_PTEST_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_GRTC_REG_PTEST_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_GRTC_REG_PTEST_ST1	9	0x0	Input Schmitt trigger strength control, bit 1
IOBLK_GRTC_REG_PTEST_HE	10	0x0	Weak bus holder enable. 0=Disabled; 1=Enabled
IOBLK_GRTC_REG_PTEST_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.51 IOBLK_GRTC_REG_PWR_WAKEUP0

Table 10.87: IOBLK_GRTC_REG_PWR_WAKEUP0

Field Name	Bit	Default	FieldDescription
IOBLK_GRTC_REG_PWR_WAKEUP0_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_GRTC_REG_PWR_WAKEUP0_PD	3	0x1	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_GRTC_REG_PWR_WAKEUP0_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_GRTC_REG_PWR_WAKEUP0_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_GRTC_REG_PWR_WAKEUP0_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_GRTC_REG_PWR_WAKEUP0_ST1	9	0x0	Input Schmitt trigger strength control, bit 1
IOBLK_GRTC_REG_PWR_WAKEUP0_HE	10	0x0	Weak bus holder enable. 0=Disabled; 1=Enabled
IOBLK_GRTC_REG_PWR_WAKEUP0_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.52 IOBLK_GRTC_REG_PWR_BUTTON1

Table 10.88: IOBLK_GRTC_REG_PWR_BUTTON1

Field Name	Bit	Default	FieldDescription
IOBLK_GRTC_REG_PWR_BUTTON1_PU	2	0x1	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_GRTC_REG_PWR_BUTTON1_PD	3	0x0	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_GRTC_REG_PWR_BUTTON1_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_GRTC_REG_PWR_BUTTON1_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_GRTC_REG_PWR_BUTTON1_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_GRTC_REG_PWR_BUTTON1_ST1	9	0x0	Input Schmitt trigger strength control, bit 1
IOBLK_GRTC_REG_PWR_BUTTON1_HE	10	0x0	Weak bus holder enable. 0=Disabled; 1=Enabled
IOBLK_GRTC_REG_PWR_BUTTON1_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.53 IOBLK_GRTC_REG_XTAL_XIN

Table 10.89: IOBLK_GRTC_REG_XTAL_XIN

Field Name	Bit	Default	FieldDescription
IOBLK_GRTC_REG_XTAL_XIN_XDS0	1 4	0x0	Crystal oscillator output drive strength level, bit 0
IOBLK_GRTC_REG_XTAL_XIN_XDS1	1 5	0x0	Crystal oscillator output drive strength level, bit 1
IOBLK_GRTC_REG_XTAL_XIN_XDS2	1 6	0x1	Crystal oscillator output drive strength level, bit 2

10.2.2.54 IOBLK_GRTC_REG_PWR_GPIO0

Table 10.90: IOBLK_GRTC_REG_PWR_GPIO0

Field Name	Bit	Default	FieldDescription
IOBLK_GRTC_REG_PWR_GPIO0_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_GRTC_REG_PWR_GPIO0_PD	3	0x1	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_GRTC_REG_PWR_GPIO0_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_GRTC_REG_PWR_GPIO0_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_GRTC_REG_PWR_GPIO0_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_GRTC_REG_PWR_GPIO0_ST1	9	0x0	Input Schmitt trigger strength control, bit 1
IOBLK_GRTC_REG_PWR_GPIO0_HE	10	0x0	Weak bus holder enable. 0=Disabled; 1=Enabled
IOBLK_GRTC_REG_PWR_GPIO0_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.55 IOBLK_GRTC_REG_PWR_GPIO1

Table 10.91: IOBLK_GRTC_REG_PWR_GPIO1

Field Name	Bit	Default	FieldDescription
IOBLK_GRTC_REG_PWR_GPIO1_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_GRTC_REG_PWR_GPIO1_PD	3	0x1	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_GRTC_REG_PWR_GPIO1_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_GRTC_REG_PWR_GPIO1_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_GRTC_REG_PWR_GPIO1_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_GRTC_REG_PWR_GPIO1_ST1	9	0x0	Input Schmitt trigger strength control, bit 1
IOBLK_GRTC_REG_PWR_GPIO1_HE	10	0x0	Weak bus holder enable. 0=Disabled; 1=Enabled
IOBLK_GRTC_REG_PWR_GPIO1_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.56 IOBLK_GRTC_REG_PWR_GPIO2

Table 10.92: IOBLK_GRTC_REG_PWR_GPIO2

Field Name	Bit	Default	FieldDescription
IOBLK_GRTC_REG_PWR_GPIO2_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_GRTC_REG_PWR_GPIO2_PD	3	0x1	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_GRTC_REG_PWR_GPIO2_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_GRTC_REG_PWR_GPIO2_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_GRTC_REG_PWR_GPIO2_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_GRTC_REG_PWR_GPIO2_ST1	9	0x0	Input Schmitt trigger strength control, bit 1
IOBLK_GRTC_REG_PWR_GPIO2_HE	10	0x0	Weak bus holder enable. 0=Disabled; 1=Enabled
IOBLK_GRTC_REG_PWR_GPIO2_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.57 IOBLK_GRTC_REG_SD1_D3

Table 10.93: IOBLK_GRTC_REG_SD1_D3

Field Name	Bit	Default	FieldDescription
IOBLK_GRTC_REG_SD1_D3_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_GRTC_REG_SD1_D3_PD	3	0x1	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_GRTC_REG_SD1_D3_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_GRTC_REG_SD1_D3_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_GRTC_REG_SD1_D3_DS2	7	0x0	Output drive strength level, bit 2
IOBLK_GRTC_REG_SD1_D3_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_GRTC_REG_SD1_D3_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.58 IOBLK_GRTC_REG_SD1_D2

Table 10.94: IOBLK_GRTC_REG_SD1_D2

Field Name	Bit	Default	FieldDescription
IOBLK_GRTC_REG_SD1_D2_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_GRTC_REG_SD1_D2_PD	3	0x1	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_GRTC_REG_SD1_D2_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_GRTC_REG_SD1_D2_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_GRTC_REG_SD1_D2_DS2	7	0x0	Output drive strength level, bit 2
IOBLK_GRTC_REG_SD1_D2_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_GRTC_REG_SD1_D2_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.59 IOBLK_GRTC_REG_SD1_D1

Table 10.95: IOBLK_GRTC_REG_SD1_D1

Field Name	Bit	Default	FieldDescription
IOBLK_GRTC_REG_SD1_D1_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_GRTC_REG_SD1_D1_PD	3	0x1	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_GRTC_REG_SD1_D1_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_GRTC_REG_SD1_D1_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_GRTC_REG_SD1_D1_DS2	7	0x0	Output drive strength level, bit 2
IOBLK_GRTC_REG_SD1_D1_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_GRTC_REG_SD1_D1_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.60 IOBLK_GRTC_REG_SD1_D0

Table 10.96: IOBLK_GRTC_REG_SD1_D0

Field Name	Bit	Default	FieldDescription
IOBLK_GRTC_REG_SD1_D0_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_GRTC_REG_SD1_D0_PD	3	0x1	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_GRTC_REG_SD1_D0_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_GRTC_REG_SD1_D0_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_GRTC_REG_SD1_D0_DS2	7	0x0	Output drive strength level, bit 2
IOBLK_GRTC_REG_SD1_D0_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_GRTC_REG_SD1_D0_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.61 IOBLK_GRTC_REG_SD1_CMD

Table 10.97: IOBLK_GRTC_REG_SD1_CMD

Field Name	Bit	Default	FieldDescription
IOBLK_GRTC_REG_SD1_CMD_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_GRTC_REG_SD1_CMD_PD	3	0x1	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_GRTC_REG_SD1_CMD_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_GRTC_REG_SD1_CMD_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_GRTC_REG_SD1_CMD_DS2	7	0x0	Output drive strength level, bit 2
IOBLK_GRTC_REG_SD1_CMD_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_GRTC_REG_SD1_CMD_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.62 IOBLK_GRTC_REG_SD1_CLK

Table 10.98: IOBLK_GRTC_REG_SD1_CLK

Field Name	Bit	Default	FieldDescription
IOBLK_GRTC_REG_SD1_CLK_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_GRTC_REG_SD1_CLK_PD	3	0x1	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_GRTC_REG_SD1_CLK_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_GRTC_REG_SD1_CLK_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_GRTC_REG_SD1_CLK_DS2	7	0x0	Output drive strength level, bit 2
IOBLK_GRTC_REG_SD1_CLK_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_GRTC_REG_SD1_CLK_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

10.2.2.63 IOBLK_GRTC_REG_GPIO_ZQ

Table 10.99: IOBLK_GRTC_REG_GPIO_ZQ

Field Name	Bit	Default	FieldDescription
IOBLK_GRTC_REG_GPIO_ZQ_PU	2	0x0	Pull-up resistor enable. 0=Disabled; 1=Enabled
IOBLK_GRTC_REG_GPIO_ZQ_PD	3	0x1	Pull-down resistor enable. 0=Disabled; 1=Enabled
IOBLK_GRTC_REG_GPIO_ZQ_DS0	5	0x0	Output drive strength level, bit 0
IOBLK_GRTC_REG_GPIO_ZQ_DS1	6	0x1	Output drive strength level, bit 1
IOBLK_GRTC_REG_GPIO_ZQ_ST0	8	0x0	Input Schmitt trigger strength control, bit 0
IOBLK_GRTC_REG_GPIO_ZQ_ST1	9	0x0	Input Schmitt trigger strength control, bit 1
IOBLK_GRTC_REG_GPIO_ZQ_HE	10	0x0	Weak bus holder enable. 0=Disabled; 1=Enabled
IOBLK_GRTC_REG_GPIO_ZQ_SL	11	0x0	Output level transition rate limit. 0=Disabled (faster); 1=Enabled (slower)

CHAPTER

ELEVEN

DMA CONTROLLER

11.1 Overview

DMA (Direct Memory Access) does not require the CPU to interfere with data one by one. Data can be transferred directly between the memory and the device. This mechanism greatly reduces the CPU control time and increases the data transfer rate. It is very suitable for use in large amounts of data transfer. When the chip is working, it often requires multiple channels of data transmission at the same time. One channel requires a DMA hardware support, and the DMAC (DMA Controller) is responsible for the control of multiple channels. Diagram *DMAC hardware control flow diagram* is a DMAC hardware control flow chart. The source and destination devices can be in different AXI sinks.

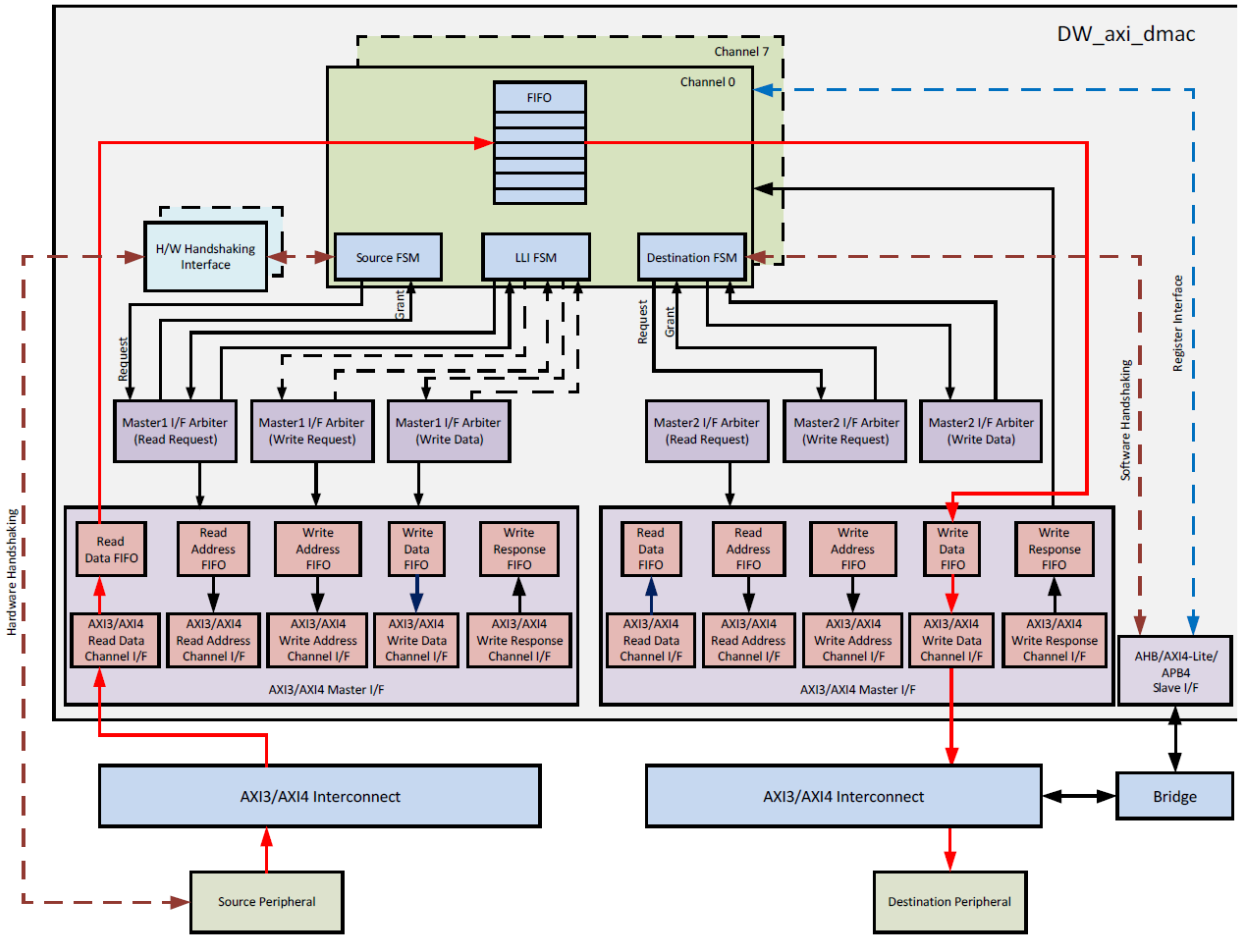


Diagram 11.1: DMAC hardware control flow diagram

11.2 Features

DMAC features are as follows:

a. A maximum of 8 DMA channels can be established simultaneously.

b. The data source and data purpose of the transmission can be set to memory or device.

c. Only one-way transfer of configuration is allowed.

d. Provide DMA transmission pause, resume, and cancellation.

e. Support DMA Burst length configuration.

f. Provide DMA channel priority configuration.

g. When channel data is transmitted between devices, flow control can be handed over to the device.

h. Support hardware linked list function.

i. Supports channel locking, and other channel requests before the locked channel is completed will be ignored.

11.3 Function Description

11.3.1 Peripheral Request Line

DMA has 8 built-in DMA channels, and each channel's peripheral request needs to be configured to be mapped to the peripheral. Please refer to *System DMA channel mapping* instructions to configure the DMA channel before enabling it.

11.3.2 Access Space

Table 11.1: DMAC Access Space Type

Space Type	Description
Memory	SRAM
	Non-safe DDR Space
Peripherals	UART0~UART3
	I2C0~I2C4
	SPI0~SPI3
	SPI_NAND
	I2S0~I2S3

11.3.3 Basic Transfer

DMA data transmission is set in blocks and completed by burst transmission. The length of burst transmission can be set, but what often happens is that the amount of block data is not perfectly an integer multiple of the burst transmission length. The length of the last transaction transmitted will be less than the set burst transmission length, and a single transmission request will be required to complete it.

The sources and destinations of the maximum supported 8 DMA channels can have the following four combinations:

a. RAM to RAM.

b. Memory to device.

c. Device to memory.

d. Device to device.

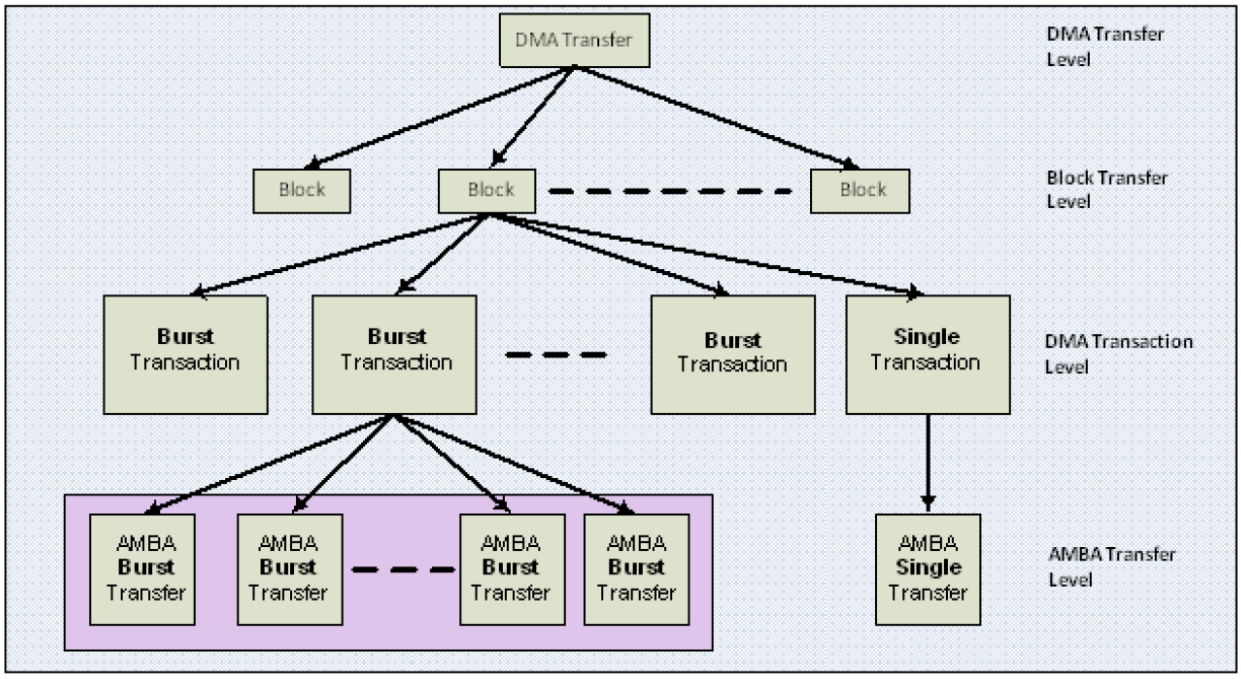


Diagram 11.2: DMA transfer layer

The amount of data to be transferred individually can be calculated by the value written to the following scratchpad

- Source transfer data volume (bytes):

  src_single_size_bytes = CHx_CTL.SRC_TR_WIDTH/8

- Source burst transfer data volume (bytes):

  src_burst_size_bytes = CHx_CTL.SRC_MSIZE * src_single_size_bytes

- Target transfer data volume (bytes):

  dst_single_size_bytes = CHx_CTL.DST_TR_WIDTH/8

- Target burst transfer data volume (bytes):

  dst_burst_size_bytes = CHx_CTL.DST_MSIZE * dst_single_size_bytes

The transmission process control can be controlled by the DMA controller or the source device or the destination device. When performing block data transfer, the amount of data transferred is calculated as follows:

- The transfer process is controlled by the DMA controller:

  blk_size_bytes_dma = CHx_BLOCK_TS.BLOCK_TS * src_single_size_bytes

- The source device controls the transfer process:

  blk_size_bytes_src = (number of block burst transfers issued by the source device * src_burst_size_bytes) + (number of individual block transfers from source device * src_single_size_bytes)

- The destination device controls the transfer process:

  blk_size_bytes_dst = (number of block burst transfers sent by the destination device * dst_burst_size_bytes) + (Number of separate transfers of destination device blocks * dst_single_size_bytes)

11.3.4 Linked-list Transfer

Linked-list transmission is used when block transmission of multiple discontinuous addresses is required. Each block of data will be followed by a linked-list information to store the information of the next node, so that no CPU intervention is required during data transmission. Block transfer of the next discontinuous space can be performed directly. Diagram *Linked list relative address and data format* is the configuration format of the linked list information. It must comply with the information format to enter the linked list transmission work.

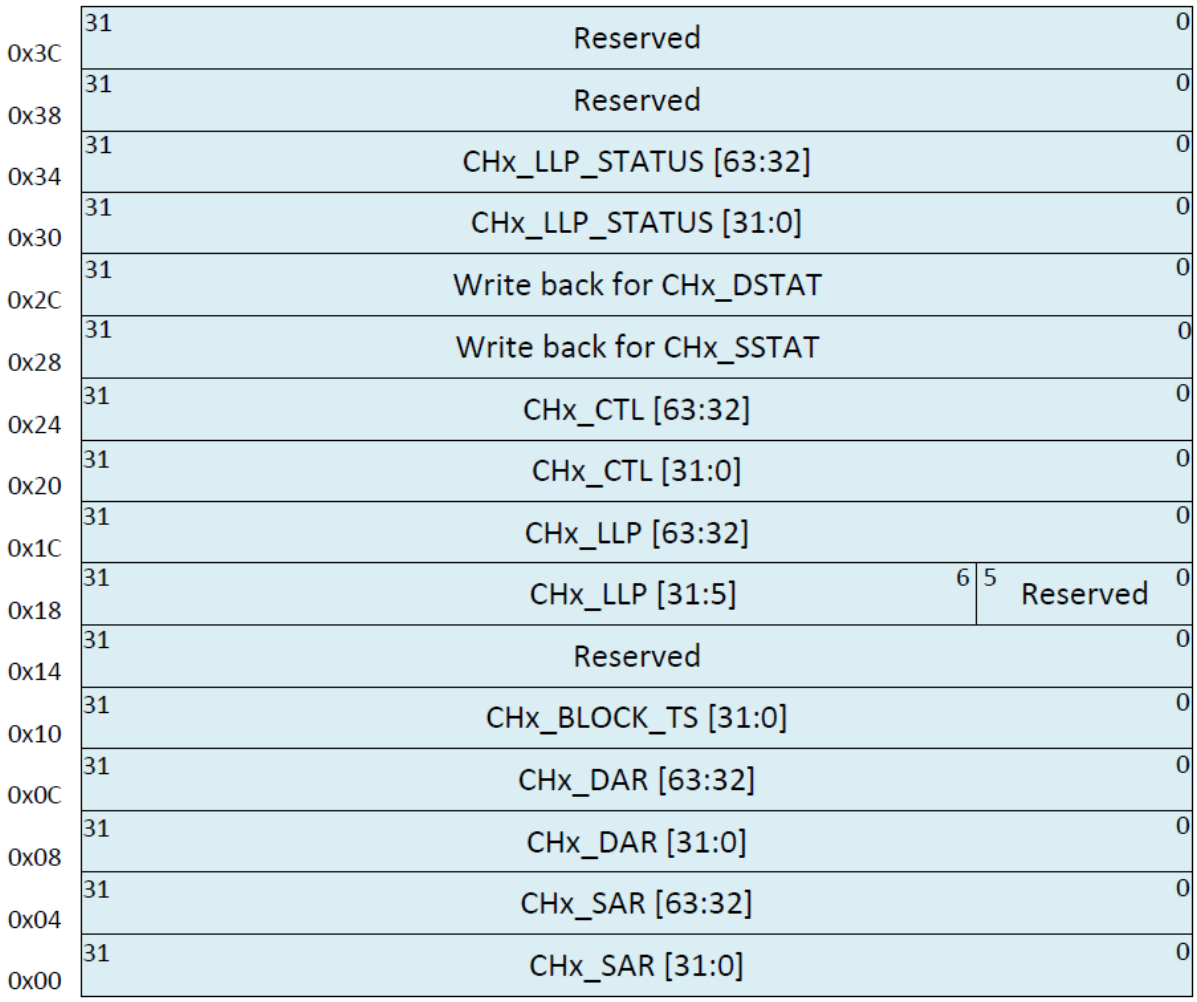


Diagram 11.3: Linked list relative address and data format

11.3.5 Interrupts and Status

The interrupt sources of DMAC are as follows:

a. Triggered by DMA transfer completion.

b. Triggered by block transfer completion.

c. Triggered when a single transfer is completed.

d. Internal error trigger.

e. Channel abort trigger or channel cancel trigger.

Diagram *Interrupt status and source diagram* is a diagram of interrupt status and source:

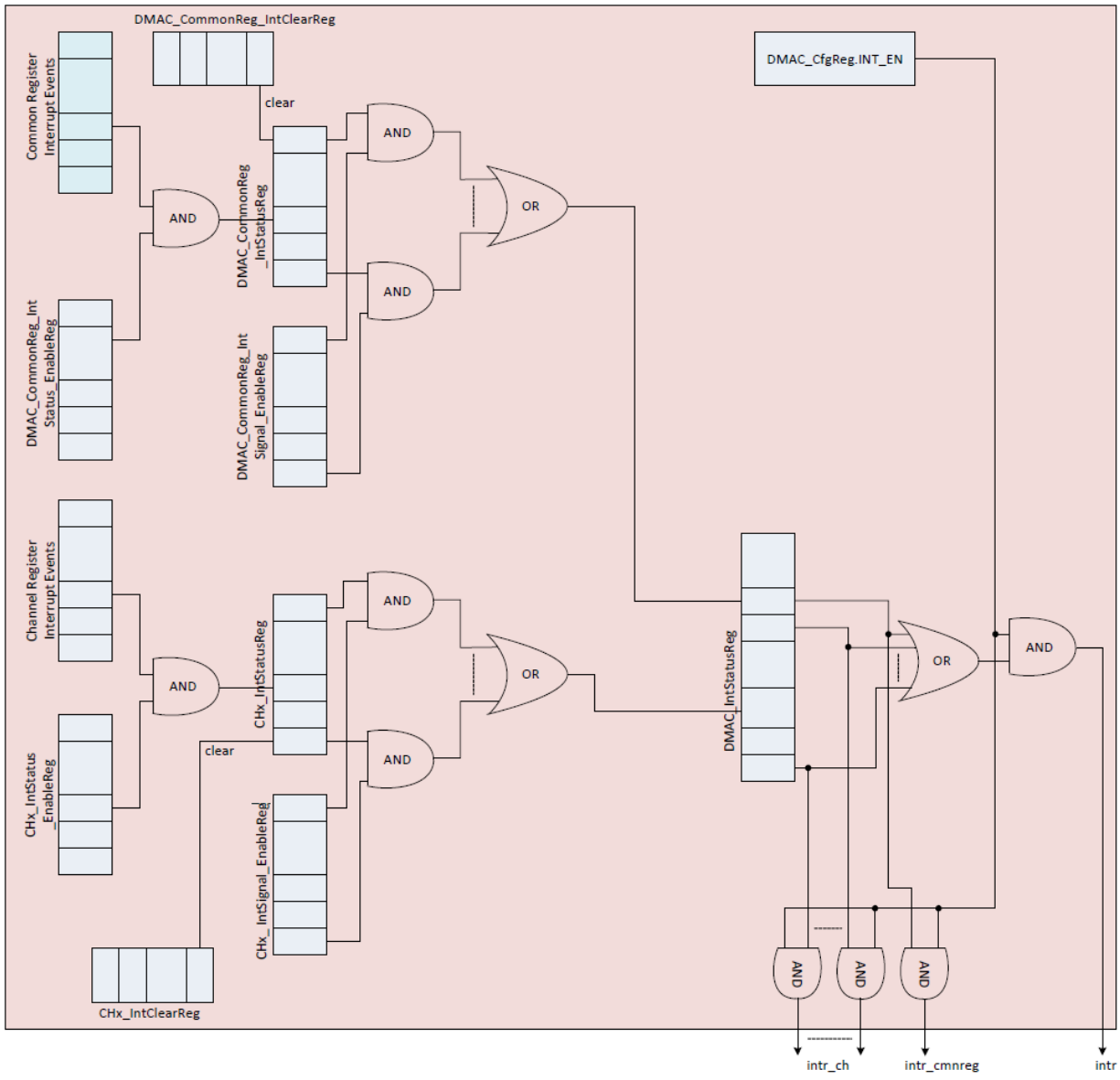


Diagram 11.4: Interrupt status and source diagram

11.3.6 Channel Security Configuration

Channel security can be implemented by the awprot value and arprot value of each channel. It complies with the AXI bus protocol. When the channel is a safe channel, the arprot or awprot value needs to be 0x0. If it is other values, it is an unsafe channel.

11.4 Way of working

11.4.1 Clock and Reset

The DMAC clock is enabled by writing 0x1 to CLK_EN_1[1], so that the clock can work normally. Writing REG_SOFT_RESET_X_SDMA_INIT to 0x0 resets the DMAC, writing 0x1 takes the DMAC out of reset.

11.4.2 Initialization

After reset, you can follow the steps below to initialize

1. Peripheral configuration: In the DMA channel mapping chapter, the configuration method of the DMA peripheral request line is explained. The binding configuration of the peripheral request line is done according to the usage scenario.
2. Confirm that the channel is closed: Write 0x0 to the enable register DMAC_ChEnReg of the 8 channels supported by DMA to confirm that the channel is closed.
3. Confirm the interrupt source: Write 0x0 to the registers DMAC_COMMONREG_INTSIGNAL_ENABLEREG and CHx_INSTATUS_ENABLEREG to turn off all interrupt sources, and then write 0x1 to enable the required interrupt source.
4. Configure the channel priority: When the channel transmits data at the same time, the order of passing will be judged according to the priority level. The larger the value written in the channel register CH_PRIOR, the higher the priority, and the priority will be passed.

11.4.3 Basic Transfer

It can support up to 8 channels for simultaneous transmission. After initialization, the DMAC channel needs to be enabled before data transmission can begin. You can refer to the following steps for memory-to-memory data transmission.

- Read the register DMAC_ChEnReg to obtain the idle channel.
- Write 0x0 to the channel register SRC_MULTBLK_TYPE and DST_MULTBLK_TYPE respectively to configure continuous block transfer.
- Write 0x0 to register TT_FC to configure the channel for memory-to-memory data transfer.
- Write the transferred information into the registers CHx_SAR, CHx_ADR, CHx_BLOCK_TS, CHx_CTL.
- Write 0x1 to the register DMAC_ChEnReg to enable the selected DMA channel.
- Software can obtain the BLOCK_TFR_DONE status through interrupts or polling. When its value rises to 1, it means that the data transmission has been completed. Then write 0x0 to DMAC_ChEnReg to close the channel and restore it to an idle channel.

11.4.4 Linked-list Transfer

Linked-list transfer does not limit the number of nodes. Except for the end node, each node must store information pointing to the next node. Linked-list transfer can be completed by referring to the following steps.

1. Read the register DMAC_ChEnReg to obtain the idle channel.
2. Write 0x3 to the channel register SRC_MULTBLK_TYPE and DST_MULTBLK_TYPE respectively to configure linked list transmission.
3. Configure the registers CHx_LLP, CHx_CTL.ShadowReg_Or_LLI_Valid, CHx_CTL.LLI_Last, and write the information required to point to the first node.
4. Write 0x1 to the register DMAC_ChEnReg to enable the selected DMA channel.
5. The software can obtain the BLOCK_TFR_DONE status through interrupts or polling. When its value rises to 1, it means that the last node data transmission has been completed, and then writes 0x0 to DMAC_ChEnReg to close the channel and restore it to an idle channel.

11.4.5 Interrupt handling

The processing steps after the interrupt is triggered are as follows:

1. Find the interrupt source: Read the registers CHx_INTSTATUS and DMAC_INSTATUSREG to get the interrupt source whose value is 0x1. When an interrupt occurs, it will be recorded as 0x1 in the corresponding selected bit. If multiple interrupts occur at the same time, the software can service them first according to their priority.
2. Clear interrupts: Write 0x1 to the selected bit of CHx_INTCLEARREG or DMAC_INTCLEARREG. At this time, the interrupt status recorded in CHx_INTSTATUS and DMAC_INSTATUSREG will return to 0x0, and the next interrupt condition will be recorded again.

11.5 DMAC Resisters

11.5.1 DMAC_IDREG

Table 11.2: DMAC_IDREG, Offset Address: 0x000

Bits	Name	Access	Description	Reset
63:0	DMAC_IDREG	RO	DMAC ID Number	

11.5.2 DMAC_COMPVERREG

Table 11.3: DMAC_COMPVERREG, Offset Address: 0x008

Bits	Name	Access	Description	Reset
31:0	DMAC_COMPVER	RO	DMAC Component Version Number.	

11.5.3 DMAC_CFGREG

Table 11.4: DMAC_CFGREG, Offset Address: 0x010

Bits	Name	Access	Description	Reset
0	DMAC_EN	R/W	This bit is used to enable the DW_axi_dmac. • 0: DW_axi_dmac disabled • 1: DW_axi_dmac enabled NOTE: If this bit DMAC_EN bit is cleared while any channel is still active, then this bit still returns 1 to indicate that there are channels still active until DW_axi_dmac hardware has terminated all activity on all channels, at which point this bit returns zero (0).	0x0
1	INT_EN	R/W	This bit is used to globally enable the interrupt generation. • 0: DW_axi_dmac Interrupts are disabled • 1: DW_axi_dmac Interrupt logic is enabled.	0x0
31:2	Reserved			

11.5.4 DMAC_CHENREG

Table 11.5: DMAC_CHENREG, Offset Address: 0x018

Bits	Name	Access	Description	Reset
0	CH1_EN	R/W	This bit is used to enable the DW_axi_dmac Channel-1. • 0: DW_axi_dmac Channel-1 is disabled • 1: DW_axi_dmac Channel-1 is enabled The bit 'DMAC_ChEnReg.CH1_EN' is automatically cleared by hardware to disable the channel after the last AMBA transfer of the DMA transfer to the destination has completed. Software can therefore poll this bit to determine when this channel is free for a new DMA transfer.	0x0
1	CH2_EN	R/W	This bit is used to enable the DW_axi_dmac Channel-2. • 0: DW_axi_dmac Channel-2 is disabled • 1: DW_axi_dmac Channel-2 is enabled The bit 'DMAC_ChEnReg.CH2_EN' is automatically cleared by hardware to disable the channel after the last AMBA transfer of the DMA transfer to the destination has completed. Software can therefore poll this bit to determine when this channel is free for a new DMA transfer.	0x0
2	CH3_EN	R/W	This bit is used to enable the DW_axi_dmac Channel-3. • 0: DW_axi_dmac Channel-3 is disabled • 1: DW_axi_dmac Channel-3 is enabled The bit 'DMAC_ChEnReg.CH3_EN' is automatically cleared by hardware to disable the channel after the last AMBA transfer of the DMA transfer to the destination has completed. Software can therefore poll this bit to determine when this channel is free for a new DMA transfer.	0x0
3	CH4_EN	R/W	This bit is used to enable the DW_axi_dmac Channel-4. • 0: DW_axi_dmac Channel-4 is disabled • 1: DW_axi_dmac Channel-4 is enabled The bit 'DMAC_ChEnReg.CH4_EN' is automatically cleared by hardware to disable the channel after the last AMBA transfer of the DMA transfer to the destination has completed. Software can therefore poll this bit to determine when this channel is free for a new DMA transfer.	0x0

continues on next page

Table 11.5 – continued from previous page

Bits	Name	Access	Description	Reset
4	CH5_EN	R/W	This bit is used to enable the DW_axi_dmac Channel-5. • 0: DW_axi_dmac Channel-5 is disabled • 1: DW_axi_dmac Channel-5 is enabled The bit 'DMAC_ChEnReg.CH5_EN' is automatically cleared by hardware to disable the channel after the last AMBA transfer of the DMA transfer to the destination has completed. Software can therefore poll this bit to determine when this channel is free for a new DMA transfer.	0x0
5	CH6_EN	R/W	This bit is used to enable the DW_axi_dmac Channel-6. • 0: DW_axi_dmac Channel-6 is disabled • 1: DW_axi_dmac Channel-6 is enabled The bit 'DMAC_ChEnReg.CH6_EN' is automatically cleared by hardware to disable the channel after the last AMBA transfer of the DMA transfer to the destination has completed. Software can therefore poll this bit to determine when this channel is free for a new DMA transfer.	0x0
6	CH7_EN	R/W	This bit is used to enable the DW_axi_dmac Channel-7. • 0: DW_axi_dmac Channel-7 is disabled • 1: DW_axi_dmac Channel-7 is enabled The bit 'DMAC_ChEnReg.CH7_EN' is automatically cleared by hardware to disable the channel after the last AMBA transfer of the DMA transfer to the destination has completed. Software can therefore poll this bit to determine when this channel is free for a new DMA transfer.	0x0
7	CH8_EN	R/W	This bit is used to enable the DW_axi_dmac Channel-8. • 0: DW_axi_dmac Channel-8 is disabled • 1: DW_axi_dmac Channel-8 is enabled The bit 'DMAC_ChEnReg.CH8_EN' is automatically cleared by hardware to disable the channel after the last AMBA transfer of the DMA transfer to the destination has completed. Software can therefore poll this bit to determine when this channel is free for a new DMA transfer.	0x0
8	CH1_EN_WE	WO	DW_axi_dmac Channel-1 Enable Write Enable bit. Read back value of this register bit is always '0'.	0x0
9	CH2_EN_WE	WO	DW_axi_dmac Channel-2 Enable Write Enable bit. Read back value of this register bit is always '0'.	0x0

continues on next page

Table 11.5 – continued from previous page

Bits	Name	Access	Description	Reset
10	CH3_EN_WE	WO	DW_axi_dmac Channel-3 Enable Write Enable bit. Read back value of this register bit is always '0'.	0x0
11	CH4_EN_WE	WO	DW_axi_dmac Channel-4 Enable Write Enable bit. Read back value of this register bit is always '0'.	0x0
12	CH5_EN_WE	WO	DW_axi_dmac Channel-5 Enable Write Enable bit. Read back value of this register bit is always '0'.	0x0
13	CH6_EN_WE	WO	DW_axi_dmac Channel-6 Enable Write Enable bit. Read back value of this register bit is always '0'.	0x0
14	CH7_EN_WE	WO	DW_axi_dmac Channel-7 Enable Write Enable bit. Read back value of this register bit is always '0'.	0x0
15	CH8_EN_WE	WO	DW_axi_dmac Channel-8 Enable Write Enable bit. Read back value of this register bit is always '0'.	0x0
16	CH1_SUSP	R/W	Channel-1 Suspend Request. Software sets this bit to 1 to request channel suspend. If this bit is set to 1, DW_axi_dmac suspends all DMA data transfers from the source gracefully until this bit is cleared. There is no guarantee that the current dma transaction will complete. This bit can also be used in conjunction with CH1_Status.CH_SUSPENDED to cleanly disable the channel without losing any data. In this case, software first sets CH1_SUSP bit to 1 and polls CH1_Status.CH_SUSPENDED till it is set to 1. Software can then clear CH1_EN bit to 0 to disable the channel. • 0: No Channel Suspend Request. • 1: Request for Channel Suspend. Software can clear CH1_SUSP bit to 0, after DW_axi_dmac sets CH1_Status.CH_SUSPENDED bit to 1, to exit the channel suspend mode. Note: CH_SUSP is cleared when channel is disabled.	0x0

continues on next page

Table 11.5 – continued from previous page

Bits	Name	Access	Description	Reset
17	CH2_SUSP	R/W	Channel-2 Suspend Request. Software sets this bit to 1 to request channel suspend. If this bit is set to 1, DW_axi_dmac suspends all DMA data transfers from the source gracefully until this bit is cleared. There is no guarantee that the current dma transaction will complete. This bit can also be used in conjunction with CH2_Status.CH_SUSPENDED to cleanly disable the channel without losing any data. In this case, software first sets CH2_SUSP bit to 1 and polls CH2_Status.CH_SUSPENDED till it is set to 1. Software can then clear CH2_EN bit to 0 to disable the channel. • 0: No Channel Suspend Request. • 1: Request for Channel Suspend. Software can clear CH2_SUSP bit to 0, after DW_axi_dmac sets CH2_Status.CH_SUSPENDED bit to 1, to exit the channel suspend mode. Note: CH_SUSP is cleared when channel is disabled.	0x0
18	CH3_SUSP	R/W	Channel-3 Suspend Request. Software sets this bit to 1 to request channel suspend. If this bit is set to 1, DW_axi_dmac suspends all DMA data transfers from the source gracefully until this bit is cleared. There is no guarantee that the current dma transaction will complete. This bit can also be used in conjunction with CH3_Status.CH_SUSPENDED to cleanly disable the channel without losing any data. In this case, software first sets CH3_SUSP bit to 1 and polls CH3_Status.CH_SUSPENDED till it is set to 1. Software can then clear CH3_EN bit to 0 to disable the channel. • 0: No Channel Suspend Request. • 1: Request for Channel Suspend. Software can clear CH3_SUSP bit to 0, after DW_axi_dmac sets CH3_Status.CH_SUSPENDED bit to 1, to exit the channel suspend mode. Note: CH_SUSP is cleared when channel is disabled.	0x0

continues on next page

Table 11.5 – continued from previous page

Bits	Name	Access	Description	Reset
19	CH4_SUSP	R/W	Channel-4 Suspend Request. Software sets this bit to 1 to request channel suspend. If this bit is set to 1, DW_axi_dmac suspends all DMA data transfers from the source gracefully until this bit is cleared. There is no guarantee that the current dma transaction will complete. This bit can also be used in conjunction with CH4_Status.CH_SUSPENDED to cleanly disable the channel without losing any data. In this case, software first sets CH4_SUSP bit to 1 and polls CH4_Status.CH_SUSPENDED till it is set to 1. Software can then clear CH4_EN bit to 0 to disable the channel. • 0: No Channel Suspend Request. • 1: Request for Channel Suspend. Software can clear CH4_SUSP bit to 0, after DW_axi_dmac sets CH4_Status.CH_SUSPENDED bit to 1, to exit the channel suspend mode. Note: CH_SUSP is cleared when channel is disabled.	0x0
20	CH5_SUSP	R/W	Channel-5 Suspend Request. Software sets this bit to 1 to request channel suspend. If this bit is set to 1, DW_axi_dmac suspends all DMA data transfers from the source gracefully until this bit is cleared. There is no guarantee that the current dma transaction will complete. This bit can also be used in conjunction with CH5_Status.CH_SUSPENDED to cleanly disable the channel without losing any data. In this case, software first sets CH5_SUSP bit to 1 and polls CH5_Status.CH_SUSPENDED till it is set to 1. Software can then clear CH5_EN bit to 0 to disable the channel. • 0: No Channel Suspend Request. • 1: Request for Channel Suspend. Software can clear CH5_SUSP bit to 0, after DW_axi_dmac sets CH5_Status.CH_SUSPENDED bit to 1, to exit the channel suspend mode. Note: CH_SUSP is cleared when channel is disabled.	0x0

continues on next page

Table 11.5 – continued from previous page

Bits	Name	Access	Description	Reset
21	CH6_SUSP	R/W	Channel-6 Suspend Request. Software sets this bit to 1 to request channel suspend. If this bit is set to 1, DW_axi_dmac suspends all DMA data transfers from the source gracefully until this bit is cleared. There is no guarantee that the current dma transaction will complete. This bit can also be used in conjunction with CH6_Status.CH_SUSPENDED to cleanly disable the channel without losing any data. In this case, software first sets CH6_SUSP bit to 1 and polls CH6_Status.CH_SUSPENDED till it is set to 1. Software can then clear CH6_EN bit to 0 to disable the channel. • 0: No Channel Suspend Request. • 1: Request for Channel Suspend. Software can clear CH6_SUSP bit to 0, after DW_axi_dmac sets CH6_Status.CH_SUSPENDED bit to 1, to exit the channel suspend mode. Note: CH_SUSP is cleared when channel is disabled.	0x0
22	CH7_SUSP	R/W	Channel-7 Suspend Request. Software sets this bit to 1 to request channel suspend. If this bit is set to 1, DW_axi_dmac suspends all DMA data transfers from the source gracefully until this bit is cleared. There is no guarantee that the current dma transaction will complete. This bit can also be used in conjunction with CH7_Status.CH_SUSPENDED to cleanly disable the channel without losing any data. In this case, software first sets CH7_SUSP bit to 1 and polls CH7_Status.CH_SUSPENDED till it is set to 1. Software can then clear CH7_EN bit to 0 to disable the channel. • 0: No Channel Suspend Request. • 1: Request for Channel Suspend. Software can clear CH7_SUSP bit to 0, after DW_axi_dmac sets CH7_Status.CH_SUSPENDED bit to 1, to exit the channel suspend mode. Note: CH_SUSP is cleared when channel is disabled.	0x0

continues on next page

Table 11.5 – continued from previous page

Bits	Name	Access	Description	Reset
23	CH8_SUSP	R/W	Channel-8 Suspend Request. Software sets this bit to 1 to request channel suspend. If this bit is set to 1, DW_axi_dmac suspends all DMA data transfers from the source gracefully until this bit is cleared. There is no guarantee that the current dma transaction will complete. This bit can also be used in conjunction with CH8_Status.CH_SUSPENDED to cleanly disable the channel without losing any data. In this case, software first sets CH8_SUSP bit to 1 and polls CH8_Status.CH_SUSPENDED till it is set to 1. Software can then clear CH8_EN bit to 0 to disable the channel. • 0: No Channel Suspend Request. • 1: Request for Channel Suspend. Software can clear CH8_SUSP bit to 0, after DW_axi_dmac sets CH8_Status.CH_SUSPENDED bit to 1, to exit the channel suspend mode. Note: CH_SUSP is cleared when channel is disabled.	0x0
24	CH1_SUSP_WE	WO	This bit is used as a write enable to the Channel-1 Suspend bit. The read back value of this register bit is always 0.	0x0
25	CH2_SUSP_WE	WO	This bit is used as a write enable to the Channel-2 Suspend bit. The read back value of this register bit is always 0.	0x0
26	CH3_SUSP_WE	WO	This bit is used as a write enable to the Channel-3 Suspend bit. The read back value of this register bit is always 0.	0x0
27	CH4_SUSP_WE	WO	This bit is used as a write enable to the Channel-4 Suspend bit. The read back value of this register bit is always 0.	0x0
28	CH5_SUSP_WE	WO	This bit is used as a write enable to the Channel-5 Suspend bit. The read back value of this register bit is always 0.	0x0
29	CH6_SUSP_WE	WO	This bit is used as a write enable to the Channel-6 Suspend bit. The read back value of this register bit is always 0.	0x0
30	CH7_SUSP_WE	WO	This bit is used as a write enable to the Channel-7 Suspend bit. The read back value of this register bit is always 0.	0x0
31	CH8_SUSP_WE	WO	This bit is used as a write enable to the Channel-8 Suspend bit. The read back value of this register bit is always 0.	0x0

continues on next page

Table 11.5 – continued from previous page

Bits	Name	Access	Description	Reset
32	CH1_ABORT	R/W	Channel-1 Abort Request. Software sets this bit to 1 to request channel abort. If this bit is set to 1, DW_axi_dmac disables the channel immediately. Aborting the channel might result in AXI Protocol violation as DW_axi_dmac does not make sure that all AXI transfers initiated on the master interface are completed.Aborting the channel is not recommended and should be used only in situations where a particular channel hangs due to no response from the corresponding AXI slave interface and software wants to disable the channel without resetting the entire DW_axi_dmac. It is recommended to try channel disabling first and then only opt for channel aborting. • 0: No Channel Abort Request. • 1: Request for Channel Abort. DW_axi_dmac clears this bit to 0 once the channel is aborted (when it sets CH1_Status.CH_ABORTED bit to 1).	0x0
33	CH2_ABORT	R/W	Channel-2 Abort Request. Software sets this bit to 1 to request channel abort. If this bit is set to 1, DW_axi_dmac disables the channel immediately. Aborting the channel might result in AXI Protocol violation as DW_axi_dmac does not make sure that all AXI transfers initiated on the master interface are completed.Aborting the channel is not recommended and should be used only in situations where a particular channel hangs due to no response from the corresponding AXI slave interface and software wants to disable the channel without resetting the entire DW_axi_dmac. It is recommended to try channel disabling first and then only opt for channel aborting. • 0: No Channel Abort Request. • 1: Request for Channel Abort. DW_axi_dmac clears this bit to 0 once the channel is aborted (when it sets CH2_Status.CH_ABORTED bit to 1).	0x0

continues on next page

Table 11.5 – continued from previous page

Bits	Name	Access	Description	Reset
34	CH3_ABORT	R/W	Channel-3 Abort Request. Software sets this bit to 1 to request channel abort. If this bit is set to 1, DW_axi_dmac disables the channel immediately. Aborting the channel might result in AXI Protocol violation as DW_axi_dmac does not make sure that all AXI transfers initiated on the master interface are completed.Aborting the channel is not recommended and should be used only in situations where a particular channel hangs due to no response from the corresponding AXI slave interface and software wants to disable the channel without resetting the entire DW_axi_dmac. It is recommended to try channel disabling first and then only opt for channel aborting. • 0: No Channel Abort Request. • 1: Request for Channel Abort. DW_axi_dmac clears this bit to 0 once the channel is aborted (when it sets CH3_Status.CH_ABORTED bit to 1).	0x0
35	CH4_ABORT	R/W	Channel-4 Abort Request. Software sets this bit to 1 to request channel abort. If this bit is set to 1, DW_axi_dmac disables the channel immediately. Aborting the channel might result in AXI Protocol violation as DW_axi_dmac does not make sure that all AXI transfers initiated on the master interface are completed.Aborting the channel is not recommended and should be used only in situations where a particular channel hangs due to no response from the corresponding AXI slave interface and software wants to disable the channel without resetting the entire DW_axi_dmac. It is recommended to try channel disabling first and then only opt for channel aborting. • 0: No Channel Abort Request. • 1: Request for Channel Abort. DW_axi_dmac clears this bit to 0 once the channel is aborted (when it sets CH4_Status.CH_ABORTED bit to 1).	0x0

continues on next page

Table 11.5 – continued from previous page

Bits	Name	Access	Description	Reset
36	CH5_ABORT	R/W	Channel-5 Abort Request. Software sets this bit to 1 to request channel abort. If this bit is set to 1, DW_axi_dmac disables the channel immediately. Aborting the channel might result in AXI Protocol violation as DW_axi_dmac does not make sure that all AXI transfers initiated on the master interface are completed.Aborting the channel is not recommended and should be used only in situations where a particular channel hangs due to no response from the corresponding AXI slave interface and software wants to disable the channel without resetting the entire DW_axi_dmac. It is recommended to try channel disabling first and then only opt for channel aborting. • 0: No Channel Abort Request. • 1: Request for Channel Abort. DW_axi_dmac clears this bit to 0 once the channel is aborted (when it sets CH5_Status.CH_ABORTED bit to 1).	0x0
37	CH6_ABORT	R/W	Channel-6 Abort Request. Software sets this bit to 1 to request channel abort. If this bit is set to 1, DW_axi_dmac disables the channel immediately. Aborting the channel might result in AXI Protocol violation as DW_axi_dmac does not make sure that all AXI transfers initiated on the master interface are completed.Aborting the channel is not recommended and should be used only in situations where a particular channel hangs due to no response from the corresponding AXI slave interface and software wants to disable the channel without resetting the entire DW_axi_dmac. It is recommended to try channel disabling first and then only opt for channel aborting. • 0: No Channel Abort Request. • 1: Request for Channel Abort. DW_axi_dmac clears this bit to 0 once the channel is aborted (when it sets CH6_Status.CH_ABORTED bit to 1).	0x0

continues on next page

Table 11.5 – continued from previous page

Bits	Name	Access	Description	Reset
38	CH7_ABORT	R/W	Channel-7 Abort Request. Software sets this bit to 1 to request channel abort. If this bit is set to 1, DW_axi_dmac disables the channel immediately. Aborting the channel might result in AXI Protocol violation as DW_axi_dmac does not make sure that all AXI transfers initiated on the master interface are completed.Aborting the channel is not recommended and should be used only in situations where a particular channel hangs due to no response from the corresponding AXI slave interface and software wants to disable the channel without resetting the entire DW_axi_dmac. It is recommended to try channel disabling first and then only opt for channel aborting. • 0: No Channel Abort Request. • 1: Request for Channel Abort. DW_axi_dmac clears this bit to 0 once the channel is aborted (when it sets CH7_Status.CH_ABORTED bit to 1).	0x0
39	CH8_ABORT	R/W	Channel-8 Abort Request. Software sets this bit to 1 to request channel abort. If this bit is set to 1, DW_axi_dmac disables the channel immediately. Aborting the channel might result in AXI Protocol violation as DW_axi_dmac does not make sure that all AXI transfers initiated on the master interface are completed.Aborting the channel is not recommended and should be used only in situations where a particular channel hangs due to no response from the corresponding AXI slave interface and software wants to disable the channel without resetting the entire DW_axi_dmac. It is recommended to try channel disabling first and then only opt for channel aborting. • 0: No Channel Abort Request. • 1: Request for Channel Abort. DW_axi_dmac clears this bit to 0 once the channel is aborted (when it sets CH8_Status.CH_ABORTED bit to 1).	0x0
40	CH1_ABORT_WE	R/W	This bit is used to write enable the Channel-1 Abort bit. The read back value of this register bit is always 0.	0x0
41	CH2_ABORT_WE	R/W	This bit is used to write enable the Channel-2 Abort bit. The read back value of this register bit is always 0.	0x0

continues on next page

Table 11.5 – continued from previous page

Bits	Name	Access	Description	Reset
42	CH3_ABORT_WE	R/W	This bit is used to write enable the Channel-3 Abort bit. The read back value of this register bit is always 0.	0x0
43	CH4_ABORT_WE	R/W	This bit is used to write enable the Channel-4 Abort bit. The read back value of this register bit is always 0.	0x0
44	CH5_ABORT_WE	R/W	This bit is used to write enable the Channel-5 Abort bit. The read back value of this register bit is always 0.	0x0
45	CH6_ABORT_WE	R/W	This bit is used to write enable the Channel-6 Abort bit. The read back value of this register bit is always 0.	0x0
46	CH7_ABORT_WE	R/W	This bit is used to write enable the Channel-7 Abort bit. The read back value of this register bit is always 0.	0x0
47	CH8_ABORT_WE	R/W	This bit is used to write enable the Channel-8 Abort bit. The read back value of this register bit is always 0.	0x0
63:48	RSVD_DMAC_CHENREG	RO	DMAC_CHENREG Reserved bits	

11.5.5 DMAC_INTSTATUSREG

Table 11.6: DMAC_INTSTATUSREG, Offset Address: 0x030

Bits	Name	Access	Description	Reset
0	CH1_IntStat	RO	Channel 1 Interrupt Status Bit.	
1	CH2_IntStat	RO	Channel 2 Interrupt Status Bit.	
2	CH3_IntStat	RO	Channel 3 Interrupt Status Bit.	
3	CH4_IntStat	RO	Channel 4 Interrupt Status Bit.	
4	CH5_IntStat	RO	Channel 5 Interrupt Status Bit.	
5	CH6_IntStat	RO	Channel 6 Interrupt Status Bit.	
6	CH7_IntStat	RO	Channel 7 Interrupt Status Bit.	
7	CH8_IntStat	RO	Channel 8 Interrupt Status Bit.	
15:8	Reserved			
16	CommonReg_IntStat	RO	Common Register Interrupt Status Bit.	
31:17	Reserved			

11.5.6 DMAC_COMMONREG_INTCLEARREG

Table 11.7: DMAC_COMMONREG_INTCLEARREG, Offset Address: 0x038

Bits	Name	Access	Description	Reset
0	Clear_SLVIF_CommonReg_DEC_ERR_IntStat	WO	Slave Interface Common Register Decode Error Interrupt clear Bit. This bit is used to clear the corresponding channel interrupt status bit (SLVIF_CommonReg_DEC_ERR_IntStat in DMAC_CommonReg_IntStatusReg.	0x0
1	Clear_SLVIF_CommonReg_WR2RO_ERR_IntStat	WO	Slave Interface Common Register Write to Read only Error Interrupt clear Bit. This bit is used to clear the corresponding channel interrupt status bit(SLVIF_CommonReg_WR2RO_ERR_IntStat in DMAC_CommonReg_IntStatusReg.	0x0
2	Clear_SLVIF_CommonReg_RD2WO_ERR_IntStat	WO	Slave Interface Common Register Read to Write only Error Interrupt clear Bit. This bit is used to clear the corresponding channel interrupt status bit(SLVIF_CommonReg_RD2WO_ERR_IntStat in DMAC_CommonReg_IntStatusReg.	0x0
3	Cl ear_SLVIF_CommonReg_WrOnHold_ERR_IntStat	WO	Slave Interface Common Register Write On Hold Error Interrupt clear Bit. This bit is used to clear the corresponding channel interrupt status bit(SLVIF_CommonReg_WrOnHold_ERR_IntStat in DMAC_CommonReg_IntStatusReg.	0x0
7:4	Reserved			
8	Clear_SLVIF_UndefinedReg_DEC_ERR_IntStat	WO	Slave Interface Undefined register Decode Error Interrupt clear Bit. This bit is used to clear the corresponding channel interrupt status bit(SLVIF_UndefinedReg_DEC_ERR_IntStat in DMAC_CommonReg_IntStatusReg.	0x0
31:9	Reserved			

11.5.7 DMAC_COMMONREG_INTSTATUS_ENABLEREG

Table 11.8: DMAC_COMMONREG_INTSTATUS_ENABLEREG,
Offset Address: 0x040

Bits	Name	Access	Description	Reset
0	Enable_SLVIF_Commo nReg_DEC_ERR_IntStat	R/W	Slave Interface Common Register Decode Error Interrupt Status Enable Bit. This bit is used to enable the corresponding channel interrupt status bit (SLVIF_Com monReg_DEC_ERR_IntStat in DMAC_C ommonReg_IntStatusReg.	0x0
1	Enable_SLVIF_CommonR eg_WR2RO_ERR_IntStat	R/W	Slave Interface Common Register Write to Read only Error Interrupt Status Enable Bit. This bit is used to enable the corresponding channel inter- rupt status bit (SLVIF_Commo nReg_WR2RO_ERR_IntStat in DMAC_C ommonReg_IntStatusReg.	0x0
2	Enable_SLVIF_CommonR eg_RD2WO_ERR_IntStat	R/W	Slave Interface Common Register Read to Write only Error Interrupt Status Enable Bit. This bit is used to enable the corresponding channel inter- rupt status bit (SLVIF_Commo nReg_RD2WO_ERR_IntStat in DMAC_C ommonReg_IntStatusReg.	0x0
3	Ena ble_SLVIF_CommonReg_ WrOnHold_ERR_IntStat	R/W	Slave Interface Common Register Write On Hold Error Interrupt Status Enable Bit. This bit is used to enable the cor- responding channel interrupt status bit (SLVIF_CommonRe g_WrOnHold_ERR_IntStat in DMAC_C ommonReg_IntStatusReg.	0x0
7:4	Reserved			
8	E nable_SLVIF_Undefine dReg_DEC_ERR_IntStat	R/W	Slave Interface Undefined register Decode Error Interrupt Status enable Bit. This bit is used to enable the corresponding channel interrupt status bit (SLVIF_Undefi nedReg_DEC_ERR_IntStat in DMAC_C ommonReg_IntStatusReg.	0x0
31:9	Reserved			

11.5.8 DMAC_COMMONREG_INTSIGNAL_ENABLEREG

Table 11.9: DMAC_COMMONREG_INTSIGNAL_ENABLEREG,
Offset Address: 0x048

Bits	Name	Access	Description	Reset
0	Enable_SLVIF_CommonReg_DEC_ERR_IntSignal	R/W	Slave Interface Common Register Decode Error Interrupt Signal Enable Bit. This bit is used to enable the propagation of corresponding channel interrupt status bit (SLVIF_Com monReg_DEC_ERR_IntStat in DMAC_C ommonReg_IntStatusReg) to generate a port level interrupt.	0x0
1	En able_SLVIF_CommonReg_WR2RO_ERR_IntSignal	R/W	Slave Interface Common Register Write to Read only Error Interrupt Signal Enable Bit. This bit is used to enable the propagation of corresponding channel interrupt status bit (SLVIF_Commo nReg_WR2RO_ERR_IntStat in DMAC_C ommonReg_IntStatusReg) to generate a port level interrupt.	0x0
2	En able_SLVIF_CommonReg_RD2WO_ERR_IntSignal	R/W	Slave Interface Common Register Read to Write only Error Interrupt Signal Enable Bit. This bit is used to enable the propagation of corresponding channel interrupt status bit (SLVIF_Commo nReg_RD2WO_ERR_IntStat in DMAC_C ommonReg_IntStatusReg) to generate a port level interrupt.	0x0
3	Enabl e_SLVIF_CommonReg_Wr OnHold_ERR_IntSignal	R/W	Slave Interface Common Register Write On Hold Error Interrupt Signal Enable Bit. This bit is used to enable the propagation of corresponding channel interrupt status bit(SLVIF_CommonRe g_WrOnHold_ERR_IntStat in DMAC_C ommonReg_IntStatusReg) to generate a port level interrupt.	0x0
7:4	Reserved			
8	Ena ble_SLVIF_UndefinedReg_DEC_ERR_IntSignal	R/W	Slave Interface Undefined register Decode Error Interrupt Signal Enable Bit. This bit is used to enable the propagation of corresponding channel interrupt status bit(SLVIF_Undefi nedReg_DEC_ERR_IntStat in DMAC_C ommonReg_IntStatusReg) to generate a port level interrupt.	0x0
31:9	Reserved			

11.5.9 DMAC_COMMONREG_INTSTATUSREG

Table 11.10: DMAC_COMMONREG_INTSTATUSREG, Offset Address: 0x050

Bits	Name	Access	Description	Reset
0	SLVIF_CommonReg_DEC_ERR_IntStat	RO	Slave Interface Common Register Decode Error Interrupt Status Bit. Decode Error generated by DW_axi_dmac during register access. This error occurs if the register access is to an invalid address in the common register space (0x000 to 0x0FF) resulting in error response by DW_axi_dmac slave interface. • 0: No Slave Interface Decode Errors. • 1: Slave Interface Decode Error detected. The Error Interrupt status is generated if the corresponding Status Enable bit in DMAC_CommonReg_IntStatus_Enable register bit is set to 1. This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in DMAC _COMMONREG_INTCLEARREG on enabling the channel (required when the interrupt is not enabled).	
1	SLVIF_CommonReg_WR2RO_ERR_IntStat	RO	Slave Interface Common Register Write to Read Only Error Interrupt Status bit. This error occurs if write operation is performed to a Read Only register in the common register space (0x000 to 0x0FF). • 0: No Slave Interface Write to Read Only Errors. • 1: Slave Interface Write to Read Only Error detected. Error Interrupt status is generated if the corresponding Status Enable bit in DMAC_CommonReg_IntStatus_Enable register bit is set to 1. This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in DMAC _COMMONREG_INTCLEARREG on enabling the channel (required when the interrupt is not enabled).	

To be continued ……

Table 11.11: DMAC_COMMONREG_INTSTATUSREG, Offset Address: 0x050 (continued)

Bits	Name	Access	Description	Reset
2	SLVIF_CommonReg_RD2WO_ERR_IntStat	RO	Slave Interface Common Register Read to Write only Error Interrupt Status bit. This error occurs if Read operation is performed to a Write Only register in the common register space (0x000 to 0x0FF). • 0: No Slave Interface Read to Write Only Errors. • 1: Slave Interface Read to Write Only Error detected. Error Interrupt status is generated if the corresponding Status Enable bit in DMAC_Comm onReg_IntStatus_Enable register bit is set to 1. This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in DMAC _COMMONREG_INTCLEARREG on enabling the channel (required when the interrupt is not enabled). Values: • 0x1 (Active_CommonReg_RD2WO_ERR): Slave Interface Read to Write Only Error detected • 0x0 (Inactive_CommonReg_RD2WO_ERR): No Slave Interface Read to Write Only Errors	
3	SLVIF_CommonReg_WrOnHold_ERR_IntStat	RO	Slave Interface Common Register Write On Hold Error Interrupt Status Bit. This error occurs if an illegal write operation is performed on a common register; this happens if a write operation is performed on a common register except DMAC_RESETREG with DMAC_RST field set to 1 when DW_axi_dmac is in Hold mode. • 0: No Slave Interface Common Register Write On Hold Errors. • 1: Slave Interface Common Register Write On Hold Error detected. Error Interrupt Status is generated if the corresponding Status Enable bit in DMAC_Comm onReg_IntStatus_Enable register bit is set to 1. This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in DMAC _COMMONREG_INTCLEARREG on enabling the channel (required when the interrupt is not enabled).	

To be continued ……

Table 11.12: DMAC_COMMONREG_INTSTATUSREG, Offset Address: 0x050 (continued)

Bits	Name	Access	Description	Reset
7:4	Reserved			
8	SLVIF_UndefinedReg_DEC_ERR_IntStat	RO	Slave Interface Undefined register Decode Error Interrupt Signal Enable Bit. Decode Error generated by DW_axi_dmac during register access. This error occurs if the register access is to undefined address range (>0x8FF if 8 channels are configured, >0x4FF if 4 channels are configured etc.) resulting in error response by DW_axi_dmac slave interface. • 0: No Slave Interface Decode Errors. • 1: Slave Interface Decode Error detected. Error Interrupt Status is generated if the corresponding Status Enable bit in DMAC_CommonReg_IntStatus_Enable register bit is set to 1. This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in DMAC_COMMONREG_INTCLEARREG on enabling the channel (required when the interrupt is not enabled).	
31:9	Reserved			

11.5.10 DMAC_RESETREG

Table 11.13: DMAC_RESETREG, Offset Address: 0x058

Bits	Name	Access	Description	Reset
0	DMAC_RST	R/W	DMAC Reset Request bit Software writes 1 to this bit to reset the DW_axi_dmac and polls this bit to see it as 0. DW_axi_dmac resets all the modules except the slave bus interface module and clears this bit to 0. NOTE: Software is not allowed to write 0 to this bit.	0x0
31:1	Reserved			

11.5.11 CHx_SAR

Table 11.14: CHx_SAR, Offset Address: 0x100

Bits	Name	Access	Description	Reset
63:0	SAR	R/W	Current Source Address of DMA transfer. Updated after each source transfer. The SINC fields in the CHx_CTL register determines whether the address increments or is left unchanged on every source transfer throughout the block transfer.	0x0

11.5.12 CHx_DAR

Table 11.15: CHx_DAR, Offset Address: 0x108

Bits	Name	Access	Description	Reset
63:0	DAR	R/W	Current Destination Address of DMA transfer. Updated after each destination transfer. The DINC fields in the CHx_CTL register determines whether the address increments or is left unchanged on every destination transfer throughout the block transfer.	0x0

11.5.13 CHx_BLOCK_TS

Table 11.16: CHx_BLOCK_TS, Offset Address: 0x110

Bits	Name	Access	Description	Reset
21:0	BLOCK_TS	R/W	Block Transfer Size. The number programmed into BLOCK_TS field indicates the total number of data of width CHx_CTL.SRC_TR_WIDTH to be transferred in a DMA block transfer. Block Transfer Size = BLOCK_TS+1 When the transfer starts, the read-back value is the total number of data items already read from the source peripheral, regardless of who is the flow controller. When the source or destination peripheral is assigned as the flow controller, the value before the transfer starts saturates at DMAX_CHx_MAX_BLK_SIZE, but the actual block size can be greater.	0x0
31:22	Reserved			

11.5.14 CHx_CTL

Table 11.17: CHx_CTL, Offset Address: 0x118

Bits	Name	Access	Description	Reset
0	SMS	R/W	Source Master Select. Identifies the Master Interface layer from which the source device (peripheral or memory) is accessed. • 0: AXI master 1 • 1: AXI Master 2	0x0
1	Reserved			
2	DMS	R/W	Destination Master Select. Identifies the Master Interface layer from which the destination device (peripheral or memory) is accessed. • 0: AXI master 1 • 1: AXI Master 2	0x0
3	Reserved			
4	SINC	R/W	Source Address Increment. Indicates whether to increment the source address on every source transfer. If the device is fetching data from a source peripheral FIFO with a fixed address, then set this field to 'No change'. • 0: Increment • 1: No Change	0x0
5	Reserved			
6	DINC	R/W	Destination Address Increment. Indicates whether to increment the destination address on every destination transfer. If the device is writing data from a source peripheral FIFO with a fixed address, then set this field to 'No change'. • 0: Increment • 1: No Change	0x0
7	Reserved			
10:8	SRC_TR_WIDTH	R/W	Source Transfer Width. Mapped to AXI bus arsize, this value must be less than or equal to DMAX_M_DATA_WIDTH.	0x0
13:11	DST_TR_WIDTH	R/W	Destination Transfer Width. Mapped to AXI bus awsize, this value must be less than or equal to DMAX_M_DATA_WIDTH.	0x0

To be continued ……

Table 11.18: CHx_CTL, Offset Address: 0x118 (continued)

Bits	Name	Access	Description	Reset
17:14	SRC_MSIZE	R/W	Source Burst Transaction Length. Number of data items, each of width CHx_CTL.SRC_TR_WIDTH, to be read from the source every time a source burst transaction request is made from the corresponding hardware or software handshaking interface. The maximum value of DST_MSIZE is limited by DMAX_CHx_MAX_MULT_SIZE.	0x0
21:18	DST_MSIZE	R/W	Destination Burst Transaction Length. Number of data items, each of width CHx_CTL.DST_TR_WIDTH, to be written to the destination every time a destination burst transaction request is made from the corresponding hardware or software handshaking interface.Note: This Value is not related to the AXI awlen signal.	0x0
25:22	AR_CACHE	R/W	AXI 'ar_cache' signal	0x0
29:26	AW_CACHE	R/W	AXI 'aw_cache' signal	0x0
30	No nPosted_LastWrite_En	R/W	Non Posted Last Write Enable This bit decides whether posted writes can be used throughout the block transfer. • 0: Posted writes may be used throughout the block transfer. • 1: Posted writes may be used till the end of the block (inside a block) and the last write in the block must be non-posted. This is to synchronize block completion interrupt generation to the last write data reaching the end memory/peripheral.	0x0
31	Reserved			
34:32	AR_PROT	R/W	AXI 'ar_prot' signal	0x0
37:35	AW_PROT	R/W	AXI 'aw_prot' signal	0x0
38	ARLEN_EN	R/W	Source Burst Length Enable If this bit is set to 1, DW_axi_dmac uses the value of CHx_CTL.ARLEN as AXI Burst length for source data transfer till the extent possible; remaining transfers use maximum possible burst length. If this bit is set to 0, DW_axi_dmac uses any possible value that is less than or equal to DMAX_CHx_MAX_AMBA_BURST_LENGTH as AXI Burst length for source data transfer.	0x0

To be continued ……

Table 11.19: CHx_CTL, Offset Address: 0x118 (continued)

Bits	Name	Access	Description	Reset
46:39	ARLEN	R/W	Source Burst Length AXI Burst length used for source data transfer. The specified burst length is used for source data transfer till the extent possible; remaining transfers use maximum possible value that is less than or equal to DMAX_CHx_ MAX_AMBA_BURST_LENGTH. The maximum value of ARLEN is limited by DMAX_CHx_MAX_AMBA_BURST_LENGTH	0x0
47	AWLEN_EN	R/W	Destination Burst Length Enable If this bit is set to 1, DW_axi_dmac uses the value of CHx_CTL.AWLEN as AXI Burst length for destination data transfer till the extent possible; remaining transfers use maximum possible burst length. If this bit is set to 0, DW_axi_dmac uses any possible value which is less than or equal to DMAX_CHx_MAX_AMBA_BURST_LENGTH as AXI Burst length for destination data transfer.	0x0
55:48	AWLEN	RO	Destination Burst Length AXI Burst length used for destination data transfer. The specified burst length is used for destination data transfer till the extent possible; remaining transfers use maximum possible value that is less than or equal to DMAX_CHx_MAX_AMBA_BURST_LENGTH. The maximum value of AWLEN is limited by DMAX_CHx_MAX_AMBA_BURST_LENGTH.	
56	SRC_STAT_EN	R/W	Source Status Enable Enable the logic to fetch status from source peripheral of channel x pointed to by the content of CHx_SSTATAR register and stores it in CHx_SSTAT register. This value is written back to the CHx_SSTAT location of linked list at end of each block transfer if DMAX_CHx_LLI_WB_EN is set to 1 and if linked list based multi-block transfer is used by either source or destination peripheral.	0x0

To be continued ……

Table 11.20: CHx_CTL, Offset Address: 0x118 (continued)

Bits	Name	Access	Description	Reset
57	DST_STAT_EN	R/W	Destination Status Enable Enable the logic to fetch status from destination peripheral of channel x pointed to by the content of CHx_DSTATAR register and stores it in CHx_DSTAT register. This value is written back to the CHx_DSTAT location of linked list at end of each block transfer if DMAX_CHx_LLI_WB_EN is set to 1 and if linked list based multi-block transfer is used by either source or destination peripheral.	0x0
58	IOC_BlkTfr	R/W	Interrupt On completion of Block Transfer This bit is used to control the block transfer completion interrupt generation on a block by block basis for shadow register or linked list based multi-block transfers. Writing 1 to this register field enables CHx_IntStatusReg. BLOCK_TFR_DONE_IntStat field if this interrupt generation is enabled in CHx_IntStatus_EnableReg register and the external interrupt output is is asserted if this interrupt generation is enabled in CHx_IntSignal_EnableReg register.	0x0
61:59	Reserved			
62	SHADOWREG_OR_LLI_LAST	R/W	Last Shadow Register/Linked List Item. Indicates whether shadow register content or the linked list item fetched from the memory is the last one or not. • 0: Not last Shadow Register/LLI • 1: Last Shadow Register/LLI	0x0
63	SHADOWREG_OR_LLI_VALID	R/W	Shadow Register content/Linked List Item valid. Indicates whether the content of shadow register or the linked list item fetched from the memory is valid. • 0: Shadow Register content/LLI is invalid. • 1: Last Shadow Register/LLI is valid.	0x0

11.5.15 CHx_CFG

Table 11.21: CHx_CFG, Offset Address: 0x120

Bits	Name	Access	Description	Reset
1:0	SRC_MULTBLK_TYPE	RO	Source Multi Block Transfer Type. These bits define the type of multi-block transfer used for source peripheral. • 00: Contiguous • 01: Reload • 10: Shadow Register • 11: Linked List If the type selected is Contiguous, the CHx_SAR register is loaded with the value of the end source address of previous block + 1 at the end of every block for multi-block transfers. A new block transfer is then initiated. If the type selected is Reload, the CHx_SAR register is reloaded from the initial value of SAR at the end of every block for multi-block transfers. A new block transfer is then initiated. If the type selected is Shadow Register, the CHx_SAR register is loaded from the content of its shadow register if CHx_CTL. ShadowReg_Or_LLI_Valid bit is set to 1 at the end of every block for multi-block transfers. A new block transfer is then initiated. If the type selected is Linked List, the CHx_SAR register is loaded from the Linked List if CTL. ShadowReg_Or_LLI_Valid bit is set to 1 at the end of every block for multi-block transfers. A new block transfer is then initiated. CHx_CTL and CHx_BLOCK_TS registers are loaded from their initial values or from the contents of their shadow registers (if CHx_CTL. ShadowReg_Or_LLI_Valid bit is set to 1) or from the linked list (if CTL. ShadowReg_Or_LLI_Valid bit is set to 1) at the end of every block for multi-block transfers based on the multi-block transfer type programmed for source and destination peripherals. Contiguous transfer on both source and destination peripheral is not a valid multi-block transfer configuration. This field does not exist if the configuration parameter DMAX_CHx_MULTI_BLK_EN is not selected; in that case, the read-back value is always 0. Values: • 0x0 (CONTINGUOUS): Contiguous Multiblock Type used for Source Transfer • 0x1 (RELOAD): Reload Multiblock Type used for Source Transfer • 0x2 (SHADOW_REGISTER): Shadow Register based Multiblock Type used for Source Transfer • 0x3 (LINKED_LIST): Linked List based Multiblock Type used for Source Transfer	

To be continued ……

Table 11.22: CHx_CFG, Offset Address: 0x120 (continued)

Bits	Name	Access	Description	Reset
3:2	DST_MULTBLK_TYPE	R/W	Destination Multi Block Transfer Type. These bits define the type of multi-block transfer used for destination peripheral. • 00: Contiguous • 01: Reload • 10: Shadow Register • 11: Linked List If the type selected is Contiguous, the CHx_DAR register is loaded with the value of the end source address of previous block + 1 at the end of every block for multi-block transfers. A new block transfer is then initiated. If the type selected is Reload, the CHx_DAR register is reloaded from the initial value of DAR at the end of every block for multi-block transfers. A new block transfer is then initiated. If the type selected is Shadow Register, the CHx_DAR register is loaded from the content of its shadow register if CHx_CTL. ShadowReg_Or_LLI_Valid bit is set to 1 at the end of every block for multi-block transfers. A new block transfer is then initiated. If the type selected is Linked List, the CHx_DAR register is loaded from the Linked List if CTL. ShadowReg_Or_LLI_Valid bit is set to 1 at the end of every block for multi-block transfers. A new block transfer is then initiated. CHx_CTL and CHx_BLOCK_TS registers are loaded from their initial values or from the contents of their shadow registers (if CHx_CTL. ShadowReg_Or_LLI_Valid bit is set to 1) or from the linked list (if CTL. ShadowReg_Or_LLI_Valid bit is set to 1) at the end of every block for multi-block transfers based on the multi-block transfer type programmed for source and destination peripherals. Contiguous transfer on both source and destination peripheral is not a valid multi-block transfer configuration. This field does not exist if the configuration parameter DMAX_CHx_MULTI_BLK_EN is not selected; in that case, the read-back value is always 0. Values: • 0x0 (CONTINGUOUS): Contiguous Multiblock Type used for Destination Transfer • 0x1 (RELOAD): Reload Multiblock Type used for Destination Transfer • 0x2 (SHADOW_REGISTER): Shadow Register based Multiblock Type used for Destination Transfer • 0x3 (LINKED_LIST): Linked List based Multiblock Type used for Destination Transfer	0x0

To be continued ……

Table 11.23: CHx_CFG, Offset Address: 0x120 (continued)

Bits	Name	Access	Description	Reset
31:4	Reserved			
34:32	TT_FC	R/W	Transfer Type and Flow Control. The following transfer types are supported. Memory to Memory Memory to Peripheral Peripheral to Memory Peripheral to Peripheral Flow Control can be assigned to the DW_axi_dmac, the source peripheral, or hte destination peripheral. Values: • 0x0 (MEM_TO_MEM_DMAC): Transfer Type is memory to memory and Flow Controller is DW_axi_dmac • 0x1 (MEM_TO_PER_DMAC): Transfer Type is memory to peripheral and Flow Controller is DW_axi_dmac • 0x2 (PER_TO_MEM_DMAC): Transfer Type is peripheral to memory and Flow Controller is DW_axi_dmac • 0x3 (PER_TO_PER_DMAC): Transfer Type is peripheral to peripheral and Flow Controller is DW_axi_dmac • 0x4 (PER_TO_MEM_SRC): Transfer Type is peripheral to Memory and Flow Controller is Source peripheral • 0x5 (PER_TO_PER_SRC): Transfer Type is peripheral to peripheral and Flow Controller is Source peripheral • 0x6 (MEM_TO_PER_DST): Transfer Type is memory to peripheral and Flow Controller is Destination peripheral • 0x7 (PER_TO_PER_DST): Transfer Type is peripheral to peripheral and Flow Controller is Destination peripheral	0x0

To be continued ……

Table 11.24: CHx_CFG, Offset Address: 0x120 (continued)

Bits	Name	Access	Description	Reset
35	HS_SEL_SRC	R/W	Source Software or Hardware Handshaking Select. This register selects which of the handshaking interfaces (hardware or software) is active for source requests on this channel. • 0: Hardware handshaking interface. Software-initiated transaction requests are ignored. • 1: Software handshaking interface. Hardware-initiated transaction requests are ignored. If the source peripheral is memory, then this bit is ignored.	0x0
36	HS_SEL_DST	R/W	Destination Software or Hardware Handshaking Select. This register selects which of the handshaking interfaces (hardware or software) is active for destination requests on this channel. • 0: Hardware handshaking interface. Software-initiated transaction requests are ignored. • 1: Software handshaking interface. Hardware-initiated transaction requests are ignored. If the destination peripheral is memory, then this bit is ignored.	0x0
37	SRC_HWHS_POL	RO	Source Hardware Handshaking Interface Polarity. • 0: ACTIVE HIGH • 1: ACTIVE LOW	
38	DST_HWHS_POL	RO	Destination Hardware Handshaking Interface Polarity. • 0: ACTIVE HIGH • 1: ACTIVE LOW	
39	SRC_PER	R/W	Assigns a hardware handshaking interface (0 - DMAX_NUM_HS_IF-1) to the source of Channelx if the CHx_CFG.HS_SEL_SRC field is 0; otherwise, this field is ignored. The channel can then communicate with the source peripheral connected to that interface through the assigned hardware handshaking interface. Reset Value = 1 Note: For correct DW_axi_dmac operation, only one peripheral (source or destination) should be assigned to the same handshaking interface.	0x0
43:40	Reserved			
44	DST_PER	R/W	Assigns a hardware handshaking interface (0 - DMAX_NUM_HS_IF-1) to the destination of Channelx if the CHx_CFG.HS_SEL_DST field is 0; otherwise, this field is ignored. The channel can then communicate with the destination peripheral connected to that interface through the assigned hardware handshaking interface. Reset Value = 1 Note: For correct DW_axi_dmac operation, only one peripheral (source or destination) should be assigned to the same handshaking interface.	0x0

To be continued ……

Table 11.25: CHx_CFG, Offset Address: 0x120 (continued)

Bits	Name	Access	Description	Reset
48:45	Reserved			
51:49	CH_PRIOR	R/W	Channel Priority A priority of 7 is the highest priority, and 0 is the lowest. This field must be programmed within the following range: 0: DMAX_NUM_CHANNELS-1 A programmed value outside this range will cause erroneous behavior.	0x0
52	LOCK_CH	R/W	Channel Lock bit When the channel is granted control of the master bus interface and if the CHx_CFG.LOCK_CH bit is asserted, then no other channels are granted control of the master bus interface for the duration specified in CHx_CFG.LOCK_CH_L. Indicates to the master bus interface arbiter that this channel wants exclusive access to the master bus interface for the duration specified in CHx_CFG.LOCK_CH_L. This field does not exist if the configuration parameter DMAX_CHx_LOCK_EN is set to False; in this case, the read-back value is always 0. Locking the channel locks AXI Read Address, Write Address and Write Data channels on the corresponding master interface. Note: Channel locking feature is supported only for memoryto- memory transfer at Block Transfer and DMA Transfer levels. Hardware does not check for the validity of channel locking setting, hence the software must take care of enabling the channel locking only for memory-to-memory transfers at Block Transfer or DMA Transfer levels. Illegal programming of channel locking might result in unpredictable behavior.	0x0
54:53	LOCK_CH_L	R/W	Channel Lock Level This bit indicates the duration over which CHx_CFG.LOCK_CH bit applies. • 00: Over complete DMA transfer • 01: Over DMA block transfer • 1x: Reserved This field does not exist if the configuration parameter DMAX_CHx_LOCK_EN is set to False; in that case, the read-back value is always 0.	0x0
58:55	SRC_OSR_LMT	R/W	Source Outstanding Request Limit Maximum outstanding request supported is 16. Source Outstanding Request Limit = SRC_OSR_LMT + 1	0x0
62:59	DST_OSR_LMT	R/W	Destination Outstanding Request Limit Maximum outstanding request supported is 16. Source Outstanding Request Limit = DST_OSR_LMT + 1	0x0
63	Reserved			

11.5.16 CHx_LLP

Table 11.26: CHx_LLP, Offset Address: 0x128

Bits	Name	Access	Description	Reset
0	LMS	R/W	LLI master Select This bit identifies the AXI layer/interface where the memory device that stores the next linked list item resides. • 0: AXI Master 1 • 1: AXI Master 2 This field does not exist if the configuration parameter DMAX_CHx_LMS is not set to NO_HARDCODE.	0x0
5:1	Reserved			
63:6	LOC	R/W	Starting Address Memory of LLI block Starting Address In Memory of next LLI if block chaining is enabled. The six LSBs of the starting address are not stored because the address is assumed to be aligned to a 64-byte boundary. LLI access always uses the burst size (arsize/awsize) that is same as the data bus width and cannot be changed or programmed to anything other than this. Burst length (awlen/arlen) is chosen based on the data bus width so that the access does not cross one complete LLI structure of 64 bytes. DW_axi_dmac will fetch the entire LLI (40 bytes) in one AXI burst if the burst length is not limited by other settings.	0x0

11.5.17 CHx_STATUSREG

Table 11.27: CHx_STATUSREG, Offset Address: 0x130

Bits	Name	Access	Description	Reset
21:0	CMPLTD_BLK_TFR_SIZE	RO	Completed Block Transfer Size. This bit indicates the total number of data of width CHx_CTL.SRC_TR_WIDTH transferred for the previous block transfer.	
31:22	Reserved			
46:32	DATA_LEFT_IN_FIFO	RO	Data Left in FIFO. This bit indicates the total number of data left in DW_axi_dmac channel FIFO after completing the current block transfer.	
63:47	Reserved			

11.5.18 CHx_SWHSSRCREG

Table 11.28: CHx_SWHSSRCREG, Offset Address: 0x138

Bits	Name	Access	Description	Reset
0	SWHS_REQ_SRC	R/W	Software Handshake Request for Channel Source. This bit is used to request dma source data transfer if software handshaking method is selected for the source of the corresponding channel. This bit is ignored if software handshaking is not enabled for the source of the Channelx. The functionality of this field depends on whether the peripheral is the flow controller or not.	0x0
1	SWHS_REQ_SRC_WE	WO	Write Enable bit for Software Handshake Request for Channel Source.	0x0
2	SWHS_SGLREQ_SRC	R/W	Software Handshake Single Request for Channel Source. This bit is used to request SINGLE (AXI burst length = 1) dma source data transfer if software handshaking method is selected for the source of the corresponding channel. This bit is ignored if software handshaking is not enabled for the source of the Channelx. The functionality of this field depends on whether the peripheral is the flow controller.	0x0
3	SWHS_SGLREQ_SRC_WE	WO	Write Enable bit for Software Handshake Single Request for Channel Source.	0x0
4	SWHS_LST_SRC	R/W	Software Handshake Last Request for Channel Source. This bit is used to request LAST dma source data transfer if software handshaking method is selected for the source of the corresponding channel. This bit is ignored if software handshaking is not enabled for the source of the Channelx or if the source of Channelx is not the flow controller.	0x0
5	SWHS_LST_SRC_WE	WO	Write Enable bit for Software Handshake Last Request for Channel Source.	0x0
31:6	Reserved			

11.5.19 CHx_SWHSDSTREG

Table 11.29: CHx_SWHSDSTREG, Offset Address: 0x140

Bits	Name	Access	Description	Reset
0	SWHS_REQ_DST	R/W	Software Handshake Request for Channel Destination. This bit is used to request dma destination data transfer if software handshaking method is selected for the destination of the corresponding channel.	0x0
1	SWHS_REQ_DST_WE	WO	Write Enable bit for Software Handshake Request for Channel Destination.	0x0
2	SWHS_SGLREQ_DST	R/W	Software Handshake Single Request for Channel Destination. This bit is used to request SINGLE (AXI burst length = 1) dma destination data transfer if software handshaking method is selected for the destination of the corresponding channel.	0x0
3	SWHS_SGLREQ_DST_WE	WO	Write Enable bit for Software Handshake Single Request for Channel Destination.	0x0
4	SWHS_LST_DST	R/W	Software Handshake Last Request for Channel Destination. This bit is used to request LAST dma destination data transfer if software handshaking method is selected for the destination of the corresponding channel.	0x0
5	SWHS_LST_DST_WE	WO	Write Enable bit for Software Handshake Last Request for Channel Destination.	0x0
31:6	Reserved			

11.5.20 CHx_BLK_TFR_RESUMEREQREG

Table 11.30: CHx_BLK_TFR_RESUMEREQREG, Offset Address: 0x148

Bits	Name	Access	Description	Reset
0	BLK_TFR_RESUMEREQ	WO	Block Transfer Resume Request during Linked-List or Shadow-Register-based multi-block transfer.	0x0
31:1	Reserved			

11.5.21 CHx_AXI_IDREG

Table 11.31: CHx_AXI_IDREG, Offset Address: 0x150

Bits	Name	Access	Description	Reset
14:0	AXI_READ_ID_SUFFIX	R/W	AXI Read ID Suffix These bits form part of the ARID output of AXI3/AXI4 master interface.	0x0
15	Reserved			
3 0:16	AXI_WRITE_ID_SUFFIX	R/W	AXI Write ID Suffix. These bits form part of the AWID output of AXI3/AXI4 master interface.	0x0
31	Reserved			

11.5.22 CHx_AXI_QOSREG

Table 11.32: CHx_AXI_QOSREG, Offset Address: 0x158

Bits	Name	Access	Description	Reset
3:0	AXI_AWQOS	R/W	AXI AWQOS. These bits form the awqos output of AXI4 master interface.	0x0
7:4	AXI_ARQOS	R/W	AXI ARQOS. These bits form the arqos output of AXI4 master interface.	0x0
31:8	Reserved			

11.5.23 CHx_SSTAT

Table 11.33: CHx_SSTAT, Offset Address: 0x160

Bits	Name	Access	Description	Reset
31:0	SSTAT	RO	Source Status Source status information retrieved by hardware from the address pointed to by the contents of the CHx_SSTATAR register.	

11.5.24 CHx_DSTAT

Table 11.34: CHx_DSTAT, Offset Address: 0x168

Bits	Name	Access	Description	Reset
31:0	DSTAT	RO	Destination Status Destination status information retrieved by hardware from the address pointed to by the contents of the CHx_DSTATAR register.	

11.5.25 CHx_SSTATAR

Table 11.35: CHx_SSTATAR, Offset Address: 0x170

Bits	Name	Access	Description	Reset
63:0	SSTATAR	R/W	Source Status Fetch Address Pointer from where hardware can fetch the source status information, which is registered in the CHx_SSTAT register and written out to the CHx_SSTAT register location of the LLI before the start of the next block if DMAX_CHx_LLI_WB_EN = 1 and linked list based multi-block transfer is enabled for either source or destination peripheral of the channel. Source peripheral should update the source status information, if any, at the location pointed to by CHx_SSTATAR to utilize this feature.	0x0

11.5.26 CHx_DSTATAR

Table 11.36: CHx_DSTATAR, Offset Address: 0x178

Bits	Name	Access	Description	Reset
63:0	DSTATAR	R/W	Destination Status Fetch Address Pointer from where hardware can fetch the Destination status information, which is registered in the CHx_DSTAT register and written out to the CHx_DSTAT register location of the LLI before the start of the next block if DMAX_CHx_LLI_WB_EN = 1 and linked list based multiblock transfer is enabled for either source or destination peripheral of the channel. Destination peripheral should update the destination status information, if any, at the location pointed to by CHx_DSTATAR to utilize this feature.	0x0

11.5.27 CHx_INTSTATUS_ENABLEREG

Table 11.37: CHx_INTSTATUS_ENABLEREG, Offset Address: 0x180

Bits	Name	Access	Description	Reset
0	Enable_BLOCK_TFR_DONE_IntStat	R/W	Block Transfer Done Interrupt Status Enable. • 0: Disable the generation of Block Transfer Done Interrupt in CHx_INTSTATUSREG • 1: Enable the generation of Block Transfer Done Interrupt in CHx_INTSTATUSREG	0x0
1	Enable_DMA_TFR_DONE_IntStat	R/W	DMA Transfer Done Interrupt Status Enable. • 0: Disable the generation of DMA Transfer Done Interrupt in CHx_INTSTATUSREG • 1: Enable the generation of DMA Transfer Done Interrupt in CHx_INTSTATUSREG	0x0
2	Reserved			
3	Enable_SRC_TRANSCOMP_IntStat	R/W	Source Transaction Completed Status Enable. • 0: Disable the generation of Source Transaction Complete Interrupt in CHx_INTSTATUSREG • 1: Enable the generation of Source Transaction Complete Interrupt in CHx_INTSTATUSREG	0x0
4	Enable_DST_TRANSCOMP_IntStat	R/W	Destination Transaction Completed Status Enable. • 0: Disable the generation of Destination Transaction complete Interrupt in CHx_INTSTATUSREG • 1: Enable the generation of Destination Transaction complete Interrupt in CHx_INTSTATUSREG	0x0

To be continued ……

Table 11.38: CHx_INTSTATUS_ENABLEREG, Offset Address: 0x180 (continued)

Bits	Name	Access	Description	Reset
5	Enable_SRC_DEC_ERR_IntStat	R/W	Source Decode Error Status Enable. • 0: Disable the generation of Source Decode Error Interrupt in CHx_INTSTATUSREG • 1: Enable the generation of Source Decode Error Interrupt in CHx_INTSTATUSREG	0x0
6	Enable_DST_DEC_ERR_IntStat	R/W	Destination Decode Error Status Enable. • 0: Disable the generation of Destination Decode Error Interrupt in CHx_INTSTATUSREG • 1: Enable the generation of Destination Decode Error Interrupt in CHx_INTSTATUSREG	0x0
7	Enable_SRC_SLV_ERR_IntStat	R/W	Source Slave Error Status Enable. • 0: Disable the generation of Source Slave Error Interrupt in CHx_INTSTATUSREG • 1: Enable the generation of Source Slave Error Interrupt in CHx_INTSTATUSREG	0x0
8	Enable_DST_SLV_ERR_IntStat	R/W	Destination Slave Error Status Enable. • 0: Disable the generation of Destination Slave Error Interrupt in CHx_INTSTATUSREG • 1: Enable the generation of Destination Slave Error Interrupt in CHx_INTSTATUSREG	0x0

To be continued ……

Table 11.39: CHx_INTSTATUS_ENABLEREG, Offset Address: 0x180 (continued)

Bits	Name	Access	Description	Reset
9	Enable_LLI_RD_DEC_ERR_IntStat	R/W	LLI Read Decode Error Status Enable. • 0: Disable the generation of LLI Read Decode Error Interrupt in CHx_INTSTATUSREG • 1: Enable the generation of LLI Read Decode Error Interrupt in CHx_INTSTATUSREG	0x0
10	Enable_LLI_WR_DEC_ERR_IntStat	R/W	LLI WRITE Decode Error Status Enable. • 0: Disable the generation of LLI WRITE Decode Error Interrupt in CHx_INTSTATUSREG • 1: Enable the generation of LLI WRITE Decode Error Interrupt in CHx_INTSTATUSREG	0x0
11	Enable_LLI_RD_SLV_ERR_IntStat	R/W	LLI Read Slave Error Status Enable. • 0: Disable the generation of LLI Read Slave Error Interrupt in CHx_INTSTATUSREG • 1: Enable the generation of LLI Read Slave Error Interrupt in CHx_INTSTATUSREG	0x0
12	Enable_LLI_WR_SLV_ERR_IntStat	R/W	LLI WRITE Slave Error Status Enable. • 0: Disable the generation of LLI WRITE Slave Error Interrupt in CHx_INTSTATUSREG • 1: Enable the generation of LLI WRITE Slave Error Interrupt in CHx_INTSTATUSREG	0x0

To be continued ……

Table 11.40: CHx_INTSTATUS_ENABLEREG, Offset Address: 0x180 (continued)

Bits	Name	Access	Description	Reset
13	Enable_SHADOWREG_OR_LLI_INVALID_ERR_IntStat	R/W	Shadow register or LLI Invalid Error Status Enable. • 0: Disable the generation of Shadow Register or LLI Invalid Error Interrupt in CHx_INTSTATUSREG • 1: Enable the generation of Shadow Register or LLI Invalid Error Interrupt in CHx_INTSTATUSREG	0x0
14	Enable_SLVIF_MULTIBLKTYPE_ERR_IntStat	R/W	Slave Interface Multi Block type Error Status Enable. • 0: Disable the generation of Slave Interface Multi Block type Error Interrupt in CHx_INTSTATUSREG • 1: Enable the generation of Slave Interface Multi Block type Error Interrupt in CHx_INTSTATUSREG	0x0
15	Reserved			
16	Enable_SLVIF_DEC_ERR_IntStat	R/W	Slave Interface Decode Error Status Enable. • 0: Disable the generation of Slave Interface Decode Error Interrupt in CHx_INTSTATUSREG • 1: Enable the generation of Slave Interface Decode Error Interrupt in CHx_INTSTATUSREG	0x0
17	Enable_SLVIF_WR2RO_ERR_IntStat	R/W	Slave Interface Write to Read Only Error Status Enable. • 0: Disable the generation of Slave Interface Write to Read only Error Interrupt in CHx_INTSTATUSREG • 1: Enable the generation of Slave Interface Write to Read Only Error Interrupt in CHx_INTSTATUSREG	0x0

To be continued ……

Table 11.41: CHx_INTSTATUS_ENABLEREG, Offset Address: 0x180 (continued)

Bits	Name	Access	Description	Reset
18	Enable_SLVIF_RD2RWO_ERR_IntStat	R/W	Slave Interface Read to write Only Error Status Enable. • 0: Disable the generation of Slave Interface Read to Write only Error Interrupt in CHx_INTSTATUSREG • 1: Enable the generation of Slave Interface Read to Write Only Error Interrupt in CHx_INTSTATUSREG	0x0
19	Enable_SLVIF_WRONCHEN_ERR_IntStat	R/W	Slave Interface Write On Channel Enabled Error Status Enable. • 0: Disable the generation of Slave Interface Write On Channel enabled Error Interrupt in CHx_INTSTATUSREG • 1: Enable the generation of Slave Interface Write On Channel enabled Error Interrupt in CHx_INTSTATUSREG	0x0
20	Enable_SLVIF_SHADOWREG_WRON_VALID_ERR_IntStat	R/W	Shadow Register Write On Valid Error Status Enable. • 0: Disable the generation of Shadow Register Write On Valid Error Interrupt in CHx_INTSTATUSREG • 1: Enable the generation of Shadow register Write On Valid Error Interrupt in CHx_INTSTATUSREG	0x0
21	Enable_SLVIF_WRONHOLD_ERR_IntStat	R/W	Slave Interface Write On Hold Error Status Enable. • 0: Disable the generation of Slave Interface Write On Hold Error Interrupt in CHx_INTSTATUSREG • 1: Enable the generation of Slave Interface Write On Hold Error Interrupt in CHx_INTSTATUSREG	0x0

To be continued ……

Table 11.42: CHx_INTSTATUS_ENABLEREG, Offset Address: 0x180 (continued)

Bits	Name	Access	Description	Reset
26:22	Reserved			
27	Enable_CH_LOCK_CLEARED_IntStat	R/W	Channel Lock Cleared Status Enable. • 0: Disable the generation of Channel LOCK CLEARED Interrupt in CHx_INTSTATUSREG • 1: Enable the generation of Channel LOCK CLEARED Interrupt in CHx_INTSTATUSREG	0x0
28	Enable_CH_SRC_SUSPENDED_IntStat	R/W	Channel Source Suspended Status Enable. • 0: Disable the generation of Channel Source Suspended Interrupt in CHx_INTSTATUSREG • 1: Enable the generation of Channel Source Suspended Interrupt in CHx_INTSTATUSREG	0x0
29	Enable_CH_SUSPENDED_IntStat	R/W	Channel Suspended Status Enable. • 0: Disable the generation of Channel Suspended Interrupt in CHx_INTSTATUSREG • 1: Enable the generation of Channel Suspended Interrupt in CHx_INTSTATUSREG	0x0
30	Enable_CH_DISABLED_IntStat	R/W	Channel Disabled Status Enable. • 0: Disable the generation of Channel Disabled Interrupt in CHx_INTSTATUSREG • 1: Enable the generation of Channel Disabled Interrupt in CHx_INTSTATUSREG	0x0
31	Enable_CH_ABORTED_IntStat	R/W	Channel Aborted Status Enable. • 0: Disable the generation of Channel Aborted Interrupt in CHx_INTSTATUSREG • 1: Enable the generation of Channel Aborted Interrupt in CHx_INTSTATUSREG	0x0

11.5.28 CHx_INTSTATUS

Table 11.43: CHx_INTSTATUS, Offset Address: 0x188

Bits	Name	Access	Description	Reset
0	BLOCK_TFR_DONE_IntStat	RO	Block Transfer Done. This indicates to the software that the DW_axi_dmac has completed the requested block transfer. The DW_axi_dmac sets this bit to 1 when the transfer is successfully completed. • 0: Block Transfer not completed. • 1: Block Transfer completed. This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in CHx_IntClearReg register.	
1	DMA_TFR_DONE_IntStat	RO	DMA Transfer Done. This indicates to the software that the DW_axi_dmac has completed the requested DMA transfer. The DW_axi_dmac sets this bit to 1 along with setting CHx_INTSTATUS.BLOCK_TFR_DONE bit to 1 when the last block transfer is completed. • 0: DMA Transfer not completed. • 1: DMA Transfer Completed This bit is cleared to 0 on writing 1	
2	Reserved			
3	SRC_TRANSCOMP_IntStat	RO	Source Transaction Completed. This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in CHx_IntClearReg register or on enabling the channel (needed when interrupt is not enabled.	
4	DST_TRANSCOMP_IntStat	RO	Destination Transaction Completed. This bit is cleared to 0 on writing 1 to the corresponding channel interrupt clear bit in CHx_IntClearReg register or on enabling the channel (needed when interrupt is not enabled.	

To be continued ……

Table 11.44: CHx_INTSTATUS, Offset Address: 0x188 (continued)

Bits	Name	Access	Description	Reset
5	SRC_DEC_ERR_IntStat	RO	Source Decode Error. Decode Error detected by Master Interface during source data transfer. This error occurs if the access is to invalid address and a Decode Error is returned from interconnect/slave. This error condition causes the DW_axi_dmac to disable the corresponding channel gracefully; the DMAC_ChEnReg.CH_EN bit corresponding to the channel which received the error is set to 0. • 0: No Source Decode Errors. • 1: Source Decode Error detected.	
6	DST_DEC_ERR_IntStat	RO	Destination Decode Error. Decode Error detected by Master Interface during destination data transfer. This error occurs if the access is to invalid address and a Decode Error is returned from interconnect/slave. This error condition causes the DW_axi_dmac to disable the corresponding channel gracefully; the DMAC_ChEnReg.CH_EN bit corresponding to the channel which received the error is set to 0. • 0: No destination Decode Errors. • 1: Destination Decode Error Detected	
7	SRC_SLV_ERR_IntStat	RO	Source Slave Error. Slave Error detected by Master Interface during source data transfer. This error occurs if the slave interface from which the data is read issues a Slave Error. This error condition causes the DW_axi_dmac to disable the corresponding channel gracefully; the DMAC_ChEnReg.CH_EN bit corresponding to the channel which received the error is set to 0. • 0: No Source Slave Errors • 1: Source Slave Error Detected	

To be continued ……

Table 11.45: CHx_INTSTATUS, Offset Address: 0x188 (continued)

Bits	Name	Access	Description	Reset
8	DST_SLV_ERR_IntStat	RO	Destination Slave Error. Slave Error detected by Master Interface during destination data transfer. This error occurs if the slave interface to which the data is written issues a Slave Error. This error condition causes the DW_axi_dmac to disable the corresponding channel gracefully; the DMAC_ChEnReg.CH_EN bit corresponding to the channel which received the error is set to 0. • 0: No Destination Slave Errors • 1: Destination Slave Errors Detected	
9	LLI_RD_DEC_ERR_IntStat	RO	LLI Read Decode Error. Decode Error detected by Master Interface during LLI read operation. This error occurs if the access is to invalid address and a Decode Error is returned from interconnect/slave. This error condition causes the DW_axi_dmac to disable the corresponding channel gracefully; the DMAC_ChEnReg.CH_EN1 bit which received the error is set to 0. • 0: NO LLI Read Decode Errors. • 1: LLI Read Decode Error detected	
10	LLI_WR_DEC_ERR_IntStat	RO	LLI WRITE Decode Error. Decode Error detected by Master Interface during LLI writeback operation. This error occurs if the access is to invalid address and a Decode Error is returned from interconnect/slave. This error condition causes the DW_axi_dmac to disable the corresponding channel gracefully; the DMAC_ChEnReg.CH_EN1 bit which received the error is set to 0. • 0: NO LLI Write Decode Errors. • 1: LLI write Decode Error detected.	

To be continued ……

Table 11.46: CHx_INTSTATUS, Offset Address: 0x188 (continued)

Bits	Name	Access	Description	Reset
11	LLI_RD_SLV_ERR_IntStat	RO	LLI Read Slave Error. Slave Error detected by Master Interface during LLI read operation. This error occurs if the slave interface on which LLI resides issues a Slave Error. This error condition causes the DW_axi_dmac to disable the corresponding channel gracefully; the DMAC_ChEnReg.CH_EN1 bit which received the error is set to 0. • 0: No LLI Read Slave Errors. • 1: LLI read Slave Error detected.	
12	LLI_WR_SLV_ERR_IntStat	RO	LLI WRITE Slave Error. Slave Error detected by Master Interface during LLI writeback operation. This error occurs if the slave interface on which LLI resides issues a Slave Error. This error condition causes the DW_axi_dmac to disable the corresponding channel gracefully; the DMAC_ChEnReg.CH_EN1 bit which received the error is set to 0. • 0: No LLI write Slave Errors. • 1: LLI Write SLAVE Error detected.	

To be continued ……

Table 11.47: CHx_INTSTATUS, Offset Address: 0x188 (continued)

Bits	Name	Access	Description	Reset
13	SHADOWREG_OR_LLI _INVALID_ERR_IntStat	RO	Shadow register or LLI Invalid Error. This error occurs if CHx_CTL. ShadowReg_Or_LLI_Valid bit is seen to be 0 during DW_axi_dmac Shadow Register / LLI fetch phase. This error condition causes the DW_axi_dmac to halt the corresponding channel gracefully; Error Interrupt is generated if the corresponding channel error interrupt mask bit is set to 0 and the channel waits till software writes (any value) to CHx_BLK_TFR_ResumeReqReg to indicate valid Shadow Register availability. In the case of LLI pre-fetching, ShadowReg_Or_LLI_Invalid_ERR Interrupt is not generated even if ShadowReg_Or_LLI_Valid bit is seen to be 0 for the pre-fetched LLI. In this case, DW_axi_dmac re-attempts the LLI fetch operation after completing the current block transfer and generates Shadow Reg_Or_LLI_Invalid_ERR Interrupt only if ShadowReg_Or_LLI_Valid bit is still seen to be 0. • 0: No Shadow Register / LLI Invalid errors. • 1: Shadow Register / LLI Invalid error detected.	
14	SLVIF_MULT IBLKTYPE_ERR_IntStat	RO	Slave Interface Multi Block type Error. This error occurs if multi-block transfer type programmed in CHx_CFG register (SRC_MLTBLK_TYPE and DST_MLTBLK_TYPE) is invalid. This error condition causes the DW_axi_dmac to halt the corresponding channel gracefully; Error Interrupt is generated if the corresponding channel error interrupt mask bit is set to 0 and the channel waits till software writes (any value) to CHx_BLK_TFR_ResumeReqReg to indicate valid multiblock transfer type availability. • 0: No Multi-block transfer type Errors. • 1: Multi-block transfer type Error detected.	

To be continued ……

Table 11.48: CHx_INTSTATUS, Offset Address: 0x188 (continued)

Bits	Name	Access	Description	Reset
15	Reserved			
16	SLVIF_DEC_ERR_IntStat	RO	Slave Interface Decode Error. Decode Error generated by DW_axi_dmac during register access. This error occurs if the register access is to invalid address in Channelx register space resulting in error response by DW_axi_dmac slave interface. • 0: No Slave Interface Decode errors. • 1: Slave Interface Decode Error detected.	
17	SLVIF_WR2RO_ERR_IntStat	RO	Slave Interface Write to Read Only Error. This error occurs if write operation is performed to a Read Only register. • 0: No Slave Interface Write to Read Only Errors. • 1: Slave Interface Write to Read Only Error detected.	
18	SLVIF_RD2RWO_ERR_IntStat	RO	Slave Interface Read to write Only Error. This error occurs if read operation is performed to a Write Only register. • 0: No Slave Interface Read to Write Only Errors. • 1: Slave Interface Read to Write Only Error detected.	
19	SLVIF_WRONCHEN_ERR_IntStat	RO	Slave Interface Write On Channel Enabled Error. This error occurs if an illegal write operation is performed on a register; this happens if a write operation is performed on a register when the channel is enabled and if it is not allowed for the corresponding register as per the DW_axi_dmac specification. • 0: No Slave Interface Write On Channel Enabled Errors. • 1: Slave Interface Write On Channel Enabled Error detected.	

To be continued ……

Table 11.49: CHx_INTSTATUS, Offset Address: 0x188 (continued)

Bits	Name	Access	Description	Reset
20	SLVIF_SHADOWREG_WRON_VALID_ERR_IntStat	RO	Shadow Register Write On Valid Error. This error occurs if shadow register based multi-block transfer is enabled and software tries to write to the shadow register when CHx_CTL.ShadowReg_Or_LLI_Valid bit is 1. • 0: No Slave Interface Shadow Register Write On Valid Errors. • 1: Slave Interface Shadow Register Write On Valid Error detected.	
21	SLVIF_WRONHOLD_ERR_IntStat	RO	Slave Interface Write On Hold Error. This error occurs if an illegal write operation is performed on a register; this happens if a write operation is performed on a channel register when DW_axi_dmac is in Hold mode. • 0: No Slave Interface Write On Hold Errors. • 1: Slave Interface Write On Hold Error detected.	
26:22	Reserved			
27	CH_LOCK_CLEARED_IntStat	RO	Channel Lock Cleared. This indicates to the software that the locking of the corresponding channel in DW_axi_dmac is cleared. • 0: Channel locking is not cleared. • 1: Channel locking is cleared.	
28	CH_SRC_SUSPENDED_IntStat	RO	Channel Source Suspended. This indicates to the software that the corresponding channel source data transfer in DW_axi_dmac is suspended. • 0: Channel source is not suspended • 1: Channel Source is suspended.	
29	CH_SUSPENDED_IntStat	RO	Channel Suspended. This indicates to the software that the corresponding channel in DW_axi_dmac is suspended. • 0: Channel is not suspended. • 1: Channel is suspended.	

To be continued ……

Table 11.50: CHx_INTSTATUS, Offset Address: 0x188 (continued)

Bits	Name	Access	Description	Reset
30	CH_DISABLED_IntStat	RO	Channel Disabled. This indicates to the software that the corresponding channel in DW_axi_dmac is disabled. • 0: Channel is not disabled. • 1: Channel is disabled. Error Interrupt is generated if the corresponding bit in CHx_INTSTATUS_ENABLEReg is enabled.	
31	CH_ABORTED_IntStat	RO	Channel Aborted. This indicates to the software that the corresponding channel in DW_axi_dmac is aborted. • 0: Channel is not aborted • 1: Channel is aborted	

11.5.29 CHx_INTSIGNAL_ENABLEREG

Table 11.51: CHx_INTSIGNAL_ENABLEREG, Offset Address: 0x190

Bits	Name	Access	Description	Reset
0	Enable_BLOCK_TFR_DONE_IntSignal	R/W	Block Transfer Done Interrupt Signal Enable. • 0: Disable the propagation of Block Transfer Done Interrupt to generate a port level interrupt • 1: Enable the propagation of Block Transfer Done Interrupt to generate a port level interrupt	0x0
1	Enable_DMA_TFR_DONE_IntSignal	R/W	DMA Transfer Done Interrupt Signal Enable. • 0: Disable the propagation of DMA Transfer Done Interrupt to generate a port level interrupt • 1: Enable the propagation of DMA Transfer Done Interrupt to generate a port level interrupt	0x0
2	Reserved			
3	Enable_SRC_TRANSCOMP_IntSignal	R/W	Source Transaction Completed Signal Enable. • 0: Disable the propagation of Source Transaction Complete Interrupt to generate a port level interrupt • 1: Enable the propagation of Source Transaction Complete Interrupt to generate a port level interrupt	0x0
4	Enable_DST_TRANSCOMP_IntSignal	R/W	Destination Transaction Completed Signal Enable. • 0: Disable the propagation of Destination Transaction complete Interrupt to generate a port level interrupt • 1: Enable the propagation of Destination Transaction complete Interrupt to generate a port level interrupt	0x0
5	Enable_SRC_DEC_ERR_IntSignal	R/W	Source Decode Error Signal Enable. • 0: Disable the propagation of Source Decode Error Interrupt to generate a port level interrupt • 1: Enable the propagation of Source Decode Error Interrupt to generate a port level interrupt	0x0

To be continued ……

Table 11.52: CHx_INTSIGNAL_ENABLEREG, Offset Address: 0x190 (continued)

Bits	Name	Access	Description	Reset
6	Enable_DST_DEC_ERR_IntSignal	R/W	Destination Decode Error Signal Enable. • 0: Disable the propagation of Destination Decode Error Interrupt to generate a port level interrupt • 1: Enable the propagation of Destination Decode Error Interrupt to generate a port level interrupt	0x0
7	Enable_SRC_SLV_ERR_IntSignal	R/W	Source Slave Error Signal Enable. • 0: Disable the propagation of Source Slave Error Interrupt to generate a port level interrupt • 1: Enable the propagation of Source Slave Error Interrupt to generate a port level interrupt	0x0
8	Enable_DST_SLV_ERR_IntSignal	R/W	Destination Slave Error Signal Enable. • 0: Disable the propagation of Destination Slave Error Interrupt to generate a port level interrupt • 1: Enable the propagation of Destination Slave Error Interrupt to generate a port level interrupt	0x0
9	Enable_LLI_RD_DEC_ERR_IntSignal	R/W	LLI Read Decode Error Signal Enable. • 0: Disable the propagation of LLI Read Decode Error Interrupt to generate a port level interrupt • 1: Enable the propagation of LLI Read Decode Error Interrupt to generate a port level interrupt	0x0
10	Enable_LLI_WR_DEC_ERR_IntSignal	R/W	LLI WRITE Decode Error Signal Enable. • 0: Disable the propagation of LLI WRITE Decode Error Interrupt to generate a port level interrupt • 1: Enable the propagation of LLI WRITE Decode Error Interrupt to generate a port level interrupt	0x0
11	Enable_LLI_RD_SLV_ERR_IntSignal	R/W	LLI Read Slave Error Signal Enable. • 0: Disable the propagation of LLI Read Slave Error Interrupt to generate a port level interrupt • 1: Enable the propagation of LLI Read Slave Error Interrupt to generate a port level interrupt	0x0

To be continued ……

Table 11.53: CHx_INTSIGNAL_ENABLEREG, Offset Address: 0x190 (continued)

Bits	Name	Access	Description	Reset
12	Enable_LLI_ WR_SLV_ERR_IntSignal	R/W	LLI WRITE Slave Error Signal Enable. • 0: Disable the propagation of LLI WRITE Slave Error Interrupt to generate a port level interrupt • 1: Enable the propagation of LLI WRITE Slave Error Interrupt to generate a port level interrupt	0x0
13	Enabl e_SHADOWREG_OR_LLI_I NVALID_ERR_IntSignal	R/W	Shadow register or LLI Invalid Error Signal Enable. • 0: Disable the propagation of Shadow Register or LLI Invalid Error Interrupt to generate a port level interrupt • 1: Enable the propagation of Shadow Register or LLI Invalid Error Interrupt to generate a port level interrupt	0x0
14	Enable_SLVIF_MULTIB LKTYPE_ERR_IntSignal	R/W	Slave Interface Multi Block type Error Signal Enable. • 0: Disable the propagation of Slave Interface Multi Block type Error Interrupt to generate a port level interrupt • 1: Enable the propagation of Slave Interface Multi Block type Error Interrupt to generate a port level interrupt	0x0
15	Reserved			
16	Enable_SLV IF_DEC_ERR_IntSignal	R/W	Slave Interface Decode Error Signal Enable. • 0: Disable the propagation of Slave Interface Decode Error Interrupt to generate a port level interrupt • 1: Enable the propagation of Slave Interface Decode Error Interrupt to generate a port level interrupt	0x0
17	Enable_SLVIF _WR2RO_ERR_IntSignal	R/W	Slave Interface Write to Read Only Error Signal Enable. • 0: Disable the propagation of Slave Interface Write to Read only Error Interrupt to generate a port level interrupt • 1: Enable the propagation of Slave Interface Write to Read Only Error Interrupt to generate a port level interrupt	0x0

To be continued ……

Table 11.54: CHx_INTSIGNAL_ENABLEREG, Offset Address: 0x190 (continued)

Bits	Name	Access	Description	Reset
18	Enable_SLVIF_ RD2RWO_ERR_IntSignal	R/W	Slave Interface Read to write Only Error Signal Enable. • 0: Disable the propagation of Slave Interface Read to Write only Error Interrupt to generate a port level interrupt • 1: Enable the propagation of Slave Interface Read to Write Only Error Interrupt to generate a port level interrupt	0x0
19	Enable_SLVIF_WR ONCHEN_ERR_IntSignal	R/W	Slave Interface Write On Channel Enabled Error Signal Enable. • 0: Disable the propagation of Slave Interface Write On Channel enabled Error Interrupt to generate a port level interrupt • 1: Enable the propagation of Slave Interface Write On Channel enabled Error Interrupt to generate a port level interrupt	0x0
20	Enable_ SLVIF_SHADOWREG_WRON _VALID_ERR_IntSignal	R/W	Shadow Register Write On Valid Error Signal Enable. • 0: Disable the propagation of Shadow Register Write On Valid Error Interrupt to generate a port level interrupt • 1: Enable the propagation of Shadow register Write On Valid Error Interrupt to generate a port level interrupt	0x0
21	Enable_SLVIF_WR ON-HOLD_ERR_IntSignal	R/W	Slave Interface Write On Hold Error Signal Enable. • 0: Disable the propagation of Slave Interface Write On Hold Error Interrupt to generate a port level interrupt • 1: Enable the propagation of Slave Interface Write On Hold Error Interrupt to generate a port level interrupt	0x0
26:22	Reserved			

To be continued ……

Table 11.55: CHx_INTSIGNAL_ENABLEREG, Offset Address: 0x190 (continued)

Bits	Name	Access	Description	Reset
27	Enable_CH_LO CK_CLEARED_IntSignal	R/W	Channel Lock Cleared Signal Enable. • 0: Disable the propagation of Channel Lock Cleared Interrupt to generate a port level interrupt • 1: Enable the propagation of Channel Lock Cleared Interrupt to generate a port level interrupt	0x0
28	Enable_CH_SRC _SUSPENDED_IntSignal	R/W	Channel Source Suspended Signal Enable. • 0: Disable the propagation of Channel Source Suspended Interrupt to generate a port level interrupt • 1: Enable the propagation of Channel Source Suspended Interrupt to generate a port level interrupt	0x0
29	Enable_CH _SUSPENDED_IntSignal	R/W	Channel Suspended Signal Enable. • 0: Disable the propagation of Channel Suspended Interrupt to generate a port level interrupt • 1: Enable the propagation of Channel Suspended Interrupt to generate a port level interrupt	0x0
30	Enable_C H_DISABLED_IntSignal	R/W	Channel Disabled Signal Enable. • 0: Disable the propagation of Channel Disabled Interrupt to generate a port level interrupt • 1: Enable the propagation of Channel Disabled Interrupt to generate a port level interrupt	0x0
31	Enable_ CH_ABORTED_IntSignal	R/W	Channel Aborted Signal Enable. • 0: Disable the propagation of Channel Aborted Interrupt to generate a port level interrupt • 1: Enable the propagation of Channel Aborted Interrupt to generate a port level interrupt	0x0

11.5.30 CHx_INTCLEARREG

Table 11.56: CHx_INTCLEARREG, Offset Address: 0x198

Bits	Name	Access	Description	Reset
0	Clear_BL OCK_TFR_DONE_IntStat	WO	Block Transfer Done Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CH1_INTSTATUSREG	0x0
1	Clear_ DMA_TFR_DONE_IntStat	WO	DMA Transfer Done Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
2	Reserved			
3	Clear_S RC_TRANSCOMP_IntStat	WO	Source Transaction Completed Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
4	Clear_D ST_TRANSCOMP_IntStat	WO	Destination Transaction Completed Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
5	Clear _SRC_DEC_ERR_IntStat	WO	Source Decode Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
6	Clear _DST_DEC_ERR_IntStat	WO	Destination Decode Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
7	Clear _SRC_SLV_ERR_IntStat	WO	Source Slave Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
8	Clear _DST_SLV_ERR_IntStat	WO	Destination Slave Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
9	Clear_LL I_RD_DEC_ERR_IntStat	WO	LLI Read Decode Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
10	Clear_LL I_WR_DEC_ERR_IntStat	WO	LLI WRITE Decode Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0

To be continued ……

Table 11.57: CHx_INTCLEARREG, Offset Address: 0x198 (continued)

Bits	Name	Access	Description	Reset
11	Clear_LL I_RD_SLV_ERR_IntStat	WO	LLI Read Slave Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
12	Clear_LL I_WR_SLV_ERR_IntStat	WO	LLI WRITE Slave Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
13	Cl ear_SHADOWREG_OR_LLI _INVALID_ERR_IntStat	WO	Shadow register or LLI Invalid Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
14	Clear_SLVIF_MULT IBLKTYPE_ERR_IntStat	WO	Slave Interface Multi Block type Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
15	Reserved			
16	Clear_S LVIF_DEC_ERR_IntStat	WO	Slave Interface Decode Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
17	Clear_SLV IF_WR2RO_ERR_IntStat	WO	Slave Interface Write to Read Only Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
18	Clear_SLVI F_RD2RWO_ERR_IntStat	WO	Slave Interface Read to write Only Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
19	Clear_SLVIF_ WRONCHEN_ERR_IntStat	WO	Slave Interface Write On Channel Enabled Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
20	Clea r_SLVIF_SHADOWREG_WR ON_VALID_ERR_IntStat	WO	Shadow Register Write On Valid Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0

To be continued ……

Table 11.58: CHx_INTCLEARREG, Offset Address: 0x198 (continued)

Bits	Name	Access	Description	Reset
21	Clear_SLVIF_ WRONHOLD_ERR_IntStat	WO	Slave Interface Write On Hold Error Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
26:22	Reserved			
27	Clear_CH_ LOCK_CLEARED_IntStat	WO	Channel Lock Cleared Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
28	Clear_CH_S RC_SUSPENDED_IntStat	WO	Channel Source Suspended Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
29	Clear_ CH_SUSPENDED_IntStat	WO	Channel Suspended Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
30	Clear _CH_DISABLED_IntStat	WO	Channel Disabled Interrupt Clear Bit. This bit is used to clear the corresponding channel interrupt status bit in CHx_INTSTATUSREG.	0x0
31	Reserved			

CHAPTER

TWELVE

TIMERS

12.1 Overview

The system is configured with 8 Timer modules providing timing and counting function. These timers can be used to implement timing and counting for applications, and can also be used for the operating system to implement the system clock.

12.2 Features

The timer has the following characteristics:

- 32bit subtraction timer/counter.
- Supports 2 counting modes: free running mode and user-defined counting mode.
- The system can read the current count value.

When the count decreases to 0, an interrupt is generated.

12.3 Function Description

Timer is based on a 32-bit down counter. The counter value is decremented by 1 on each rising edge of the count clock. When the count value decreases to zero, the Timer will generate an interrupt.

Timer has the following 2 counting modes:

- Free running mode: The timer continues to count, and when the count value decreases to 0, it automatically returns to its maximum value and continues counting. The maximum count length is 0xFFFF_FFFF.
- User-defined counting mode: The timer continues to count. When the count value decreases to 0, the initial value is loaded again from the TimerNLoadCount (N=1~8) register and continues counting.

The method of loading the initial count value of the timer is as follows:

The initial count value of the timer can be loaded by writing the TimerNLoadCount (N=1~8) register.

12.4 Way of Working

12.4.1 Initialization

- Step 1: Write the TimerNLoadCount (N=1~8) register to load the initial count value for the Timer.
- Step 2: Set the TimerNControlReg[2:0] (N=1~8) register, select the Timer counting mode, mask the Timer interrupt, and start the Timer to start counting.

12.4.2 Interrupt Handling

When the timer generates an interrupt, the steps are as follows:

- Step 1: Read the TimerNEOI (N=1~8) register to clear the TimerN interrupt.
- Step 2: Execute the process waiting for the interrupt.
- Step 3: After the process is completed, resume execution of the interrupted program.

12.4.3 Clock Selection

System Timer optional 25MHz/32KHz counting clock. Use reg_timer_clk_sel to make the selection.

12.5 Timer Register Overview

Timer registers are accessed via the bus.

An overview of Timer registers is shown in *Timer Registers Overview (Base: 0x030A0000)*.

Table 12.1: Timer Registers Overview (Base: 0x030A0000)

Name	Address Offset	Description
Timer1LoadCount	0x000	Value to be loaded into Timer1
Timer1CurrentValue	0x004	Current Value of Timer1
Timer1ControlReg	0x008	Control Register for Timer1
Timer1EOI	0x00c	Clears the interrupt from Timer1
Timer1IntStatus	0x010	Contains the interrupt status for Timer1
Timer2 Registers	0x014~ 0x024	There are 5 registers in total, the contents are the same as Timer1.
Timer3 Registers	0x028~ 0x038	There are 5 registers in total, the contents are the same as Timer1.
Timer4 Registers	0x03c~ 0x04c	There are 5 registers in total, the contents are the same as Timer1.
Timer5 Registers	0x050~ 0x060	There are 5 registers in total, the contents are the same as Timer1.
Timer6 Registers	0x064~ 0x074	There are 5 registers in total, the contents are the same as Timer1.
Timer7 Registers	0x078~ 0x088	There are 5 registers in total, the contents are the same as Timer1.
Timer8 Registers	0x08c~ 0x09c	There are 5 registers in total, the contents are the same as Timer1.
TimersIntStatus	0x0a0	Contains the interrupt status of all timers in the component.
TimersEOI	0x0a4	Returns all zeroes (0) and clears all active interrupts.
TimersRawIntStatus	0x0a8	Contains the unmasked interrupt status of all timers in the component.

12.6 Timer Register Description

12.6.1 Timer1LoadCount

Table 12.2: Timer1LoadCount, Offset Address: 0x000

Bits	Name	Access	Description	Reset
31:0	Timer1LoadCount	R/W	Timer1 Load Count Register Value to be loaded into Timer1. This is the value from which counting commences. Any value written to this register is loaded into the associated timer.	0x0

12.6.2 Timer1CurrentValue

Table 12.3: Timer1CurrentValue, Offset Address: 0x004

Bits	Name	Access	Description	Reset
31:0	Timer1CurrentValue	RO	Timer1 Current Value Current Value of Timer1. This register is supported only when timer_1_clk is synchronous to pclk. Reading this register when using independent clocks results in an undefined value.	

12.6.3 Timer1ControlReg

Table 12.4: Timer1ControlReg, Offset Address: 0x008

Bits	Name	Access	Description	Reset
2:0	Timer1ControlReg	R/W	[2] Timer interrupt mask for Timer1 0 - not masked 1 - masked [1] Timer mode for Timer1 0 - free-running mode 1 - user-defined count mode [0] Timer enable bit for Timer1 0 - disable 1 - enable	0x0
31:3	Reserved			

12.6.4 Timer1EOI

Table 12.5: Timer1EOI, Offset Address: 0x00c

Bits	Name	Access	Description	Reset
0	Timer1EOI	RO	Reading from this register returns all zeroes (0) and clears the interrupt from Timer1.	
31:1	Reserved			

12.6.5 Timer1IntStatus

Table 12.6: Timer1IntStatus, Offset Address: 0x010

Bits	Name	Access	Description	Reset
0	Timer1IntStatus	RO	Contains the interrupt status for Timer1.	
31:1	Reserved			

12.6.6 TimersIntStatus

Table 12.7: TimersIntStatus, Offset Address: 0x0a0

Bits	Name	Access	Description	Reset
7:0	TimersIntStatus	RO	Contains the interrupt status of all timers. Reading from this register does not clear any active interrupts: 0 – either timer_intr or timer_intr_n is not active after masking 1 – either timer_intr or timer_intr_n is active after masking	
31:8	Reserved			

12.6.7 TimersEOI

Table 12.8: TimersEOI, Offset Address: 0x0a4

Bits	Name	Access	Description	Reset
7:0	TimersEOI	RO	Reading this register returns all zeroes (0) and clears all active interrupts.	
31:8	Reserved			

12.6.8 TimersRawIntStatus

Table 12.9: TimersRawIntStatus, Offset Address: 0x0a8

Bits	Name	Access	Description	Reset
7:0	TimersRawIntStatus	RO	The register contains the unmasked interrupt status of all timers. 0 - either timer_intr or timer_intr_n is not active prior to masking 1 - either timer_intr or timer_intr_n is active prior to masking	
31:8	Reserved			

CHAPTER

THIRTEEN

WATCHDOG

13.1 Overview

The system is configured with 4 watchdog modules. It is used to issue an interrupt or reset signal after a certain period of time to interrupt or reset the entire system when an abnormality occurs in the system.

13.2 Features

WatchDog has the following features:

- Configure a 32bit down counter.
- Supports configurable initial counting value (i.e. time interval).
- Supports WatchDog restart protection to prevent WatchDog from being restarted due to misoperation.
- Support reset signal generation.
- Support timeout interrupt generation.

13.3 Function Description

The system configures register parameter values for WatchDog through the system bus. WatchDog periodically sends WDT_INTR interrupt requests to the system, and when the system does not respond to the interrupt (such as crashes), it sends a WDT_SYS_RST reset signal to reset the system to achieve the purpose of monitoring system operation.

13.3.1 Application Block Diagram

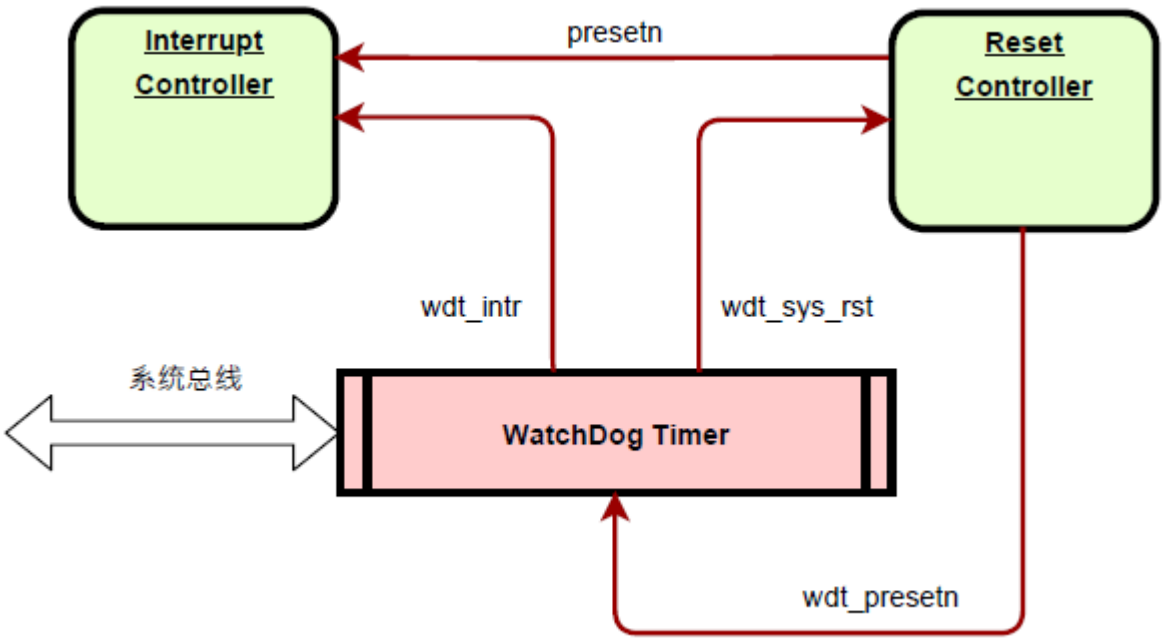


Diagram 13.1: WatchDog Application Block Diagram

13.3.2 Functional Principle

WatchDog runs based on a 32-bit down counter. Its initial count value has two sources, loaded from WDT_ITORR and WDT_TORR respectively, and calculated based on the value of ITOR_MODE (for specific calculation methods, refer to the description in *WDT_TORR, Offset Address: 0x004*). WDT_ITORR is used for the first timer count of WatchDog after power-on, and subsequent timer counts are based on WDT_TORR.

When the WatchDog clock is enabled, the count value is decremented by 1 on the rising edge of each count clock. When the count value decreases to 0, WatchDog will generate an interrupt. Then at the next rising edge of the counting clock, the counter reloads the initial counting value from the register WDT_TORR and starts counting down.

The user can set the register WDT_CR[1] to decide whether to send a reset signal WDT_SYS_RST immediately when the counter count value decreases to 0 for the first time. If it is set to 0, a reset signal is sent immediately. Otherwise, if it is 1, an interrupt is generated and the second count starts. If the CPU has not cleared the WatchDog interrupt when the second count decreases to 0, WatchDog will send a reset signal WDT_SYS_RST and the counter will stop counting.

13.4 Way of Working

13.4.1 Count Clock Frequency Configuration

WatchDog counting clock is 25MHz/32KHz clock. Use reg_wdt_clk_sel to make the selection.

13.4.2 System Initialization Configuration

After the system is powered on and reset, the WatchDog counter stops counting. During the system initialization process, WatchDog needs to be initialized and started to run. The initialization process of WatchDog is as follows:

- Step 1: Write register WDT_TORR to set the initial value of WatchDog count.
- Step 2: Write register WDT_CR[1] to set WatchDog count timeout response mode.
- Step 3: Write register WDT_CR[0] to start WatchDog counting.

13.4.3 Interrupt Handling Process

After the system receives an interrupt from WatchDog, it should clear its interrupt status in time.

The process of WatchDog interrupt processing is as follows:

- Step 1: Read the register WDT_EOI and clear the interrupt status of WatchDog.
- Step 2: Write 0x76 to the register WDT_CRR and restart WatchDog.

13.4.4 Close WatchDog

Write 0 or 1 to register WDT_CR[0] to control the status of WatchDog:

- 0: WDT is off.
- 1: WDT is on. Only a system reset can turn off the WDT after startup.

13.5 WDT Register Overview

The WDT register is accessed via the bus. Includes four WDTs, 3 are Active Domains, and 1 is No-die Domain. Their base addresses are:

- WDT0: 0x03010000
- WDT1: 0x03011000
- WDT2: 0x03012000
- RTCSYS_WDT: 0x0502D000

Each WDT consists of a set of control registers, each set has the same definition, and an overview is shown in *WDT Registers Overview*.

Table 13.1: WDT Registers Overview

Name	Address Offset	Description
WDT_CR	0x000	Control register
WDT_TORR	0x004	Timeout range register
WDT_CCVR	0x008	Current counter value register
WDT_CRR	0x00c	Counter restart register
WDT_STAT	0x010	Interrupt status register
WDT_EOI	0x014	Interrupt clear register
WDT_TOC	0x01C	Time Out Count

13.6 WDT Register Description

13.6.1 WDT_CR

Table 13.2: WDT_CR, Offset Address: 0x000

Bits	Name	Access	Description	Reset
4:0	WDT_CR	R/W	[4:2] Reset pulse length. This is used to select the number of pclk cycles for which the system reset stays asserted. The range of values available is 2 to 256 pclk cycles. 000 - 2 pclk cycles 001 - 4 pclk cycles 010 - 8 pclk cycles 011 - 16 pclk cycles 100 - 32 pclk cycles 101 - 64 pclk cycles 110 - 128 pclk cycles 111 - 256 pclk cycles [1] Response mode. Selects the output response generated to a time-out. 0 = Generate a system reset. 1 = First generate an interrupt and if it is not cleared by the time a second timeout occurs then generate a system reset. [0] WDT enable. This bit is used to enable and disable the Watch-Dog. When disabled, the counter does not decrement. Thus, no interrupts or system resets are generated. Once this bit has been enabled, it can be cleared only by a system reset. 0 = WDT disabled. 1 = WDT enabled.	0x0
5	Reserved			
6	TOR_MODE	R/W	The Mode of Timeout Period	0x0
7	ITOR_MODE	R/W	The Mode of Timeout Period for initialization	0x0
31:8	Reserved			

13.6.2 WDT_TORR

Table 13.3: WDT_TORR, Offset Address: 0x004

Bits	Name	Access	Description	Reset
3:0	WDT_TORR	R/W	[3:0] TOP(TimeOut Period). This field is used to select the timeout period from which the watchdog counter restarts. A change of the timeout period takes effect only after the next counter restart (kick). The range of values is limited by 32-bit width. If TOP is programmed to select a range that is greater than the counter width, the timeout period is truncated to fit to the counter width. This affects only the non-user specified values as users are limited to these boundaries during configuration. The range of values available for a 32-bit watchdog counter are: TOR_MODE = 0 T = 2^(16 + WDT_TORR) TOR_MODE = 1 T = WDT_TOC <<(WDT_TORR +1)	0x0
7:4	WDT_ITORR		Initial TimeOut Period ITOR_MODE = 0 T = 2^(16 + WDT_ITORR) ITOR_MODE = 1 T = WDT_TOC <<(WDT_ITORR +1)	
31:8	Reserved			

13.6.3 WDT_CCVR

Table 13.4: WDT_CCVR, Offset Address: 0x008

Bits	Name	Access	Description	Reset
31:0	WDT_CCVR	RO	This register, when read, is the current value of the internal counter.	

13.6.4 WDT_CRR

Table 13.5: WDT_CRR, Offset Address: 0x00c

Bits	Name	Access	Description	Reset
7:0	WDT_CRR	R/W	[7:0] Counter Restart Register This register is used to restart the WDT counter. As a safety feature to prevent accidental restarts, the value 0x76 must be written. A restart also clears the WDT interrupt. Reading this register returns zero.	0x0
31:8	Reserved			

13.6.5 WDT_STAT

Table 13.6: WDT_STAT, Offset Address: 0x010

Bits	Name	Access	Description	Reset
0	WDT_STAT	RO	[0] Interrupt Status Register This register shows the interrupt status of the WDT. 1 = Interrupt is active regardless of polarity. 0 = Interrupt is inactive.	
31:1	Reserved			

13.6.6 WDT_EOI

Table 13.7: WDT_EOI, Offset Address: 0x014

Bits	Name	Access	Description	Reset
0	WDT_EOI	RO	[0] Interrupt Clear Register Clears the watchdog interrupt. This can be used to clear the interrupt without restarting the watchdog counter.	
31:1	Reserved			

13.6.7 WDT_TOC

Table 13.8: WDT_TOC, Offset Address: 0x01C

Bits	Name	Access	Description	Reset
15:0	WDT_TOC	R/W	Time out counter	0x0

CHAPTER

FOURTEEN

8051 SUBSYSTEM

14.1 Overview

The 8051 subsystem is located within RTCSYS and is independently powered. System software can use the 8051 to manage wake conditions and wake the system while the system is asleep, and communicate with external devices through peripheral controllers.

14.2 Features

The 8051 microprocessor has the following features:

- Supports standard 8051 instruction set.
- Frequency range: 25MHz ~ 300 MHz.
- Debugging function: single step execution/jump2pc/snapshot PSW, DPTR, PC.
- Support 32bit data access.
- Reset vector configurable to system AHB SRAM/DRAM/SPINOR.
- Support WFI (Clock gating).
- Supports code banking (max 64x64 KB).

14.3 Way of Working

14.3.1 Power Domain Control Process

The RTCSYS subsystem is divided into two power domains: AO (Always-On) domain and MCU domain. The MCU domain can be powered on and off through the configuration register. The process is as follows:

MCU power_off:

1. Soft reset register = 0
2. Set register reg_mcu_iso_en = 1
3. Set register reg_mcu_pwr_req = 0

MCU power_on:

1. Set register reg_mcu_pwr_req = 1
2. Polling register reg_mcu_pwr_ack = 1
3. Set register reg_mcu_iso_en = 0
4. Soft reset register = 1

14.3.2 8051 Initialization

8051 is in reset state at the beginning of the system. Using 8051, you can complete the following software process through ACPU:

1. The 8051 reset state is in reset state (register reg_soft_rstn_mcu = 0).
2. Configure the register reg_mcu_rom_addr_size to determine the instruction TCM size.
3. Configure the register reg_51irom_ioffset to determine the location on the AHB SRAM where TCM is executed.
4. Configure register reg_soft_rstn_mcu =1 to release the 8051 reset state.

14.3.3 Interrupt Handling

8051 can receive external level-triggered interrupts through the int0_n and int1_n interfaces. int0_n/int1_n selects to output interrupt signals to 8051 from ictl (interrupt control) and configuration register reg_51_int1_src_mask respectively.

Table 14.1: 8051 subsystem interrupt list

Interrupt No.	Interrupt Name	Description
0	Vbat_det	System power-down Interrupt
1	mbox_int0	Mailbox Interrupt
2	NA	Reserved
3	irrx	IRRX Receive Interrupt
4	gpio_int	PWR GPIO Interrupt
5	uart_int	PWR UART Interrupt
6	spinor1_int	SPINOR1 Interrupt
7	timer_int0	TIMER0 Interrupt
8	timer_int1	TIMER1 Interrupt
9	Irq_ap2rtc[0]	System Interrupt
10	Irq_ap2rtc[1]	System Interrupt
11	I2c_int	PWR I2C Interrupt
12	st_change_int	RTC Status Change Interrupt
13	hw_thm_shdn	System overheating Interrupt
14	saradc	SARADC Interrupt
15	wdt_int	WatchDog Interrupt

14.3.4 MAILBOX

Mailbox provides 2 sets of spinlock function fields and 4 sets of 32bit information fields, allowing ACPU/8051 to transmit information to each other.

14.4 8051 Subsystem Register Overview

The registers of the 8051 subsystem are defined in the RTC CTRL register. For details, refer to the RTC_CTRL_REG section of the *RTC Register Overview* chapter.

14.5 8051 Subsystem Register Description

The registers of the 8051 subsystem are defined in the RTC CTRL register, please refer to the *RTC_CTRL_REG* section for details.

CHAPTER

FIFTEEN

DDR CONTROLLER

15.1 Overview

The DDR controller implements data access to dynamic memory (DRAM). It converts the data access commands of each master device in the SoC into dynamic memory commands that comply with the JEDEC standard, and schedules them appropriately, thereby improving the use efficiency of the dynamic memory bus.

15.2 Features

Function Features:

- Support:
  - DDR2 maximum data rate 1333 Mbps.
  - DDR3 maximum data rate 1866 Mbps.
- Supports interface data width 16-bit.
- Supports single channel, single rank.
- Support automatic refresh control.
- Support priority control.
- Support data traffic statistics.
- Support low power consumption mode.
- Support address mapping.
- Support pin multiplexing.

15.3 Function Description

15.3.1 Application Block Diagram

The DRAM interface supports 16-bit data width and 32-bit data width. *SoC/DRAM interconnection diagram* is the interconnection diagram between the main chip and the single-chip DRAM device:

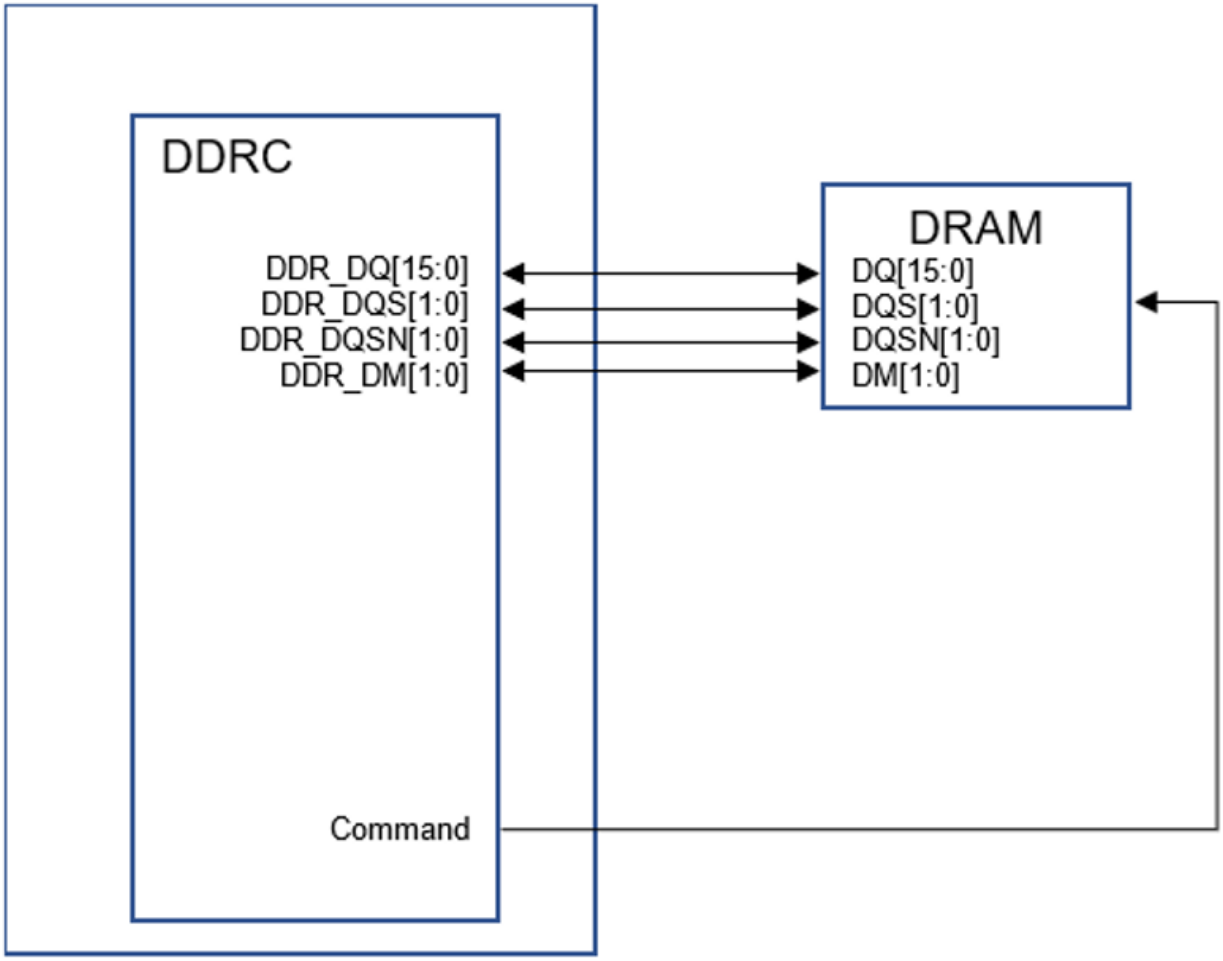


Diagram 15.1: SoC/DRAM interconnection diagram

Command is consists of several signals, which vary depending on the DRAM type. Table *DDR2/DDR3 command signal comparison table* compares command signals for DDR2 and DDR3:

Table 15.1: DDR2/DDR3 command signal comparison table

Function	DDR2	DDR3
DDR CKE	D2x_CKE	D3x_CKE
DDR CLKN	D2x_CK_N	D3x_CK_N
DDR CLKP	D2x_CK_P	D3x_CK_P
DDR CSB	D2x_CS_N	D3x_CS_N
DDR RESETN	D2x_RESET_N	D3x_RESET_N
DDR RASN	D2x_RAS_N	D3x_RAS_N
DDR CASN	D2x_CAS_N	D3x_CAS_N
DDR WEN	D2x_WE_N	D3x_WE_N
DDR ACTN	N/A	N/A
DDR BA	D2x_BAn	D3x_BAn
DDR BG	N/A	N/A
DDR MA	D2x_An (n = 0 - 15)	D3x_An (n = 0 - 15)
DDR ODT	D2x_ODT	D3x_ODT
Note: Pin names may vary slightly depending on the package.		

15.3.2 Functional Principle

Based on the storage characteristics of DRAM, JEDEC developed a set of standards that specify the commands and timing required to access DRAM data and control DRAM status. By properly configuring the DDR register, the DDR controller can issue timing sequences that meet JEDEC standards to complete actions such as reading, writing, and low-power control.

15.3.2.1 Command Truth Table

The DDR interface timing meets the JDEDC standard. Following are table *DDR2 Command Truth Table* and table *DDR3 Command Truth Table*. They are the truth tables of DDR2 and DDR3 support commands for user reference. The rest of the information can refer to the JEDEC standard.

Table 15.2: DDR2 Command Truth Table

Function	CKE Pre	CKE Cur	CS#	RAS#	CAS#	WE#	BA0 - BA2	A11 - A15	A10 / AP	A0 - A9
Mode Register Set	H	H	L	L	L	L	BA	OP		
Refresh	H	H	L	L	L	H	V	V	V	V
Self Refresh Entry	H	L	L	L	L	H	V	V	V	V
Self Refresh Exit	L	H	H	X	X	X	X	X	X	X
			L	H	H	H	V	V	V	V
Single Bank Precharge	H	H	L	L	H	L	BA	V	L	V
Precharge all Banks	H	H	L	L	H	L	V	V	H	V
Bank Activate	H	H	L	L	H	H	BA	RA		
Write	H	H	L	H	L	L	BA	RFU	L	CA
Write with Auto Precharge	H	H	L	H	L	L	BA	RFU	H	CA
Read	H	H	L	H	L	H	BA	RFU	L	CA
Read with Auto Precharge	H	H	L	H	L	H	BA	RFU	H	CA
Read with Auto Precharge	H	H	L	H	L	H	BA	RFU	H	CA
No Operation	H	H	L	H	H	H	V	V	V	V
Device Deselected	H	H	H	X	X	X	X	X	X	X
Power Down Entry	H	L	L	H	H	H	V	V	V	V
			H	X	X	X	X	X	X	X
Power Down Exit	L	H	L	H	H	H	V	V	V	V
			H	X	X	X	X	X	X	X
H: High level; L: Low level; V: Valid; X: N/A. RFU: Reserved for Future Usage.										

Table 15.3: DDR3 Command Truth Table

Function	CKE Pre	CKE Cur	CS#	RAS#	CAS#	WE#	BA0 - BA2	A13 - A15	A12 / BC#	A10 / AP	A0 - A9,11
Mode Register Set	H	H	L	L	L	L	BA	OP			
Refresh	H	H	L	L	L	H	V	V	V	V	V
Self Refresh Entry	H	L	L	L	L	H	V	V	V	V	V
Self Refresh Exit	L	H	H	X	X	X	X	X	X	X	X
			L	H	H	H	V	V	V	V	V
Single Bank Precharge	H	H	L	L	H	L	BA	V	V	L	V
Precharge all Banks	H	H	L	L	H	L	V	V	V	H	V
Bank Activate	H	H	L	L	H	H	BA	RA			
Write (Fixed BL8 or BC4)	H	H	L	H	L	L	BA	RFU	V	L	CA
Write (BC4, on the Fly)	H	H	L	H	L	L	BA	RFU	L	L	CA
Write (BL8, on the Fly)	H	H	L	H	L	L	BA	RFU	H	L	CA
Write with Auto Precharge (Fixed BL8 or BC4)	H	H	L	H	L	L	BA	RFU	V	H	CA
Write with Auto Precharge (BC4, on the Fly)	H	H	L	H	L	L	BA	RFU	L	H	CA
Write with Auto Precharge (BL8, on the Fly)	H	H	L	H	L	L	BA	RFU	H	H	CA
Read (Fixed BL8 or BC4)	H	H	L	H	L	H	BA	RFU	V	L	CA
Read (BC4, on the Fly)	H	H	L	H	L	H	BA	RFU	L	L	CA
Read (BL8, on the Fly)	H	H	L	H	L	H	BA	RFU	H	L	CA
Read with Auto Precharge (Fixed BL8 or BC4)	H	H	L	H	L	H	BA	RFU	V	H	CA
Read with Auto Precharge (BC4, on the Fly)	H	H	L	H	L	H	BA	RFU	L	H	CA
Read with Auto Precharge (BL8, on the Fly)	H	H	L	H	L	H	BA	RFU	H	H	CA
No Operation	H	H	L	H	H	H	V	V	V	V	V
Device Deselected	H	H	H	X	X	X	X	X	X	X	X
Power Down Entry	H	L	L	H	H	H	V	V	V	V	V
			H	X	X	X	X	X	X	X	X
Power Down Exit	L	H	L	H	H	H	V	V	V	V	V
			H	X	X	X	X	X	X	X	X
ZQ Calibration Long	H	H	L	H	H	L	X	X	X	H	X
ZQ Calibration Short	H	H	L	H	H	L	X	X	X	L	X
H: High level; L: Low level; V: Valid; X: N/A. RFU: Reserved for Future Usage.											

15.3.2.2 Auto Refresh

The DDR controller has the ability to control the automatic refresh function. The purpose of controlling automatic refresh is to reduce the delay in accessing data or reduce the impact of refresh commands on DRAM bandwidth. Try to send refresh commands when the DRAM is idle. The specific available methods are:

- Refresh at equal intervals: Send a refresh command every tREFI time.
- Tricky refresh: The DDR controller internally counts the number of expired tREFIs, and then uses idle time to continuously send them.

15.3.2.3 Low Power Management

DDR controller supports low power modes:

- Normal low-power mode: Set an idle time through the register. When the normal low-power mode is enabled and the DDR controller meets the idle time, it automatically controls the DRAM to enter the normal low-power mode.
- Self-refresh mode: Self-refresh is a mode with lower power consumption. An idle time is set through the register. When the self-refresh mode is enabled and the DDR controller meets the idle time, it automatically controls the DRAM to enter the self-refresh mode.

15.3.2.4 Arbitration Mechanism

The DDR controller mainly optimizes the bandwidth usage of the system based on various DRAM control timings, and schedules each command through a priority scheduling algorithm. In addition, DDRC also implements two scheduling auxiliary means, timeout control and real-time control (these two control means are enabled according to business needs, and can be enabled at the same time or separately) to control command requests. .

- Consecutive address access restrictions

  The limit level is: 0 ~ 15 DRAM read/write instructions, and the configuration of each AXI port is independent. The DDR controller has high priority for contiguous addresses by default to optimize DRAM utilization. This mechanism limits the maximum length of contiguous DRAM that can be accessed by each AXI port.

- Priority scheduling

  The priority level is: 0 ~ 15. The higher the value, the higher the priority. The read/write priority configuration of each AXI port is independent.

- Timeout control

  For read/write transfers of each AXI port, the timeout register can be configured to avoid too long waiting. After the waiting time is reached, AXI ports that have not yet reached the waiting time or that have not been configured with the timeout attribute are forcibly blocked.

- Real-time control

  For real-time functions, the hardware buffering threshold can be configured. If the buffering is insufficient, the priority will be automatically raised to the highest level, and other AXI ports can be restricted from generating new transmissions.

15.3.2.5 Traffic Statistics and Command-latency Statistics

The DDR controller supports traffic statistics function: it can count the read and write traffic of each AXI port to collect current traffic information to determine whether flow control is required. Statistics of overall DRAM read/write traffic can be collected.

The DDR controller supports the AXI latency statistics function and supports cumulative latency statistics for specified/unspecified transmissions.

15.3.2.6 Address Mapping Method

The DDR controller converts the access address of the system bus into the access address of the DRAM. Can implement RBC (row_bank_column), BRC, and support bank interleave in row/column bit.

15.4 Way of Working

15.4.1 Soft Reset

Soft reset is not supported.

15.4.2 DDR Initialization Configuration Process

This controller initialization process is provided in the form of a software package.

15.5 AXI Registers

15.5.1 AXI Register Overview

Table 15.4: AXI Registers, Base: 0x0800_4000

Name	Address Offset	Description
AXI_CTRL0_1	0x4b4	AXI1 read timeout control
AXI_CTRL1_1	0x4b8	AXI1 write timeout control
AXI_CTRL0_2	0x564	AXI2 read timeout control
AXI_CTRL1_2	0x568	AXI2 write timeout control
AXI_CTRL0_3	0x614	AXI3 read timeout control
AXI_CTRL1_3	0x618	AXI3 write timeout control

Table 15.5: AXI Registers, Base: 0x0800_8000

Name	Address Offset	Description
AXI_MON0_CTRL	0x000	AXI monitor 0 control
AXI_MON0_INPUT	0x004	AXI monitor 0 input selection
AXI_MON0_FILTER0	0x010	AXI monitor 0 filter settings
AXI_MON0_FILTER1	0x014	AXI monitor 0 filter settings
AXI_MON0_FILTER2	0x018	AXI monitor 0 filter settings
AXI_MON0_FILTER3	0x01c	AXI monitor 0 filter settings
AXI_MON0_FILTER4	0x020	AXI monitor 0 filter settings
AXI_MON0_FILTER5	0x024	AXI monitor 0 filter settings
AXI_MON0_FILTER6	0x028	AXI monitor 0 filter settings
AXI_MON0_FILTER7	0x02c	AXI monitor 0 filter settings
AXI_MON0_FILTER8	0x030	AXI monitor 0 filter settings
AXI_MON0_RPT0	0x040	AXI monitor 0 cycle count
AXI_MON0_RPT1	0x044	AXI monitor 0 hit count
AXI_MON0_RPT2	0x048	AXI monitor 0 byte count
AXI_MON0_RPT3	0x04c	AXI monitor 0 latency count
AXI_MON1_CTRL	0x080	AXI monitor 1 control
AXI_MON1_INPUT	0x084	AXI monitor 1 input selection
AXI_MON1_FILTER0	0x090	AXI monitor 1 filter settings
AXI_MON1_FILTER1	0x094	AXI monitor 1 filter settings
AXI_MON1_FILTER2	0x098	AXI monitor 1 filter settings
AXI_MON1_FILTER3	0x09c	AXI monitor 1 filter settings
AXI_MON1_FILTER4	0x0a0	AXI monitor 1 filter settings
AXI_MON1_FILTER5	0x0a4	AXI monitor 1 filter settings
AXI_MON1_FILTER6	0x0a8	AXI monitor 1 filter settings
AXI_MON1_FILTER7	0x0ac	AXI monitor 1 filter settings
AXI_MON1_FILTER8	0x0b0	AXI monitor 1 filter settings
AXI_MON1_RPT0	0x0c0	AXI monitor 1 cycle count
AXI_MON1_RPT1	0x0c4	AXI monitor 1 hit count
AXI_MON1_RPT2	0x0c8	AXI monitor 1 byte count
AXI_MON1_RPT3	0x0cc	AXI monitor 1 latency count
AXI_MON2_CTRL	0x100	AXI monitor 2 control
AXI_MON2_INPUT	0x104	AXI monitor 2 input selection
AXI_MON2_FILTER0	0x110	AXI monitor 2 filter settings
AXI_MON2_FILTER1	0x114	AXI monitor 2 filter settings

continues on next page

Table 15.5 – continued from previous page

Name	Address Offset	Description
AXI_MON2_FILTER2	0x118	AXI monitor 2 filter settings
AXI_MON2_FILTER3	0x11c	AXI monitor 2 filter settings
AXI_MON2_FILTER4	0x120	AXI monitor 2 filter settings
AXI_MON2_FILTER5	0x124	AXI monitor 2 filter settings
AXI_MON2_FILTER6	0x128	AXI monitor 2 filter settings
AXI_MON2_FILTER7	0x12c	AXI monitor 2 filter settings
AXI_MON2_FILTER8	0x130	AXI monitor 2 filter settings
AXI_MON2_RPT0	0x140	AXI monitor 2 cycle count
AXI_MON2_RPT1	0x144	AXI monitor 2 hit count
AXI_MON2_RPT2	0x148	AXI monitor 2 byte count
AXI_MON2_RPT3	0x14c	AXI monitor 2 latency count
AXI_MON3_CTRL	0x180	AXI monitor 3 control
AXI_MON3_INPUT	0x184	AXI monitor 3 input selection
AXI_MON3_FILTER0	0x190	AXI monitor 3 filter settings
AXI_MON3_FILTER1	0x194	AXI monitor 3 filter settings
AXI_MON3_FILTER2	0x198	AXI monitor 3 filter settings
AXI_MON3_FILTER3	0x19c	AXI monitor 3 filter settings
AXI_MON3_FILTER4	0x1a0	AXI monitor 3 filter settings
AXI_MON3_FILTER5	0x1a4	AXI monitor 3 filter settings
AXI_MON3_FILTER6	0x1a8	AXI monitor 3 filter settings
AXI_MON3_FILTER7	0x1ac	AXI monitor 3 filter settings
AXI_MON3_FILTER8	0x1b0	AXI monitor 3 filter settings
AXI_MON3_RPT0	0x1c0	AXI monitor 3 cycle count
AXI_MON3_RPT1	0x1c4	AXI monitor 3 hit count
AXI_MON3_RPT2	0x1c8	AXI monitor 3 byte count
AXI_MON3_RPT3	0x1cc	AXI monitor 3 latency count
AXI_MON4_CTRL	0x200	AXI monitor 4 control
AXI_MON4_INPUT	0x204	AXI monitor 4 input selection
AXI_MON4_FILTER0	0x210	AXI monitor 4 filter settings
AXI_MON4_FILTER1	0x214	AXI monitor 4 filter settings
AXI_MON4_FILTER2	0x218	AXI monitor 4 filter settings
AXI_MON4_FILTER3	0x21c	AXI monitor 4 filter settings
AXI_MON4_FILTER4	0x220	AXI monitor 4 filter settings
AXI_MON4_FILTER5	0x224	AXI monitor 4 filter settings
AXI_MON4_FILTER6	0x228	AXI monitor 4 filter settings
AXI_MON4_FILTER7	0x22c	AXI monitor 4 filter settings
AXI_MON4_FILTER8	0x230	AXI monitor 4 filter settings
AXI_MON4_RPT0	0x240	AXI monitor 4 cycle count
AXI_MON4_RPT1	0x244	AXI monitor 4 hit count
AXI_MON4_RPT2	0x248	AXI monitor 4 byte count
AXI_MON4_RPT3	0x24c	AXI monitor 4 latency count
AXI_MON5_CTRL	0x280	AXI monitor 5 control
AXI_MON5_INPUT	0x284	AXI monitor 5 input selection
AXI_MON5_FILTER0	0x290	AXI monitor 5 filter settings
AXI_MON5_FILTER1	0x294	AXI monitor 5 filter settings
AXI_MON5_FILTER2	0x298	AXI monitor 5 filter settings
AXI_MON5_FILTER3	0x29c	AXI monitor 5 filter settings
AXI_MON5_FILTER4	0x2a0	AXI monitor 5 filter settings
AXI_MON5_FILTER5	0x2a4	AXI monitor 5 filter settings

continues on next page

Table 15.5 – continued from previous page

Name	Address Offset	Description
AXI_MON5_FILTER6	0x2a8	AXI monitor 5 filter settings
AXI_MON5_FILTER7	0x2ac	AXI monitor 5 filter settings
AXI_MON5_FILTER8	0x2b0	AXI monitor 5 filter settings
AXI_MON5_RPT0	0x2c0	AXI monitor 5 cycle count
AXI_MON5_RPT1	0x2c4	AXI monitor 5 hit count
AXI_MON5_RPT2	0x2c8	AXI monitor 5 byte count
AXI_MON5_RPT3	0x2cc	AXI monitor 5 latency count
AXI_MON6_CTRL	0x300	AXI monitor 6 control
AXI_MON6_INPUT	0x304	AXI monitor 6 input selection
AXI_MON6_FILTER0	0x310	AXI monitor 6 filter settings
AXI_MON6_FILTER1	0x314	AXI monitor 6 filter settings
AXI_MON6_FILTER2	0x318	AXI monitor 6 filter settings
AXI_MON6_FILTER3	0x31c	AXI monitor 6 filter settings
AXI_MON6_FILTER4	0x320	AXI monitor 6 filter settings
AXI_MON6_FILTER5	0x324	AXI monitor 6 filter settings
AXI_MON6_FILTER6	0x328	AXI monitor 6 filter settings
AXI_MON6_FILTER7	0x32c	AXI monitor 6 filter settings
AXI_MON6_FILTER8	0x330	AXI monitor 6 filter settings
AXI_MON6_RPT0	0x340	AXI monitor 6 cycle count
AXI_MON6_RPT1	0x344	AXI monitor 6 hit count
AXI_MON6_RPT2	0x348	AXI monitor 6 byte count
AXI_MON6_RPT3	0x34c	AXI monitor 6 latency count
AXI_MON7_CTRL	0x380	AXI monitor 7 control
AXI_MON7_INPUT	0x384	AXI monitor 7 input selection
AXI_MON7_FILTER0	0x390	AXI monitor 7 filter settings
AXI_MON7_FILTER1	0x394	AXI monitor 7 filter settings
AXI_MON7_FILTER2	0x398	AXI monitor 7 filter settings
AXI_MON7_FILTER3	0x39c	AXI monitor 7 filter settings
AXI_MON7_FILTER4	0x3a0	AXI monitor 7 filter settings
AXI_MON7_FILTER5	0x3a4	AXI monitor 7 filter settings
AXI_MON7_FILTER6	0x3a8	AXI monitor 7 filter settings
AXI_MON7_FILTER7	0x3ac	AXI monitor 7 filter settings
AXI_MON7_FILTER8	0x3b0	AXI monitor 7 filter settings
AXI_MON7_RPT0	0x3c0	AXI monitor 7 cycle count
AXI_MON7_RPT1	0x3c4	AXI monitor 7 hit count
AXI_MON7_RPT2	0x3c8	AXI monitor 7 byte count
AXI_MON7_RPT3	0x3cc	AXI monitor 7 latency count
AXI_MON8_CTRL	0x400	AXI monitor 8 control
AXI_MON8_INPUT	0x404	AXI monitor 8 input selection
AXI_MON8_FILTER0	0x410	AXI monitor 8 filter settings
AXI_MON8_FILTER1	0x414	AXI monitor 8 filter settings
AXI_MON8_FILTER2	0x418	AXI monitor 8 filter settings
AXI_MON8_FILTER3	0x41c	AXI monitor 8 filter settings
AXI_MON8_FILTER4	0x420	AXI monitor 8 filter settings
AXI_MON8_FILTER5	0x424	AXI monitor 8 filter settings
AXI_MON8_FILTER6	0x428	AXI monitor 8 filter settings
AXI_MON8_FILTER7	0x42c	AXI monitor 8 filter settings
AXI_MON8_FILTER8	0x430	AXI monitor 8 filter settings
AXI_MON8_RPT0	0x440	AXI monitor 8 cycle count

continues on next page

Table 15.5 – continued from previous page

Name	Address Offset	Description
AXI_MON8_RPT1	0x444	AXI monitor 8 hit count
AXI_MON8_RPT2	0x448	AXI monitor 8 byte count
AXI_MON8_RPT3	0x44c	AXI monitor 8 latency count
AXI_MON9_CTRL	0x480	AXI monitor 9 control
AXI_MON9_INPUT	0x484	AXI monitor 9 input selection
AXI_MON9_FILTER0	0x490	AXI monitor 9 filter settings
AXI_MON9_FILTER1	0x494	AXI monitor 9 filter settings
AXI_MON9_FILTER2	0x498	AXI monitor 9 filter settings
AXI_MON9_FILTER3	0x49c	AXI monitor 9 filter settings
AXI_MON9_FILTER4	0x4a0	AXI monitor 9 filter settings
AXI_MON9_FILTER5	0x4a4	AXI monitor 9 filter settings
AXI_MON9_FILTER6	0x4a8	AXI monitor 9 filter settings
AXI_MON9_FILTER7	0x4ac	AXI monitor 9 filter settings
AXI_MON9_FILTER8	0x4b0	AXI monitor 9 filter settings
AXI_MON9_RPT0	0x4c0	AXI monitor 9 cycle count
AXI_MON9_RPT1	0x4c4	AXI monitor 9 hit count
AXI_MON9_RPT2	0x4c8	AXI monitor 9 byte count
AXI_MON9_RPT3	0x4cc	AXI monitor 9 latency count
AXI_MON10_CTRL	0x500	AXI monitor 10 control
AXI_MON10_INPUT	0x504	AXI monitor 10 input selection
AXI_MON10_FILTER0	0x510	AXI monitor 10 filter settings
AXI_MON10_FILTER1	0x514	AXI monitor 10 filter settings
AXI_MON10_FILTER2	0x518	AXI monitor 10 filter settings
AXI_MON10_FILTER3	0x51c	AXI monitor 10 filter settings
AXI_MON10_FILTER4	0x520	AXI monitor 10 filter settings
AXI_MON10_FILTER5	0x524	AXI monitor 10 filter settings
AXI_MON10_FILTER6	0x528	AXI monitor 10 filter settings
AXI_MON10_FILTER7	0x52c	AXI monitor 10 filter settings
AXI_MON10_FILTER8	0x530	AXI monitor 10 filter settings
AXI_MON10_RPT0	0x540	AXI monitor 10 cycle count
AXI_MON10_RPT1	0x544	AXI monitor 10 hit count
AXI_MON10_RPT2	0x548	AXI monitor 10 byte count
AXI_MON10_RPT3	0x54c	AXI monitor 10 latency count
AXI_MON11_CTRL	0x580	AXI monitor 11 control
AXI_MON11_INPUT	0x584	AXI monitor 11 input selection
AXI_MON11_FILTER0	0x590	AXI monitor 11 filter settings
AXI_MON11_FILTER1	0x594	AXI monitor 11 filter settings
AXI_MON11_FILTER2	0x598	AXI monitor 11 filter settings
AXI_MON11_FILTER3	0x59c	AXI monitor 11 filter settings
AXI_MON11_FILTER4	0x5a0	AXI monitor 11 filter settings
AXI_MON11_FILTER5	0x5a4	AXI monitor 11 filter settings
AXI_MON11_FILTER6	0x5a8	AXI monitor 11 filter settings
AXI_MON11_FILTER7	0x5ac	AXI monitor 11 filter settings
AXI_MON11_FILTER8	0x5b0	AXI monitor 11 filter settings
AXI_MON11_RPT0	0x5c0	AXI monitor 11 cycle count
AXI_MON11_RPT1	0x5c4	AXI monitor 11 hit count
AXI_MON11_RPT2	0x5c8	AXI monitor 11 byte count
AXI_MON11_RPT3	0x5cc	AXI monitor 11 latency count

15.5.2 AXI Register Description (Base address: 0x0800_4000)

15.5.2.1 AXI_CTRL0_1

Table 15.6: AXI_CTRL0_1, Offset Address: 0x4b4

Bits	Name	Access	Description	Reset
9:0	axi1_rd_timeout_val	R/W	After an AXI read transaction is granted, a timeout counter starts to count. When it counts to axi<n>_rd_timeout_val, the corresponding channel has the highest priority.	0x0
11:10	Reserved			
12	axi1_rd_timeout_en	R/W	If set to 1, enables the timeout function for the read channel of port n.	0x0
31:13	Reserved			

15.5.2.2 AXI_CTRL1_1

Table 15.7: AXI_CTRL1_1, Offset Address: 0x4b8

Bits	Name	Access	Description	Reset
9:0	axi1_wr_timeout_val	R/W	After an AXI write transaction is granted, a timeout counter starts to count. When it counts to axi<n>_wr_timeout_val, the corresponding channel has the highest priority.	0x0
11:10	Reserved			
12	axi1_wr_timeout_en	R/W	If set to 1, enables the timeout function for the write channel of port n.	0x0
31:13	Reserved			

15.5.2.3 AXI_CTRL0_2

Table 15.8: AXI_CTRL0_2, Offset Address: 0x564

Bits	Name	Access	Description	Reset
9:0	axi2_rd_timeout_val	R/W	After an AXI read transaction is granted, a timeout counter starts to count. When it counts to axi<n>_rd_timeout_val, the corresponding channel has the highest priority.	0x0
11:10	Reserved			
12	axi2_rd_timeout_en	R/W	If set to 1, enables the timeout function for the read channel of port n.	0x0
31:13	Reserved			

15.5.2.4 AXI_CTRL1_2

Table 15.9: AXI_CTRL1_2, Offset Address: 0x568

Bits	Name	Access	Description	Reset
9:0	axi2_wr_timeout_val	R/W	After an AXI write transaction is granted, a timeout counter starts to count. When it counts to axi<n>_wr_timeout_val, the corresponding channel has the highest priority.	0x0
11:10	Reserved			
12	axi2_wr_timeout_en	R/W	If set to 1, enables the timeout function for the write channel of port n.	0x0
31:13	Reserved			

15.5.2.5 AXI_CTRL0_3

Table 15.10: AXI_CTRL0_3, Offset Address: 0x614

Bits	Name	Access	Description	Reset
9:0	axi3_rd_timeout_val	R/W	After an AXI read transaction is granted, a timeout counter starts to count. When it counts to axi<n>_rd_timeout_val, the corresponding channel has the highest priority.	0x0
11:10	Reserved			
12	axi3_rd_timeout_en	R/W	If set to 1, enables the timeout function for the read channel of port n.	0x0
31:13	Reserved			

15.5.2.6 AXI_CTRL1_3

Table 15.11: AXI_CTRL1_3, Offset Address: 0x618

Bits	Name	Access	Description	Reset
9:0	axi3_wr_timeout_val	R/W	After an AXI write transaction is granted, a timeout counter starts to count. When it counts to axi<n>_wr_timeout_val, the corresponding channel has the highest priority.	0x0
11:10	Reserved			
12	axi3_wr_timeout_en	R/W	If set to 1, enables the timeout function for the write channel of port n.	0x0
31:13	Reserved			

15.5.3 AXI Register Description (Base address: 0x0800_8000)

15.5.3.1 AXI_MON0_CTRL

Table 15.12: AXI_MON0_CTRL, Offset Address: 0x000

Bits	Name	Access	Description	Reset
0	axi_mon0_en	R/W	If set to 1, enables the AXI monitor function.	0x0
1	axi_mon0_clear	R/W	Clear all the counter.	0x0
2	axi_mon0_snapshot	R/W	Snapshot all the counter.	0x0
3	axi_mon0_snapshot_all	R/W	Snapshot all the counter of all AXI monitors	0x0
4	axi_mon0_irq_en	R/W	If set to 1, enables the AXI monitor interrupt.	0x0
5	axi_mon0_irq_clear	R/W	If set to 1, clears the axi_mon_irq.	0x0
6	Reserved			
7	axi_mon0_irq	RO	Assert when all axi_mon<n>_hit_sel succes.	
31:8	Reserved			

15.5.3.2 AXI_MON0_INPUT

Table 15.13: AXI_MON0_INPUT, Offset Address: 0x004

Bits	Name	Access	Description	Reset
5:0	axi_mon0_input_sel	R/W	Input/clk selection, 0 = No selection.	0x0
31:6	Reserved			

15.5.3.3 AXI_MON0_FILTER0

Table 15.14: AXI_MON0_FILTER0, Offset Address: 0x010

Bits	Name	Access	Description	Reset
9:0	axi_mon0_hit_sel	R/W	Select which conditions are used to judege hit bit [0]: addr_st/addr_sp bit [1]: id/id_mask bit [2]: len bit [3]: size bit [4]: burst bit [5]: lock bit [6]: cache bit [7]: prot bit [8]: qos bit [9]: AXI transaction cross 4KB boundary	0x0
31:10	Reserved			

15.5.3.4 AXI_MON0_FILTER1

Table 15.15: AXI_MON0_FILTER1, Offset Address: 0x014

Bits	Name	Access	Description	Reset
31:0	axi_mon0_hit_addr_st_lo	R/W	Hit start address[31:0] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0

15.5.3.5 AXI_MON0_FILTER2

Table 15.16: AXI_MON0_FILTER2, Offset Address: 0x018

Bits	Name	Access	Description	Reset
7:0	axi_mon0_hit_addr_st_hi	R/W	Hit start address[39:32] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0
31:8	Reserved			

15.5.3.6 AXI_MON0_FILTER3

Table 15.17: AXI_MON0_FILTER3, Offset Address: 0x01c

Bits	Name	Access	Description	Reset
31:0	axi_mon0_hit_addr_sp_lo	R/W	Hit end address[31:0]	0x0

15.5.3.7 AXI_MON0_FILTER4

Table 15.18: AXI_MON0_FILTER4, Offset Address: 0x020

Bits	Name	Access	Description	Reset
7:0	axi_mon0_hit_addr_sp_hi	R/W	Hit end address[39:32]	0x0
31:8	Reserved			

15.5.3.8 AXI_MON0_FILTER5

Table 15.19: AXI_MON0_FILTER5, Offset Address: 0x024

Bits	Name	Access	Description	Reset
23:0	axi_mon0_hit_id_mask	R/W	hit = (hit_id & hit_id_mask) == (Ax_id & hit_id_mask)	0x0
31:24	Reserved			

15.5.3.9 AXI_MON0_FILTER6

Table 15.20: AXI_MON0_FILTER6, Offset Address: 0x028

Bits	Name	Access	Description	Reset
23:0	axi_mon0_hit_id	R/W	hit = Ax_id == hit_id	0x0
31:24	Reserved			

15.5.3.10 AXI_MON0_FILTER7

Table 15.21: AXI_MON0_FILTER7, Offset Address: 0x02c

Bits	Name	Access	Description	Reset
7:0	axi_mon0_hit_len	R/W	hit = Ax_len == hit_len	0x0
10:8	axi_mon0_hit_size	R/W	hit = Ax_size == hit_size	0x0
11	Reserved			
13:12	axi_mon0_hit_burst	R/W	hit = Ax_burst == hit_burst	0x0
31:14	Reserved			

15.5.3.11 AXI_MON0_FILTER8

Table 15.22: AXI_MON0_FILTER8, Offset Address: 0x030

Bits	Name	Access	Description	Reset
0	axi_mon0_hit_lock	R/W	hit = Ax_lock == hit_lock	0x0
3:1	Reserved			
7:4	axi_mon0_hit_cache	R/W	hit = Ax_cache == hit_cache	0x0
10:8	axi_mon0_hit_prot	R/W	hit = Ax_prot == hit_prot	0x0
11	Reserved			
15:12	axi_mon0_hit_qos	R/W	hit = Ax_qos == hit_qos	0x0
31:16	Reserved			

15.5.3.12 AXI_MON0_RPT0

Table 15.23: AXI_MON0_RPT0, Offset Address: 0x040

Bits	Name	Access	Description	Reset
31:0	axi_mon0_cycle_count	RO	AXI monitor 0 cycle count, counting after func_en assert	

15.5.3.13 AXI_MON0_RPT1

Table 15.24: AXI_MON0_RPT1, Offset Address: 0x044

Bits	Name	Access	Description	Reset
31:0	axi_mon0_hit_count	RO	AXI monitor 0 hit count, counting after func_en assert	

15.5.3.14 AXI_MON0_RPT2

Table 15.25: AXI_MON0_RPT2, Offset Address: 0x048

Bits	Name	Access	Description	Reset
31:0	axi_mon0_byte_count	RO	AXI monitor 0 byte count, counting after func_en assert, (Ax_len + 1) << Ax_size	

15.5.3.15 AXI_MON0_RPT3

Table 15.26: AXI_MON0_RPT3, Offset Address: 0x04c

Bits	Name	Access	Description	Reset
31:0	axi_mon0_latency_count	RO	AXI monitor 0 latency count, counting after func_en assert, += oustanding	

15.5.3.16 AXI_MON1_CTRL

Table 15.27: AXI_MON1_CTRL, Offset Address: 0x080

Bits	Name	Access	Description	Reset
0	axi_mon1_en	R/W	If set to 1, enables the AXI monitor function.	0x0
1	axi_mon1_clear	R/W	Clear all the counter.	0x0
2	axi_mon1_snapshot	R/W	Snapshot all the counter.	0x0
3	axi_mon0_snapshot_all	R/W	Snapshot all the counter of all AXI monitors	0x0
4	axi_mon1_irq_en	R/W	If set to 1, enables the AXI monitor interrupt.	0x0
5	axi_mon1_irq_clear	R/W	If set to 1, clears the axi_mon_irq.	0x0
6	Reserved			
7	axi_mon1_irq	RO	Assert when all axi_mon<n>_hit_sel succes.	
31:8	Reserved			

15.5.3.17 AXI_MON1_INPUT

Table 15.28: AXI_MON1_INPUT, Offset Address: 0x084

Bits	Name	Access	Description	Reset
5:0	axi_mon1_input_sel	R/W	Input/clk selection, 0 = No selection.	0x0
31:6	Reserved			

15.5.3.18 AXI_MON1_FILTER0

Table 15.29: AXI_MON1_FILTER0, Offset Address: 0x090

Bits	Name	Access	Description	Reset
9:0	axi_mon1_hit_sel	R/W	Select which conditions are used to judege hit bit [0]: addr_st/addr_sp bit [1]: id/id_mask bit [2]: len bit [3]: size bit [4]: burst bit [5]: lock bit [6]: cache bit [7]: prot bit [8]: qos bit [9]: AXI transaction cross 4KB boundary	0x0
31:10	Reserved			

15.5.3.19 AXI_MON1_FILTER1

Table 15.30: AXI_MON1_FILTER1, Offset Address: 0x094

Bits	Name	Access	Description	Reset
31:0	axi_mon1_hit_addr_st_lo	R/W	Hit start address[31:0] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0

15.5.3.20 AXI_MON1_FILTER2

Table 15.31: AXI_MON1_FILTER2, Offset Address: 0x098

Bits	Name	Access	Description	Reset
7:0	axi_mon1_hit_addr_st_hi	R/W	Hit start address[39:32] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0
31:8	Reserved			

15.5.3.21 AXI_MON1_FILTER3

Table 15.32: AXI_MON1_FILTER3, Offset Address: 0x09c

Bits	Name	Access	Description	Reset
31:0	axi_mon1_hit_addr_sp_lo	R/W	Hit end address[31:0]	0x0

15.5.3.22 AXI_MON1_FILTER4

Table 15.33: AXI_MON1_FILTER4, Offset Address: 0x0a0

Bits	Name	Access	Description	Reset
7:0	axi_mon1_hit_addr_sp_hi	R/W	Hit end address[39:32]	0x0
31:8	Reserved			

15.5.3.23 AXI_MON1_FILTER5

Table 15.34: AXI_MON1_FILTER5, Offset Address: 0x0a4

Bits	Name	Access	Description	Reset
23:0	axi_mon1_hit_id_mask	R/W	hit = (hit_id & hit_id_mask) == (Ax_id & hit_id_mask)	0x0
31:24	Reserved			

15.5.3.24 AXI_MON1_FILTER6

Table 15.35: AXI_MON1_FILTER6, Offset Address: 0x0a8

Bits	Name	Access	Description	Reset
23:0	axi_mon1_hit_id	R/W	hit = Ax_id == hit_id	0x0
31:24	Reserved			

15.5.3.25 AXI_MON1_FILTER7

Table 15.36: AXI_MON1_FILTER7, Offset Address: 0x0ac

Bits	Name	Access	Description	Reset
7:0	axi_mon1_hit_len	R/W	hit = Ax_len == hit_len	0x0
10:8	axi_mon1_hit_size	R/W	hit = Ax_size == hit_size	0x0
11	Reserved			
13:12	axi_mon1_hit_burst	R/W	hit = Ax_burst == hit_burst	0x0
31:14	Reserved			

15.5.3.26 AXI_MON1_FILTER8

Table 15.37: AXI_MON1_FILTER8, Offset Address: 0x0b0

Bits	Name	Access	Description	Reset
0	axi_mon1_hit_lock	R/W	hit = Ax_lock == hit_lock	0x0
3:1	Reserved			
7:4	axi_mon1_hit_cache	R/W	hit = Ax_cache == hit_cache	0x0
10:8	axi_mon1_hit_prot	R/W	hit = Ax_prot == hit_prot	0x0
11	Reserved			
15:12	axi_mon1_hit_qos	R/W	hit = Ax_qos == hit_qos	0x0
31:16	Reserved			

15.5.3.27 AXI_MON1_RPT0

Table 15.38: AXI_MON1_RPT0, Offset Address: 0x0c0

Bits	Name	Access	Description	Reset
31:0	axi_mon1_cycle_count	RO	AXI monitor 1 cycle count, counting after func_en assert	

15.5.3.28 AXI_MON1_RPT1

Table 15.39: AXI_MON1_RPT1, Offset Address: 0x0c4

Bits	Name	Access	Description	Reset
31:0	axi_mon1_hit_count	RO	AXI monitor 1 hit count, counting after func_en assert	

15.5.3.29 AXI_MON1_RPT2

Table 15.40: AXI_MON1_RPT2, Offset Address: 0x0c8

Bits	Name	Access	Description	Reset
31:0	axi_mon1_byte_count	RO	AXI monitor 1 byte count, counting after func_en assert, (Ax_len + 1) << Ax_size	

15.5.3.30 AXI_MON1_RPT3

Table 15.41: AXI_MON1_RPT3, Offset Address: 0x0cc

Bits	Name	Access	Description	Reset
31:0	axi_mon1_latency_count	RO	AXI monitor 1 latency count, counting after func_en assert, += oustanding	

15.5.3.31 AXI_MON2_CTRL

Table 15.42: AXI_MON2_CTRL, Offset Address: 0x100

Bits	Name	Access	Description	Reset
0	axi_mon2_en	R/W	If set to 1, enables the AXI monitor function.	0x0
1	axi_mon2_clear	R/W	Clear all the counter.	0x0
2	axi_mon2_snapshot	R/W	Snapshot all the counter.	0x0
3	axi_mon0_snapshot_all	R/W	Snapshot all the counter of all AXI monitors	0x0
4	axi_mon2_irq_en	R/W	If set to 1, enables the AXI monitor interrupt.	0x0
5	axi_mon2_irq_clear	R/W	If set to 1, clears the axi_mon_irq.	0x0
6	Reserved			
7	axi_mon2_irq	RO	Assert when all axi_mon<n>_hit_sel succes.	
31:8	Reserved			

15.5.3.32 AXI_MON2_INPUT

Table 15.43: AXI_MON2_INPUT, Offset Address: 0x104

Bits	Name	Access	Description	Reset
5:0	axi_mon2_input_sel	R/W	Input/clk selection, 0 = No selection.	0x0
31:6	Reserved			

15.5.3.33 AXI_MON2_FILTER0

Table 15.44: AXI_MON2_FILTER0, Offset Address: 0x110

Bits	Name	Access	Description	Reset
9:0	axi_mon2_hit_sel	R/W	Select which conditions are used to judege hit bit [0]: addr_st/addr_sp bit [1]: id/id_mask bit [2]: len bit [3]: size bit [4]: burst bit [5]: lock bit [6]: cache bit [7]: prot bit [8]: qos bit [9]: AXI transaction cross 4KB boundary	0x0
31:10	Reserved			

15.5.3.34 AXI_MON2_FILTER1

Table 15.45: AXI_MON2_FILTER1, Offset Address: 0x114

Bits	Name	Access	Description	Reset
31:0	axi_mon2_hit_addr_st_lo	R/W	Hit start address[31:0] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0

15.5.3.35 AXI_MON2_FILTER2

Table 15.46: AXI_MON2_FILTER2, Offset Address: 0x118

Bits	Name	Access	Description	Reset
7:0	axi_mon2_hit_addr_st_hi	R/W	Hit start address[39:32] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0
31:8	Reserved			

15.5.3.36 AXI_MON2_FILTER3

Table 15.47: AXI_MON2_FILTER3, Offset Address: 0x11c

Bits	Name	Access	Description	Reset
31:0	axi_mon2_hit_addr_sp_lo	R/W	Hit end address[31:0]	0x0

15.5.3.37 AXI_MON2_FILTER4

Table 15.48: AXI_MON2_FILTER4, Offset Address: 0x120

Bits	Name	Access	Description	Reset
7:0	axi_mon2_hit_addr_sp_hi	R/W	Hit end address[39:32]	0x0
31:8	Reserved			

15.5.3.38 AXI_MON2_FILTER5

Table 15.49: AXI_MON2_FILTER5, Offset Address: 0x124

Bits	Name	Access	Description	Reset
23:0	axi_mon2_hit_id_mask	R/W	hit = (hit_id & hit_id_mask) == (Ax_id & hit_id_mask)	0x0
31:24	Reserved			

15.5.3.39 AXI_MON2_FILTER6

Table 15.50: AXI_MON2_FILTER6, Offset Address: 0x128

Bits	Name	Access	Description	Reset
23:0	axi_mon2_hit_id	R/W	hit = Ax_id == hit_id	0x0
31:24	Reserved			

15.5.3.40 AXI_MON2_FILTER7

Table 15.51: AXI_MON2_FILTER7, Offset Address: 0x12c

Bits	Name	Access	Description	Reset
7:0	axi_mon2_hit_len	R/W	hit = Ax_len == hit_len	0x0
10:8	axi_mon2_hit_size	R/W	hit = Ax_size == hit_size	0x0
11	Reserved			
13:12	axi_mon2_hit_burst	R/W	hit = Ax_burst == hit_burst	0x0
31:14	Reserved			

15.5.3.41 AXI_MON2_FILTER8

Table 15.52: AXI_MON2_FILTER8, Offset Address: 0x130

Bits	Name	Access	Description	Reset
0	axi_mon2_hit_lock	R/W	hit = Ax_lock == hit_lock	0x0
3:1	Reserved			
7:4	axi_mon2_hit_cache	R/W	hit = Ax_cache == hit_cache	0x0
10:8	axi_mon2_hit_prot	R/W	hit = Ax_prot == hit_prot	0x0
11	Reserved			
15:12	axi_mon2_hit_qos	R/W	hit = Ax_qos == hit_qos	0x0
31:16	Reserved			

15.5.3.42 AXI_MON2_RPT0

Table 15.53: AXI_MON2_RPT0, Offset Address: 0x140

Bits	Name	Access	Description	Reset
31:0	axi_mon2_cycle_count	RO	AXI monitor 2 cycle count, counting after func_en assert	

15.5.3.43 AXI_MON2_RPT1

Table 15.54: AXI_MON2_RPT1, Offset Address: 0x144

Bits	Name	Access	Description	Reset
31:0	axi_mon2_hit_count	RO	AXI monitor 2 hit count, counting after func_en assert	

15.5.3.44 AXI_MON2_RPT2

Table 15.55: AXI_MON2_RPT2, Offset Address: 0x148

Bits	Name	Access	Description	Reset
31:0	axi_mon2_byte_count	RO	AXI monitor 2 byte count, counting after func_en assert, (Ax_len + 1) << Ax_size	

15.5.3.45 AXI_MON2_RPT3

Table 15.56: AXI_MON2_RPT3, Offset Address: 0x14c

Bits	Name	Access	Description	Reset
31:0	axi_mon2_latency_count	RO	AXI monitor 2 latency count, counting after func_en assert, += oustanding	

15.5.3.46 AXI_MON3_CTRL

Table 15.57: AXI_MON3_CTRL, Offset Address: 0x180

Bits	Name	Access	Description	Reset
0	axi_mon3_en	R/W	If set to 1, enables the AXI monitor function.	0x0
1	axi_mon3_clear	R/W	Clear all the counter.	0x0
2	axi_mon3_snapshot	R/W	Snapshot all the counter.	0x0
3	axi_mon0_snapshot_all	R/W	Snapshot all the counter of all AXI monitors	0x0
4	axi_mon3_irq_en	R/W	If set to 1, enables the AXI monitor interrupt.	0x0
5	axi_mon3_irq_clear	R/W	If set to 1, clears the axi_mon_irq.	0x0
6	Reserved			
7	axi_mon3_irq	RO	Assert when all axi_mon<n>_hit_sel succes.	
31:8	Reserved			

15.5.3.47 AXI_MON3_INPUT

Table 15.58: AXI_MON3_INPUT, Offset Address: 0x184

Bits	Name	Access	Description	Reset
5:0	axi_mon3_input_sel	R/W	Input/clk selection, 0 = No selection.	0x0
31:6	Reserved			

15.5.3.48 AXI_MON3_FILTER0

Table 15.59: AXI_MON3_FILTER0, Offset Address: 0x190

Bits	Name	Access	Description	Reset
9:0	axi_mon3_hit_sel	R/W	Select which conditions are used to judege hit bit [0]: addr_st/addr_sp bit [1]: id/id_mask bit [2]: len bit [3]: size bit [4]: burst bit [5]: lock bit [6]: cache bit [7]: prot bit [8]: qos bit [9]: AXI transaction cross 4KB boundary	0x0
31:10	Reserved			

15.5.3.49 AXI_MON3_FILTER1

Table 15.60: AXI_MON3_FILTER1, Offset Address: 0x194

Bits	Name	Access	Description	Reset
31:0	axi_mon3_hit_addr_st_lo	R/W	Hit start address[31:0] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0

15.5.3.50 AXI_MON3_FILTER2

Table 15.61: AXI_MON3_FILTER2, Offset Address: 0x198

Bits	Name	Access	Description	Reset
7:0	axi_mon3_hit_addr_st_hi	R/W	Hit start address[39:32] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0
31:8	Reserved			

15.5.3.51 AXI_MON3_FILTER3

Table 15.62: AXI_MON3_FILTER3, Offset Address: 0x19c

Bits	Name	Access	Description	Reset
31:0	axi_mon3_hit_addr_sp_lo	R/W	Hit end address[31:0]	0x0

15.5.3.52 AXI_MON3_FILTER4

Table 15.63: AXI_MON3_FILTER4, Offset Address: 0x1a0

Bits	Name	Access	Description	Reset
7:0	axi_mon3_hit_addr_sp_hi	R/W	Hit end address[39:32]	0x0
31:8	Reserved			

15.5.3.53 AXI_MON3_FILTER5

Table 15.64: AXI_MON3_FILTER5, Offset Address: 0x1a4

Bits	Name	Access	Description	Reset
23:0	axi_mon3_hit_id_mask	R/W	hit = (hit_id & hit_id_mask) == (Ax_id & hit_id_mask)	0x0
31:24	Reserved			

15.5.3.54 AXI_MON3_FILTER6

Table 15.65: AXI_MON3_FILTER6, Offset Address: 0x1a8

Bits	Name	Access	Description	Reset
23:0	axi_mon3_hit_id	R/W	hit = Ax_id == hit_id	0x0
31:24	Reserved			

15.5.3.55 AXI_MON3_FILTER7

Table 15.66: AXI_MON3_FILTER7, Offset Address: 0x1ac

Bits	Name	Access	Description	Reset
7:0	axi_mon3_hit_len	R/W	hit = Ax_len == hit_len	0x0
10:8	axi_mon3_hit_size	R/W	hit = Ax_size == hit_size	0x0
11	Reserved			
13:12	axi_mon3_hit_burst	R/W	hit = Ax_burst == hit_burst	0x0
31:14	Reserved			

15.5.3.56 AXI_MON3_FILTER8

Table 15.67: AXI_MON3_FILTER8, Offset Address: 0x1b0

Bits	Name	Access	Description	Reset
0	axi_mon3_hit_lock	R/W	hit = Ax_lock == hit_lock	0x0
3:1	Reserved			
7:4	axi_mon3_hit_cache	R/W	hit = Ax_cache == hit_cache	0x0
10:8	axi_mon3_hit_prot	R/W	hit = Ax_prot == hit_prot	0x0
11	Reserved			
15:12	axi_mon3_hit_qos	R/W	hit = Ax_qos == hit_qos	0x0
31:16	Reserved			

15.5.3.57 AXI_MON3_RPT0

Table 15.68: AXI_MON3_RPT0, Offset Address: 0x1c0

Bits	Name	Access	Description	Reset
31:0	axi_mon3_cycle_count	RO	AXI monitor 3 cycle count, counting after func_en assert	

15.5.3.58 AXI_MON3_RPT1

Table 15.69: AXI_MON3_RPT1, Offset Address: 0x1c4

Bits	Name	Access	Description	Reset
31:0	axi_mon3_hit_count	RO	AXI monitor 3 hit count, counting after func_en assert	

15.5.3.59 AXI_MON3_RPT2

Table 15.70: AXI_MON3_RPT2, Offset Address: 0x1c8

Bits	Name	Access	Description	Reset
31:0	axi_mon3_byte_count	RO	AXI monitor 3 byte count, counting after func_en assert, (Ax_len + 1) << Ax_size	

15.5.3.60 AXI_MON3_RPT3

Table 15.71: AXI_MON3_RPT3, Offset Address: 0x1cc

Bits	Name	Access	Description	Reset
31:0	axi_mon3_latency_count	RO	AXI monitor 3 latency count, counting after func_en assert, += oustanding	

15.5.3.61 AXI_MON4_CTRL

Table 15.72: AXI_MON4_CTRL, Offset Address: 0x200

Bits	Name	Access	Description	Reset
0	axi_mon4_en	R/W	If set to 1, enables the AXI monitor function.	0x0
1	axi_mon4_clear	R/W	Clear all the counter.	0x0
2	axi_mon4_snapshot	R/W	Snapshot all the counter.	0x0
3	axi_mon0_snapshot_all	R/W	Snapshot all the counter of all AXI monitors	0x0
4	axi_mon4_irq_en	R/W	If set to 1, enables the AXI monitor interrupt.	0x0
5	axi_mon4_irq_clear	R/W	If set to 1, clears the axi_mon_irq.	0x0
6	Reserved			
7	axi_mon4_irq	RO	Assert when all axi_mon<n>_hit_sel succes.	
31:8	Reserved			

15.5.3.62 AXI_MON4_INPUT

Table 15.73: AXI_MON4_INPUT, Offset Address: 0x204

Bits	Name	Access	Description	Reset
5:0	axi_mon4_input_sel	R/W	Input/clk selection, 0 = No selection.	0x0
31:6	Reserved			

15.5.3.63 AXI_MON4_FILTER0

Table 15.74: AXI_MON4_FILTER0, Offset Address: 0x210

Bits	Name	Access	Description	Reset
9:0	axi_mon4_hit_sel	R/W	Select which conditions are used to judege hit bit [0]: addr_st/addr_sp bit [1]: id/id_mask bit [2]: len bit [3]: size bit [4]: burst bit [5]: lock bit [6]: cache bit [7]: prot bit [8]: qos bit [9]: AXI transaction cross 4KB boundary	0x0
31:10	Reserved			

15.5.3.64 AXI_MON4_FILTER1

Table 15.75: AXI_MON4_FILTER1, Offset Address: 0x214

Bits	Name	Access	Description	Reset
31:0	axi_mon4_hit_addr_st_lo	R/W	Hit start address[31:0] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0

15.5.3.65 AXI_MON4_FILTER2

Table 15.76: AXI_MON4_FILTER2, Offset Address: 0x218

Bits	Name	Access	Description	Reset
7:0	axi_mon4_hit_addr_st_hi	R/W	Hit start address[39:32] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0
31:8	Reserved			

15.5.3.66 AXI_MON4_FILTER3

Table 15.77: AXI_MON4_FILTER3, Offset Address: 0x21c

Bits	Name	Access	Description	Reset
31:0	axi_mon4_hit_addr_sp_lo	R/W	Hit end address[31:0]	0x0

15.5.3.67 AXI_MON4_FILTER4

Table 15.78: AXI_MON4_FILTER4, Offset Address: 0x220

Bits	Name	Access	Description	Reset
7:0	axi_mon4_hit_addr_sp_hi	R/W	Hit end address[39:32]	0x0
31:8	Reserved			

15.5.3.68 AXI_MON4_FILTER5

Table 15.79: AXI_MON4_FILTER5, Offset Address: 0x224

Bits	Name	Access	Description	Reset
23:0	axi_mon4_hit_id_mask	R/W	hit = (hit_id & hit_id_mask) == (Ax_id & hit_id_mask)	0x0
31:24	Reserved			

15.5.3.69 AXI_MON4_FILTER6

Table 15.80: AXI_MON4_FILTER6, Offset Address: 0x228

Bits	Name	Access	Description	Reset
23:0	axi_mon4_hit_id	R/W	hit = Ax_id == hit_id	0x0
31:24	Reserved			

15.5.3.70 AXI_MON4_FILTER7

Table 15.81: AXI_MON4_FILTER7, Offset Address: 0x22c

Bits	Name	Access	Description	Reset
7:0	axi_mon4_hit_len	R/W	hit = Ax_len == hit_len	0x0
10:8	axi_mon4_hit_size	R/W	hit = Ax_size == hit_size	0x0
11	Reserved			
13:12	axi_mon4_hit_burst	R/W	hit = Ax_burst == hit_burst	0x0
31:14	Reserved			

15.5.3.71 AXI_MON4_FILTER8

Table 15.82: AXI_MON4_FILTER8, Offset Address: 0x230

Bits	Name	Access	Description	Reset
0	axi_mon4_hit_lock	R/W	hit = Ax_lock == hit_lock	0x0
3:1	Reserved			
7:4	axi_mon4_hit_cache	R/W	hit = Ax_cache == hit_cache	0x0
10:8	axi_mon4_hit_prot	R/W	hit = Ax_prot == hit_prot	0x0
11	Reserved			
15:12	axi_mon4_hit_qos	R/W	hit = Ax_qos == hit_qos	0x0
31:16	Reserved			

15.5.3.72 AXI_MON4_RPT0

Table 15.83: AXI_MON4_RPT0, Offset Address: 0x240

Bits	Name	Access	Description	Reset
31:0	axi_mon4_cycle_count	RO	AXI monitor 4 cycle count, counting after func_en assert	

15.5.3.73 AXI_MON4_RPT1

Table 15.84: AXI_MON4_RPT1, Offset Address: 0x244

Bits	Name	Access	Description	Reset
31:0	axi_mon4_hit_count	RO	AXI monitor 4 hit count, counting after func_en assert	

15.5.3.74 AXI_MON4_RPT2

Table 15.85: AXI_MON4_RPT2, Offset Address: 0x248

Bits	Name	Access	Description	Reset
31:0	axi_mon4_byte_count	RO	AXI monitor 4 byte count, counting after func_en assert, (Ax_len + 1) << Ax_size	

15.5.3.75 AXI_MON4_RPT3

Table 15.86: AXI_MON4_RPT3, Offset Address: 0x24c

Bits	Name	Access	Description	Reset
31:0	axi_mon4_latency_count	RO	AXI monitor 4 latency count, counting after func_en assert, += oustanding	

15.5.3.76 AXI_MON5_CTRL

Table 15.87: AXI_MON5_CTRL, Offset Address: 0x280

Bits	Name	Access	Description	Reset
0	axi_mon5_en	R/W	If set to 1, enables the AXI monitor function.	0x0
1	axi_mon5_clear	R/W	Clear all the counter.	0x0
2	axi_mon5_snapshot	R/W	Snapshot all the counter.	0x0
3	axi_mon0_snapshot_all	R/W	Snapshot all the counter of all AXI monitors	0x0
4	axi_mon5_irq_en	R/W	If set to 1, enables the AXI monitor interrupt.	0x0
5	axi_mon5_irq_clear	R/W	If set to 1, clears the axi_mon_irq.	0x0
6	Reserved			
7	axi_mon5_irq	RO	Assert when all axi_mon<n>_hit_sel suscces.	
31:8	Reserved			

15.5.3.77 AXI_MON5_INPUT

Table 15.88: AXI_MON5_INPUT, Offset Address: 0x284

Bits	Name	Access	Description	Reset
5:0	axi_mon5_input_sel	R/W	Input/clk selection, 0 = No selection.	0x0
31:6	Reserved			

15.5.3.78 AXI_MON5_FILTER0

Table 15.89: AXI_MON5_FILTER0, Offset Address: 0x290

Bits	Name	Access	Description	Reset
9:0	axi_mon5_hit_sel	R/W	Select which conditions are used to judege hit bit [0]: addr_st/addr_sp bit [1]: id/id_mask bit [2]: len bit [3]: size bit [4]: burst bit [5]: lock bit [6]: cache bit [7]: prot bit [8]: qos bit [9]: AXI transaction cross 4KB boundary	0x0
31:10	Reserved			

15.5.3.79 AXI_MON5_FILTER1

Table 15.90: AXI_MON5_FILTER1, Offset Address: 0x294

Bits	Name	Access	Description	Reset
31:0	axi_mon5_hit_addr_st_lo	R/W	Hit start address[31:0] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0

15.5.3.80 AXI_MON5_FILTER2

Table 15.91: AXI_MON5_FILTER2, Offset Address: 0x298

Bits	Name	Access	Description	Reset
7:0	axi_mon5_hit_addr_st_hi	R/W	Hit start address[39:32] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0
31:8	Reserved			

15.5.3.81 AXI_MON5_FILTER3

Table 15.92: AXI_MON5_FILTER3, Offset Address: 0x29c

Bits	Name	Access	Description	Reset
31:0	axi_mon5_hit_addr_sp_lo	R/W	Hit end address[31:0]	0x0

15.5.3.82 AXI_MON5_FILTER4

Table 15.93: AXI_MON5_FILTER4, Offset Address: 0x2a0

Bits	Name	Access	Description	Reset
7:0	axi_mon5_hit_addr_sp_hi	R/W	Hit end address[39:32]	0x0
31:8	Reserved			

15.5.3.83 AXI_MON5_FILTER5

Table 15.94: AXI_MON5_FILTER5, Offset Address: 0x2a4

Bits	Name	Access	Description	Reset
23:0	axi_mon5_hit_id_mask	R/W	hit = (hit_id & hit_id_mask) == (Ax_id & hit_id_mask)	0x0
31:24	Reserved			

15.5.3.84 AXI_MON5_FILTER6

Table 15.95: AXI_MON5_FILTER6, Offset Address: 0x2a8

Bits	Name	Access	Description	Reset
23:0	axi_mon5_hit_id	R/W	hit = Ax_id == hit_id	0x0
31:24	Reserved			

15.5.3.85 AXI_MON5_FILTER7

Table 15.96: AXI_MON5_FILTER7, Offset Address: 0x2ac

Bits	Name	Access	Description	Reset
7:0	axi_mon5_hit_len	R/W	hit = Ax_len == hit_len	0x0
10:8	axi_mon5_hit_size	R/W	hit = Ax_size == hit_size	0x0
11	Reserved			
13:12	axi_mon5_hit_burst	R/W	hit = Ax_burst == hit_burst	0x0
31:14	Reserved			

15.5.3.86 AXI_MON5_FILTER8

Table 15.97: AXI_MON5_FILTER8, Offset Address: 0x2b0

Bits	Name	Access	Description	Reset
0	axi_mon5_hit_lock	R/W	hit = Ax_lock == hit_lock	0x0
3:1	Reserved			
7:4	axi_mon5_hit_cache	R/W	hit = Ax_cache == hit_cache	0x0
10:8	axi_mon5_hit_prot	R/W	hit = Ax_prot == hit_prot	0x0
11	Reserved			
15:12	axi_mon5_hit_qos	R/W	hit = Ax_qos == hit_qos	0x0
31:16	Reserved			

15.5.3.87 AXI_MON5_RPT0

Table 15.98: AXI_MON5_RPT0, Offset Address: 0x2c0

Bits	Name	Access	Description	Reset
31:0	axi_mon5_cycle_count	RO	AXI monitor 5 cycle count, counting after func_en assert	

15.5.3.88 AXI_MON5_RPT1

Table 15.99: AXI_MON5_RPT1, Offset Address: 0x2c4

Bits	Name	Access	Description	Reset
31:0	axi_mon5_hit_count	RO	AXI monitor 5 hit count, counting after func_en assert	

15.5.3.89 AXI_MON5_RPT2

Table 15.100: AXI_MON5_RPT2, Offset Address: 0x2c8

Bits	Name	Access	Description	Reset
31:0	axi_mon5_byte_count	RO	AXI monitor 5 byte count, counting after func_en assert, (Ax_len + 1) << Ax_size	

15.5.3.90 AXI_MON5_RPT3

Table 15.101: AXI_MON5_RPT3, Offset Address: 0x2cc

Bits	Name	Access	Description	Reset
31:0	axi_mon5_latency_count	RO	AXI monitor 5 latency count, counting after func_en assert, += oustanding	

15.5.3.91 AXI_MON6_CTRL

Table 15.102: AXI_MON6_CTRL, Offset Address: 0x300

Bits	Name	Access	Description	Reset
0	axi_mon6_en	R/W	If set to 1, enables the AXI monitor function.	0x0
1	axi_mon6_clear	R/W	Clear all the counter.	0x0
2	axi_mon6_snapshot	R/W	Snapshot all the counter.	0x0
3	axi_mon0_snapshot_all	R/W	Snapshot all the counter of all AXI monitors	0x0
4	axi_mon6_irq_en	R/W	If set to 1, enables the AXI monitor interrupt.	0x0
5	axi_mon6_irq_clear	R/W	If set to 1, clears the axi_mon_irq.	0x0
6	Reserved			
7	axi_mon6_irq	RO	Assert when all axi_mon<n>_hit_sel succes.	
31:8	Reserved			

15.5.3.92 AXI_MON6_INPUT

Table 15.103: AXI_MON6_INPUT, Offset Address: 0x304

Bits	Name	Access	Description	Reset
5:0	axi_mon6_input_sel	R/W	Input/clk selection, 0 = No selection.	0x0
31:6	Reserved			

15.5.3.93 AXI_MON6_FILTER0

Table 15.104: AXI_MON6_FILTER0, Offset Address: 0x310

Bits	Name	Access	Description	Reset
9:0	axi_mon6_hit_sel	R/W	Select which conditions are used to judege hit bit [0]: addr_st/addr_sp bit [1]: id/id_mask bit [2]: len bit [3]: size bit [4]: burst bit [5]: lock bit [6]: cache bit [7]: prot bit [8]: qos bit [9]: AXI transaction cross 4KB boundary	0x0
31:10	Reserved			

15.5.3.94 AXI_MON6_FILTER1

Table 15.105: AXI_MON6_FILTER1, Offset Address: 0x314

Bits	Name	Access	Description	Reset
31:0	axi_mon6_hit_addr_st_lo	R/W	Hit start address[31:0] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0

15.5.3.95 AXI_MON6_FILTER2

Table 15.106: AXI_MON6_FILTER2, Offset Address: 0x318

Bits	Name	Access	Description	Reset
7:0	axi_mon6_hit_addr_st_hi	R/W	Hit start address[39:32] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0
31:8	Reserved			

15.5.3.96 AXI_MON6_FILTER3

Table 15.107: AXI_MON6_FILTER3, Offset Address: 0x31c

Bits	Name	Access	Description	Reset
31:0	axi_mon6_hit_addr_sp_lo	R/W	Hit end address[31:0]	0x0

15.5.3.97 AXI_MON6_FILTER4

Table 15.108: AXI_MON6_FILTER4, Offset Address: 0x320

Bits	Name	Access	Description	Reset
7:0	axi_mon6_hit_addr_sp_hi	R/W	Hit end address[39:32]	0x0
31:8	Reserved			

15.5.3.98 AXI_MON6_FILTER5

Table 15.109: AXI_MON6_FILTER5, Offset Address: 0x324

Bits	Name	Access	Description	Reset
23:0	axi_mon6_hit_id_mask	R/W	hit = (hit_id & hit_id_mask) == (Ax_id & hit_id_mask)	0x0
31:24	Reserved			

15.5.3.99 AXI_MON6_FILTER6

Table 15.110: AXI_MON6_FILTER6, Offset Address: 0x328

Bits	Name	Access	Description	Reset
23:0	axi_mon6_hit_id	R/W	hit = Ax_id == hit_id	0x0
31:24	Reserved			

15.5.3.100 AXI_MON6_FILTER7

Table 15.111: AXI_MON6_FILTER7, Offset Address: 0x32c

Bits	Name	Access	Description	Reset
7:0	axi_mon6_hit_len	R/W	hit = Ax_len == hit_len	0x0
10:8	axi_mon6_hit_size	R/W	hit = Ax_size == hit_size	0x0
11	Reserved			
13:12	axi_mon6_hit_burst	R/W	hit = Ax_burst == hit_burst	0x0
31:14	Reserved			

15.5.3.101 AXI_MON6_FILTER8

Table 15.112: AXI_MON6_FILTER8, Offset Address: 0x330

Bits	Name	Access	Description	Reset
0	axi_mon6_hit_lock	R/W	hit = Ax_lock == hit_lock	0x0
3:1	Reserved			
7:4	axi_mon6_hit_cache	R/W	hit = Ax_cache == hit_cache	0x0
10:8	axi_mon6_hit_prot	R/W	hit = Ax_prot == hit_prot	0x0
11	Reserved			
15:12	axi_mon6_hit_qos	R/W	hit = Ax_qos == hit_qos	0x0
31:16	Reserved			

15.5.3.102 AXI_MON6_RPT0

Table 15.113: AXI_MON6_RPT0, Offset Address: 0x340

Bits	Name	Access	Description	Reset
31:0	axi_mon6_cycle_count	RO	AXI monitor 6 cycle count, counting after func_en assert	

15.5.3.103 AXI_MON6_RPT1

Table 15.114: AXI_MON6_RPT1, Offset Address: 0x344

Bits	Name	Access	Description	Reset
31:0	axi_mon6_hit_count	RO	AXI monitor 6 hit count, counting after func_en assert	

15.5.3.104 AXI_MON6_RPT2

Table 15.115: AXI_MON6_RPT2, Offset Address: 0x348

Bits	Name	Access	Description	Reset
31:0	axi_mon6_byte_count	RO	AXI monitor 6 byte count, counting after func_en assert, (Ax_len + 1) << Ax_size	

15.5.3.105 AXI_MON6_RPT3

Table 15.116: AXI_MON6_RPT3, Offset Address: 0x34c

Bits	Name	Access	Description	Reset
31:0	axi_mon6_latency_count	RO	AXI monitor 6 latency count, counting after func_en assert, += oustanding	

15.5.3.106 AXI_MON7_CTRL

Table 15.117: AXI_MON7_CTRL, Offset Address: 0x380

Bits	Name	Access	Description	Reset
0	axi_mon7_en	R/W	If set to 1, enables the AXI monitor function.	0x0
1	axi_mon7_clear	R/W	Clear all the counter.	0x0
2	axi_mon7_snapshot	R/W	Snapshot all the counter.	0x0
3	axi_mon0_snapshot_all	R/W	Snapshot all the counter of all AXI monitors	0x0
4	axi_mon7_irq_en	R/W	If set to 1, enables the AXI monitor interrupt.	0x0
5	axi_mon7_irq_clear	R/W	If set to 1, clears the axi_mon_irq.	0x0
6	Reserved			
7	axi_mon7_irq	RO	Assert when all axi_mon<n>_hit_sel succes.	
31:8	Reserved			

15.5.3.107 AXI_MON7_INPUT

Table 15.118: AXI_MON7_INPUT, Offset Address: 0x384

Bits	Name	Access	Description	Reset
5:0	axi_mon7_input_sel	R/W	Input/clk selection, 0 = No selection.	0x0
31:6	Reserved			

15.5.3.108 AXI_MON7_FILTER0

Table 15.119: AXI_MON7_FILTER0, Offset Address: 0x390

Bits	Name	Access	Description	Reset
9:0	axi_mon7_hit_sel	R/W	Select which conditions are used to judege hit bit [0]: addr_st/addr_sp bit [1]: id/id_mask bit [2]: len bit [3]: size bit [4]: burst bit [5]: lock bit [6]: cache bit [7]: prot bit [8]: qos bit [9]: AXI transaction cross 4KB boundary	0x0
31:10	Reserved			

15.5.3.109 AXI_MON7_FILTER1

Table 15.120: AXI_MON7_FILTER1, Offset Address: 0x394

Bits	Name	Access	Description	Reset
31:0	axi_mon7_hit_addr_st_lo	R/W	Hit start address[31:0] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0

15.5.3.110 AXI_MON7_FILTER2

Table 15.121: AXI_MON7_FILTER2, Offset Address: 0x398

Bits	Name	Access	Description	Reset
7:0	axi_mon7_hit_addr_st_hi	R/W	Hit start address[39:32] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0
31:8	Reserved			

15.5.3.111 AXI_MON7_FILTER3

Table 15.122: AXI_MON7_FILTER3, Offset Address: 0x39c

Bits	Name	Access	Description	Reset
31:0	axi_mon7_hit_addr_sp_lo	R/W	Hit end address[31:0]	0x0

15.5.3.112 AXI_MON7_FILTER4

Table 15.123: AXI_MON7_FILTER4, Offset Address: 0x3a0

Bits	Name	Access	Description	Reset
7:0	axi_mon7_hit_addr_sp_hi	R/W	Hit end address[39:32]	0x0
31:8	Reserved			

15.5.3.113 AXI_MON7_FILTER5

Table 15.124: AXI_MON7_FILTER5, Offset Address: 0x3a4

Bits	Name	Access	Description	Reset
23:0	axi_mon7_hit_id_mask	R/W	hit = (hit_id & hit_id_mask) == (Ax_id & hit_id_mask)	0x0
31:24	Reserved			

15.5.3.114 AXI_MON7_FILTER6

Table 15.125: AXI_MON7_FILTER6, Offset Address: 0x3a8

Bits	Name	Access	Description	Reset
23:0	axi_mon7_hit_id	R/W	hit = Ax_id == hit_id	0x0
31:24	Reserved			

15.5.3.115 AXI_MON7_FILTER7

Table 15.126: AXI_MON7_FILTER7, Offset Address: 0x3ac

Bits	Name	Access	Description	Reset
7:0	axi_mon7_hit_len	R/W	hit = Ax_len == hit_len	0x0
10:8	axi_mon7_hit_size	R/W	hit = Ax_size == hit_size	0x0
11	Reserved			
13:12	axi_mon7_hit_burst	R/W	hit = Ax_burst == hit_burst	0x0
31:14	Reserved			

15.5.3.116 AXI_MON7_FILTER8

Table 15.127: AXI_MON7_FILTER8, Offset Address: 0x3b0

Bits	Name	Access	Description	Reset
0	axi_mon7_hit_lock	R/W	hit = Ax_lock == hit_lock	0x0
3:1	Reserved			
7:4	axi_mon7_hit_cache	R/W	hit = Ax_cache == hit_cache	0x0
10:8	axi_mon7_hit_prot	R/W	hit = Ax_prot == hit_prot	0x0
11	Reserved			
15:12	axi_mon7_hit_qos	R/W	hit = Ax_qos == hit_qos	0x0
31:16	Reserved			

15.5.3.117 AXI_MON7_RPT0

Table 15.128: AXI_MON7_RPT0, Offset Address: 0x3c0

Bits	Name	Access	Description	Reset
31:0	axi_mon7_cycle_count	RO	AXI monitor 7 cycle count, counting after func_en assert	

15.5.3.118 AXI_MON7_RPT1

Table 15.129: AXI_MON7_RPT1, Offset Address: 0x3c4

Bits	Name	Access	Description	Reset
31:0	axi_mon7_hit_count	RO	AXI monitor 7 hit count, counting after func_en assert	

15.5.3.119 AXI_MON7_RPT2

Table 15.130: AXI_MON7_RPT2, Offset Address: 0x3c8

Bits	Name	Access	Description	Reset
31:0	axi_mon7_byte_count	RO	AXI monitor 7 byte count, counting after func_en assert, (Ax_len + 1) << Ax_size	

15.5.3.120 AXI_MON7_RPT3

Table 15.131: AXI_MON7_RPT3, Offset Address: 0x3cc

Bits	Name	Access	Description	Reset
31:0	axi_mon7_latency_count	RO	AXI monitor 7 latency count, counting after func_en assert, += oustanding	

15.5.3.121 AXI_MON8_CTRL

Table 15.132: AXI_MON8_CTRL, Offset Address: 0x400

Bits	Name	Access	Description	Reset
0	axi_mon8_en	R/W	If set to 1, enables the AXI monitor function.	0x0
1	axi_mon8_clear	R/W	Clear all the counter.	0x0
2	axi_mon8_snapshot	R/W	Snapshot all the counter.	0x0
3	axi_mon0_snapshot_all	R/W	Snapshot all the counter of all AXI monitors	0x0
4	axi_mon8_irq_en	R/W	If set to 1, enables the AXI monitor interrupt.	0x0
5	axi_mon8_irq_clear	R/W	If set to 1, clears the axi_mon_irq.	0x0
6	Reserved			
7	axi_mon8_irq	RO	Assert when all axi_mon<n>_hit_sel succes.	
31:8	Reserved			

15.5.3.122 AXI_MON8_INPUT

Table 15.133: AXI_MON8_INPUT, Offset Address: 0x404

Bits	Name	Access	Description	Reset
5:0	axi_mon8_input_sel	R/W	Input/clk selection, 0 = No selection.	0x0
31:6	Reserved			

15.5.3.123 AXI_MON8_FILTER0

Table 15.134: AXI_MON8_FILTER0, Offset Address: 0x410

Bits	Name	Access	Description	Reset
9:0	axi_mon8_hit_sel	R/W	Select which conditions are used to judege hit bit [0]: addr_st/addr_sp bit [1]: id/id_mask bit [2]: len bit [3]: size bit [4]: burst bit [5]: lock bit [6]: cache bit [7]: prot bit [8]: qos bit [9]: AXI transaction cross 4KB boundary	0x0
31:10	Reserved			

15.5.3.124 AXI_MON8_FILTER1

Table 15.135: AXI_MON8_FILTER1, Offset Address: 0x414

Bits	Name	Access	Description	Reset
31:0	axi_mon8_hit_addr_st_lo	R/W	Hit start address[31:0] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0

15.5.3.125 AXI_MON8_FILTER2

Table 15.136: AXI_MON8_FILTER2, Offset Address: 0x418

Bits	Name	Access	Description	Reset
7:0	axi_mon8_hit_addr_st_hi	R/W	Hit start address[39:32] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0
31:8	Reserved			

15.5.3.126 AXI_MON8_FILTER3

Table 15.137: AXI_MON8_FILTER3, Offset Address: 0x41c

Bits	Name	Access	Description	Reset
31:0	axi_mon8_hit_addr_sp_lo	R/W	Hit end address[31:0]	0x0

15.5.3.127 AXI_MON8_FILTER4

Table 15.138: AXI_MON8_FILTER4, Offset Address: 0x420

Bits	Name	Access	Description	Reset
7:0	axi_mon8_hit_addr_sp_hi	R/W	Hit end address[39:32]	0x0
31:8	Reserved			

15.5.3.128 AXI_MON8_FILTER5

Table 15.139: AXI_MON8_FILTER5, Offset Address: 0x424

Bits	Name	Access	Description	Reset
23:0	axi_mon8_hit_id_mask	R/W	hit = (hit_id & hit_id_mask) == (Ax_id & hit_id_mask)	0x0
31:24	Reserved			

15.5.3.129 AXI_MON8_FILTER6

Table 15.140: AXI_MON8_FILTER6, Offset Address: 0x428

Bits	Name	Access	Description	Reset
23:0	axi_mon8_hit_id	R/W	hit = Ax_id == hit_id	0x0
31:24	Reserved			

15.5.3.130 AXI_MON8_FILTER7

Table 15.141: AXI_MON8_FILTER7, Offset Address: 0x42c

Bits	Name	Access	Description	Reset
7:0	axi_mon8_hit_len	R/W	hit = Ax_len == hit_len	0x0
10:8	axi_mon8_hit_size	R/W	hit = Ax_size == hit_size	0x0
11	Reserved			
13:12	axi_mon8_hit_burst	R/W	hit = Ax_burst == hit_burst	0x0
31:14	Reserved			

15.5.3.131 AXI_MON8_FILTER8

Table 15.142: AXI_MON8_FILTER8, Offset Address: 0x430

Bits	Name	Access	Description	Reset
0	axi_mon8_hit_lock	R/W	hit = Ax_lock == hit_lock	0x0
3:1	Reserved			
7:4	axi_mon8_hit_cache	R/W	hit = Ax_cache == hit_cache	0x0
10:8	axi_mon8_hit_prot	R/W	hit = Ax_prot == hit_prot	0x0
11	Reserved			
15:12	axi_mon8_hit_qos	R/W	hit = Ax_qos == hit_qos	0x0
31:16	Reserved			

15.5.3.132 AXI_MON8_RPT0

Table 15.143: AXI_MON8_RPT0, Offset Address: 0x440

Bits	Name	Access	Description	Reset
31:0	axi_mon8_cycle_count	RO	AXI monitor 8 cycle count, counting after func_en assert	

15.5.3.133 AXI_MON8_RPT1

Table 15.144: AXI_MON8_RPT1, Offset Address: 0x444

Bits	Name	Access	Description	Reset
31:0	axi_mon8_hit_count	RO	AXI monitor 8 hit count, counting after func_en assert	

15.5.3.134 AXI_MON8_RPT2

Table 15.145: AXI_MON8_RPT2, Offset Address: 0x448

Bits	Name	Access	Description	Reset
31:0	axi_mon8_byte_count	RO	AXI monitor 8 byte count, counting after func_en assert, (Ax_len + 1) << Ax_size	

15.5.3.135 AXI_MON8_RPT3

Table 15.146: AXI_MON8_RPT3, Offset Address: 0x44c

Bits	Name	Access	Description	Reset
31:0	axi_mon8_latency_count	RO	AXI monitor 8 latency count, counting after func_en assert, += oustanding	

15.5.3.136 AXI_MON9_CTRL

Table 15.147: AXI_MON9_CTRL, Offset Address: 0x480

Bits	Name	Access	Description	Reset
0	axi_mon9_en	R/W	If set to 1, enables the AXI monitor function.	0x0
1	axi_mon9_clear	R/W	Clear all the counter.	0x0
2	axi_mon9_snapshot	R/W	Snapshot all the counter.	0x0
3	axi_mon0_snapshot_all	R/W	Snapshot all the counter of all AXI monitors	0x0
4	axi_mon9_irq_en	R/W	If set to 1, enables the AXI monitor interrupt.	0x0
5	axi_mon9_irq_clear	R/W	If set to 1, clears the axi_mon_irq.	0x0
6	Reserved			
7	axi_mon9_irq	RO	Assert when all axi_mon<n>_hit_sel succes.	
31:8	Reserved			

15.5.3.137 AXI_MON9_INPUT

Table 15.148: AXI_MON9_INPUT, Offset Address: 0x484

Bits	Name	Access	Description	Reset
5:0	axi_mon9_input_sel	R/W	Input/clk selection, 0 = No selection.	0x0
31:6	Reserved			

15.5.3.138 AXI_MON9_FILTER0

Table 15.149: AXI_MON9_FILTER0, Offset Address: 0x490

Bits	Name	Access	Description	Reset
9:0	axi_mon9_hit_sel	R/W	Select which conditions are used to judege hit bit [0]: addr_st/addr_sp bit [1]: id/id_mask bit [2]: len bit [3]: size bit [4]: burst bit [5]: lock bit [6]: cache bit [7]: prot bit [8]: qos bit [9]: AXI transaction cross 4KB boundary	0x0
31:10	Reserved			

15.5.3.139 AXI_MON9_FILTER1

Table 15.150: AXI_MON9_FILTER1, Offset Address: 0x494

Bits	Name	Access	Description	Reset
31:0	axi_mon9_hit_addr_st_lo	R/W	Hit start address[31:0] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0

15.5.3.140 AXI_MON9_FILTER2

Table 15.151: AXI_MON9_FILTER2, Offset Address: 0x498

Bits	Name	Access	Description	Reset
7:0	axi_mon9_hit_addr_st_hi	R/W	Hit start address[39:32] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0
31:8	Reserved			

15.5.3.141 AXI_MON9_FILTER3

Table 15.152: AXI_MON9_FILTER3, Offset Address: 0x49c

Bits	Name	Access	Description	Reset
31:0	axi_mon9_hit_addr_sp_lo	R/W	Hit end address[31:0]	0x0

15.5.3.142 AXI_MON9_FILTER4

Table 15.153: AXI_MON9_FILTER4, Offset Address: 0x4a0

Bits	Name	Access	Description	Reset
7:0	axi_mon9_hit_addr_sp_hi	R/W	Hit end address[39:32]	0x0
31:8	Reserved			

15.5.3.143 AXI_MON9_FILTER5

Table 15.154: AXI_MON9_FILTER5, Offset Address: 0x4a4

Bits	Name	Access	Description	Reset
23:0	axi_mon9_hit_id_mask	R/W	hit = (hit_id & hit_id_mask) == (Ax_id & hit_id_mask)	0x0
31:24	Reserved			

15.5.3.144 AXI_MON9_FILTER6

Table 15.155: AXI_MON9_FILTER6, Offset Address: 0x4a8

Bits	Name	Access	Description	Reset
23:0	axi_mon9_hit_id	R/W	hit = Ax_id == hit_id	0x0
31:24	Reserved			

15.5.3.145 AXI_MON9_FILTER7

Table 15.156: AXI_MON9_FILTER7, Offset Address: 0x4ac

Bits	Name	Access	Description	Reset
7:0	axi_mon9_hit_len	R/W	hit = Ax_len == hit_len	0x0
10:8	axi_mon9_hit_size	R/W	hit = Ax_size == hit_size	0x0
11	Reserved			
13:12	axi_mon9_hit_burst	R/W	hit = Ax_burst == hit_burst	0x0
31:14	Reserved			

15.5.3.146 AXI_MON9_FILTER8

Table 15.157: AXI_MON9_FILTER8, Offset Address: 0x4b0

Bits	Name	Access	Description	Reset
0	axi_mon9_hit_lock	R/W	hit = Ax_lock == hit_lock	0x0
3:1	Reserved			
7:4	axi_mon9_hit_cache	R/W	hit = Ax_cache == hit_cache	0x0
10:8	axi_mon9_hit_prot	R/W	hit = Ax_prot == hit_prot	0x0
11	Reserved			
15:12	axi_mon9_hit_qos	R/W	hit = Ax_qos == hit_qos	0x0
31:16	Reserved			

15.5.3.147 AXI_MON9_RPT0

Table 15.158: AXI_MON9_RPT0, Offset Address: 0x4c0

Bits	Name	Access	Description	Reset
31:0	axi_mon9_cycle_count	RO	AXI monitor 9 cycle count, counting after func_en assert	

15.5.3.148 AXI_MON9_RPT1

Table 15.159: AXI_MON9_RPT1, Offset Address: 0x4c4

Bits	Name	Access	Description	Reset
31:0	axi_mon9_hit_count	RO	AXI monitor 9 hit count, counting after func_en assert	

15.5.3.149 AXI_MON9_RPT2

Table 15.160: AXI_MON9_RPT2, Offset Address: 0x4c8

Bits	Name	Access	Description	Reset
31:0	axi_mon9_byte_count	RO	AXI monitor 9 byte count, counting after func_en assert, (Ax_len + 1) << Ax_size	

15.5.3.150 AXI_MON9_RPT3

Table 15.161: AXI_MON9_RPT3, Offset Address: 0x4cc

Bits	Name	Access	Description	Reset
31:0	axi_mon9_latency_count	RO	AXI monitor 9 latency count, counting after func_en assert, += oustanding	

15.5.3.151 AXI_MON10_CTRL

Table 15.162: AXI_MON10_CTRL, Offset Address: 0x500

Bits	Name	Access	Description	Reset
0	axi_mon10_en	R/W	If set to 1, enables the AXI monitor function.	0x0
1	axi_mon10_clear	R/W	Clear all the counter.	0x0
2	axi_mon10_snapshot	R/W	Snapshot all the counter.	0x0
3	axi_mon0_snapshot_all	R/W	Snapshot all the counter of all AXI monitors	0x0
4	axi_mon10_irq_en	R/W	If set to 1, enables the AXI monitor interrupt.	0x0
5	axi_mon10_irq_clear	R/W	If set to 1, clears the axi_mon_irq.	0x0
6	Reserved			
7	axi_mon10_irq	RO	Assert when all axi_mon<n>_hit_sel succes.	
31:8	Reserved			

15.5.3.152 AXI_MON10_INPUT

Table 15.163: AXI_MON10_INPUT, Offset Address: 0x504

Bits	Name	Access	Description	Reset
5:0	axi_mon10_input_sel	R/W	Input/clk selection, 0 = No selection.	0x0
31:6	Reserved			

15.5.3.153 AXI_MON10_FILTER0

Table 15.164: AXI_MON10_FILTER0, Offset Address: 0x510

Bits	Name	Access	Description	Reset
9:0	axi_mon10_hit_sel	R/W	Select which conditions are used to judege hit bit [0]: addr_st/addr_sp bit [1]: id/id_mask bit [2]: len bit [3]: size bit [4]: burst bit [5]: lock bit [6]: cache bit [7]: prot bit [8]: qos bit [9]: AXI transaction cross 4KB boundary	0x0
31:10	Reserved			

15.5.3.154 AXI_MON10_FILTER1

Table 15.165: AXI_MON10_FILTER1, Offset Address: 0x514

Bits	Name	Access	Description	Reset
31:0	axi_mon10_hit_addr_st_lo	R/W	Hit start address[31:0] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0

15.5.3.155 AXI_MON10_FILTER2

Table 15.166: AXI_MON10_FILTER2, Offset Address: 0x518

Bits	Name	Access	Description	Reset
7:0	axi_mon10_hit_addr_st_hi	R/W	Hit start address[39:32] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0
31:8	Reserved			

15.5.3.156 AXI_MON10_FILTER3

Table 15.167: AXI_MON10_FILTER3, Offset Address: 0x51c

Bits	Name	Access	Description	Reset
31:0	axi_mon10_hit_addr_sp_lo	R/W	Hit end address[31:0]	0x0

15.5.3.157 AXI_MON10_FILTER4

Table 15.168: AXI_MON10_FILTER4, Offset Address: 0x520

Bits	Name	Access	Description	Reset
7:0	axi_mon10_hit_addr_sp_hi	R/W	Hit end address[39:32]	0x0
31:8	Reserved			

15.5.3.158 AXI_MON10_FILTER5

Table 15.169: AXI_MON10_FILTER5, Offset Address: 0x524

Bits	Name	Access	Description	Reset
23:0	axi_mon10_hit_id_mask	R/W	hit = (hit_id & hit_id_mask) == (Ax_id & hit_id_mask)	0x0
31:24	Reserved			

15.5.3.159 AXI_MON10_FILTER6

Table 15.170: AXI_MON10_FILTER6, Offset Address: 0x528

Bits	Name	Access	Description	Reset
23:0	axi_mon10_hit_id	R/W	hit = Ax_id == hit_id	0x0
31:24	Reserved			

15.5.3.160 AXI_MON10_FILTER7

Table 15.171: AXI_MON10_FILTER7, Offset Address: 0x52c

Bits	Name	Access	Description	Reset
7:0	axi_mon10_hit_len	R/W	hit = Ax_len == hit_len	0x0
10:8	axi_mon10_hit_size	R/W	hit = Ax_size == hit_size	0x0
11	Reserved			
13:12	axi_mon10_hit_burst	R/W	hit = Ax_burst == hit_burst	0x0
31:14	Reserved			

15.5.3.161 AXI_MON10_FILTER8

Table 15.172: AXI_MON10_FILTER8, Offset Address: 0x530

Bits	Name	Access	Description	Reset
0	axi_mon10_hit_lock	R/W	hit = Ax_lock == hit_lock	0x0
3:1	Reserved			
7:4	axi_mon10_hit_cache	R/W	hit = Ax_cache == hit_cache	0x0
10:8	axi_mon10_hit_prot	R/W	hit = Ax_prot == hit_prot	0x0
11	Reserved			
15:12	axi_mon10_hit_qos	R/W	hit = Ax_qos == hit_qos	0x0
31:16	Reserved			

15.5.3.162 AXI_MON10_RPT0

Table 15.173: AXI_MON10_RPT0, Offset Address: 0x540

Bits	Name	Access	Description	Reset
31:0	axi_mon10_cycle_count	RO	AXI monitor 10 cycle count, counting after func_en assert	

15.5.3.163 AXI_MON10_RPT1

Table 15.174: AXI_MON10_RPT1, Offset Address: 0x544

Bits	Name	Access	Description	Reset
31:0	axi_mon10_hit_count	RO	AXI monitor 10 hit count, counting after func_en assert	

15.5.3.164 AXI_MON10_RPT2

Table 15.175: AXI_MON10_RPT2, Offset Address: 0x548

Bits	Name	Access	Description	Reset
31:0	axi_mon10_byte_count	RO	AXI monitor 10 byte count, counting after func_en assert, (Ax_len + 1) << Ax_size	

15.5.3.165 AXI_MON10_RPT3

Table 15.176: AXI_MON10_RPT3, Offset Address: 0x54c

Bits	Name	Access	Description	Reset
31:0	axi_mon10_latency_count	RO	AXI monitor 10 latency count, counting after func_en assert, += oustanding	

15.5.3.166 AXI_MON11_CTRL

Table 15.177: AXI_MON11_CTRL, Offset Address: 0x580

Bits	Name	Access	Description	Reset
0	axi_mon11_en	R/W	If set to 1, enables the AXI monitor function.	0x0
1	axi_mon11_clear	R/W	Clear all the counter.	0x0
2	axi_mon11_snapshot	R/W	Snapshot all the counter.	0x0
3	axi_mon0_snapshot_all	R/W	Snapshot all the counter of all AXI monitors	0x0
4	axi_mon11_irq_en	R/W	If set to 1, enables the AXI monitor interrupt.	0x0
5	axi_mon11_irq_clear	R/W	If set to 1, clears the axi_mon_irq.	0x0
6	Reserved			
7	axi_mon11_irq	RO	Assert when all axi_mon<n>_hit_sel succes.	
31:8	Reserved			

15.5.3.167 AXI_MON11_INPUT

Table 15.178: AXI_MON11_INPUT, Offset Address: 0x584

Bits	Name	Access	Description	Reset
5:0	axi_mon11_input_sel	R/W	Input/clk selection, 0 = No selection.	0x0
31:6	Reserved			

15.5.3.168 AXI_MON11_FILTER0

Table 15.179: AXI_MON11_FILTER0, Offset Address: 0x590

Bits	Name	Access	Description	Reset
9:0	axi_mon11_hit_sel	R/W	Select which conditions are used to judege hit bit [0]: addr_st/addr_sp bit [1]: id/id_mask bit [2]: len bit [3]: size bit [4]: burst bit [5]: lock bit [6]: cache bit [7]: prot bit [8]: qos bit [9]: AXI transaction cross 4KB boundary	0x0
31:10	Reserved			

15.5.3.169 AXI_MON11_FILTER1

Table 15.180: AXI_MON11_FILTER1, Offset Address: 0x594

Bits	Name	Access	Description	Reset
31:0	axi_mon11_hit_addr_st_lo	R/W	Hit start address[31:0] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0

15.5.3.170 AXI_MON11_FILTER2

Table 15.181: AXI_MON11_FILTER2, Offset Address: 0x598

Bits	Name	Access	Description	Reset
7:0	axi_mon11_hit_addr_st_hi	R/W	Hit start address[39:32] hit = (Ax_addr >= hit_addr_st) && (Ax_addr < hit_addr_sp)	0x0
31:8	Reserved			

15.5.3.171 AXI_MON11_FILTER3

Table 15.182: AXI_MON11_FILTER3, Offset Address: 0x59c

Bits	Name	Access	Description	Reset
31:0	axi_mon11_hit_addr_sp_lo	R/W	Hit end address[31:0]	0x0

15.5.3.172 AXI_MON11_FILTER4

Table 15.183: AXI_MON11_FILTER4, Offset Address: 0x5a0

Bits	Name	Access	Description	Reset
7:0	axi_mon11_hit_addr_sp_hi	R/W	Hit end address[39:32]	0x0
31:8	Reserved			

15.5.3.173 AXI_MON11_FILTER5

Table 15.184: AXI_MON11_FILTER5, Offset Address: 0x5a4

Bits	Name	Access	Description	Reset
23:0	a xi_mon11_hit_id_mask	R/W	hit = (hit_id & hit_id_mask) == (Ax_id & hit_id_mask)	0x0
31:24	Reserved			

15.5.3.174 AXI_MON11_FILTER6

Table 15.185: AXI_MON11_FILTER6, Offset Address: 0x5a8

Bits	Name	Access	Description	Reset
23:0	axi_mon11_hit_id	R/W	hit = Ax_id == hit_id	0x0
31:24	Reserved			

15.5.3.175 AXI_MON11_FILTER7

Table 15.186: AXI_MON11_FILTER7, Offset Address: 0x5ac

Bits	Name	Access	Description	Reset
7:0	axi_mon11_hit_len	R/W	hit = Ax_len == hit_len	0x0
10:8	axi_mon11_hit_size	R/W	hit = Ax_size == hit_size	0x0
11	Reserved			
13:12	axi_mon11_hit_burst	R/W	hit = Ax_burst == hit_burst	0x0
31:14	Reserved			

15.5.3.176 AXI_MON11_FILTER8

Table 15.187: AXI_MON11_FILTER8, Offset Address: 0x5b0

Bits	Name	Access	Description	Reset
0	axi_mon11_hit_lock	R/W	hit = Ax_lock == hit_lock	0x0
3:1	Reserved			
7:4	axi_mon11_hit_cache	R/W	hit = Ax_cache == hit_cache	0x0
10:8	axi_mon11_hit_prot	R/W	hit = Ax_prot == hit_prot	0x0
11	Reserved			
15:12	axi_mon11_hit_qos	R/W	hit = Ax_qos == hit_qos	0x0
31:16	Reserved			

15.5.3.177 AXI_MON11_RPT0

Table 15.188: AXI_MON11_RPT0, Offset Address: 0x5c0

Bits	Name	Access	Description	Reset
31:0	axi_mon11_cycle_count	RO	AXI monitor 11 cycle count, counting after func_en assert	

15.5.3.178 AXI_MON11_RPT1

Table 15.189: AXI_MON11_RPT1, Offset Address: 0x5c4

Bits	Name	Access	Description	Reset
31:0	axi_mon11_hit_count	RO	AXI monitor 11 hit count, counting after func_en assert	

15.5.3.179 AXI_MON11_RPT2

Table 15.190: AXI_MON11_RPT2, Offset Address: 0x5c8

Bits	Name	Access	Description	Reset
31:0	axi_mon11_byte_count	RO	AXI monitor 11 byte count, counting after func_en assert, (Ax_len + 1) << Ax_size	

15.5.3.180 AXI_MON11_RPT3

Table 15.191: AXI_MON11_RPT3, Offset Address: 0x5cc

Bits	Name	Access	Description	Reset
31:0	axi_mon11_latency_count	RO	AXI monitor 11 latency count, counting after func_en assert, += oustanding	

15.6 DDRC Registers

15.6.1 DDRC Register Overview

Base address: 0x0800_4000

Table 15.192: DDRC Registers Overview

Name	Address Offset	Description
DRAM_REF_CTRL	0x064	DRAM refresh parameter
DRAM_MRD0	0x0dc	DRAM MR value
DRAM_MRD1	0x0e0	DRAM MR value

15.6.2 DDRC Register Description

Base address: 0x0800_4000

15.6.2.1 DRAM_REF_CTRL

Table 15.193: DRAM_REF_CTRL, Offset Address: 0x064

Bits	Name	Access	Description	Reset
9:0	t_rfc	R/W	Specify tRFC Unit: ddr core clock cycles	0x8c
15:10	Reserved			
27:16	t_refi	R/W	Specify tREFI Unit: 32 ddr core clocks	0x62
31:28	Reserved			

15.6.2.2 DRAM_MRD0

Table 15.194: DRAM_MRD0, Offset Address: 0x0dc

Bits	Name	Access	Description	Reset
15:0	ddr_mr1	R/W	DDR3: Write value for MR1 register	0x510
31:16	ddr_mr0	R/W	DDR3: Write value for MR0 register	0x0

15.6.2.3 DRAM_MRD1

Table 15.195: DRAM_MRD1, Offset Address: 0x0e0

Bits	Name	Access	Description	Reset
15:0	ddr_mr3	R/W	DDR3: Write value for MR3 register	0x0
31:16	ddr_mr2	R/W	DDR3: Write value for MR2 register	0x0

CHAPTER

SIXTEEN

SPI NOR FLASH CONTROLLER

16.1 Overview

Provides off-chip SPI NOR Flash data access.

16.2 Features

- Supports external 1 chip select.
- Supports Dual/Qual read and write operations.
- Supports devices of various specifications.
  - Supports 3Byte address devices and 4Byte address devices.
  - Supports devices with a maximum capacity of 256MB.
- Support BOOT function.

16.3 Function Description

16.3.1 Interface Description

The SPI NOR Flash controller can support three SPI NOR interface types, which are Standard SPI, Dual SPI interface mode and Qual SPI interface mode.

16.3.1.1 Standard SPI Interface Mode

Standard SPI interface mode has 1bit data input line and 1bit data output line. The chart shows the write operation timing diagram of the Standard SPI interface mode, and the read operation timing diagram of the Standard SPI interface mode.

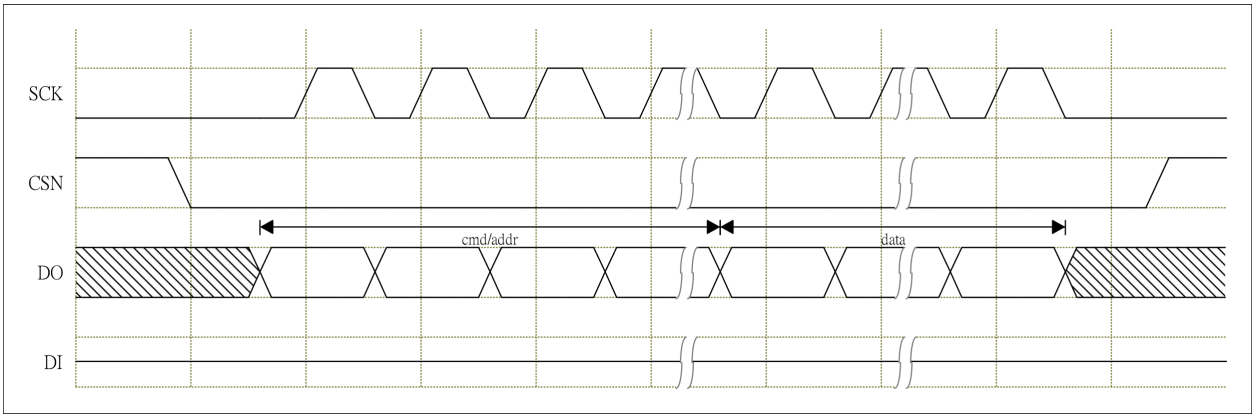


Diagram 16.1: Standard SPI interface mode write operation timing diagram

Timing description:

- command/address/dummy cycles are output on the DO line in single-bit serial mode.
- Data is output on the DO line in single-bit serial mode.

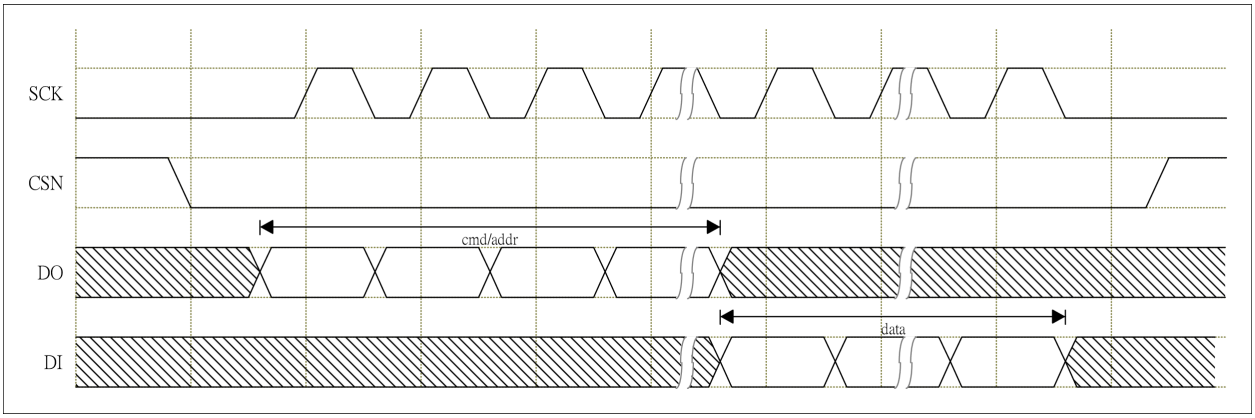


Diagram 16.2: Standard SPI interface mode read operation timing diagram

Timing description:

- command/address/dummy cycles are output on the DO line in single-bit serial mode.
- Data is input on the DI line in single-bit serial mode.

16.3.1.2 Dual-Input SPI Interface Mode

Dual Input SPI interface mode, parallel 2bit data line in the data input stage. *Dual-Input SPI interface timing* is the operation sequence diagram of Dual Input SPI interface mode.

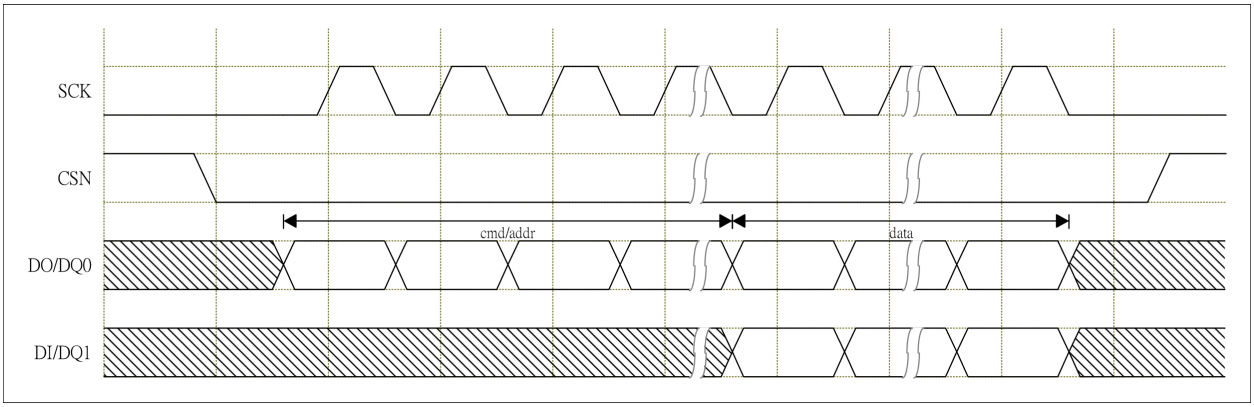


Diagram 16.3: Dual-Input SPI interface timing

Timing description:

- command/address/dummy cycles are output on the DO line in single-bit serial mode.
- Data is input (read) on the DO/DI line in 2 Bits.

16.3.1.3 Dual-IO SPI Interface Mode

Dual IO SPI interface mode, parallel 2-bit data lines in the address output and data input stages. *Dual-IO SPI interface timing* is the operation sequence diagram of Dual IO SPI interface mode.

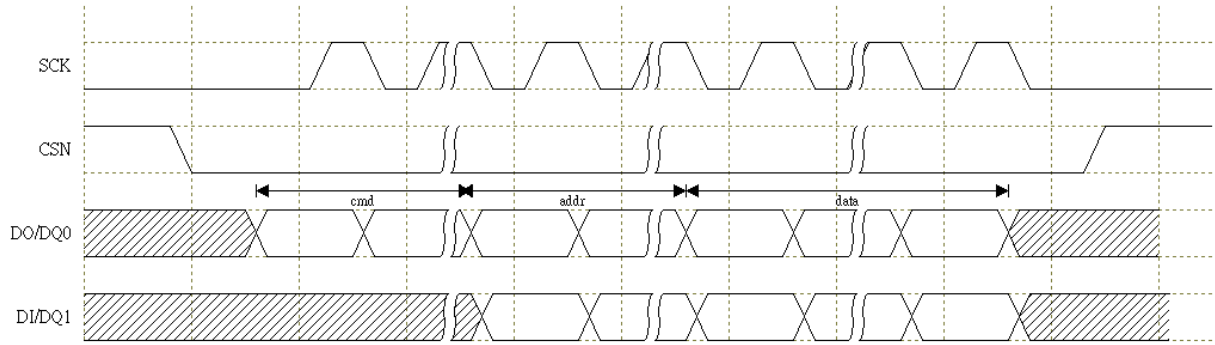


Diagram 16.4: Dual-IO SPI interface timing

Timing description:

- Command is output on the DO line in single-bit serial mode.

- address/dummy cycles/Data is output (written) or input (read) on the DO/DI line in 2 Bits mode.

16.3.1.4 Quad-Input SPI Interface Mode

Quad Input SPI interface mode, parallel 4bit data lines in the data input stage. *Quad-Input SPI mode timing diagram* is the Quad Input SPI interface mode operation sequence diagram.

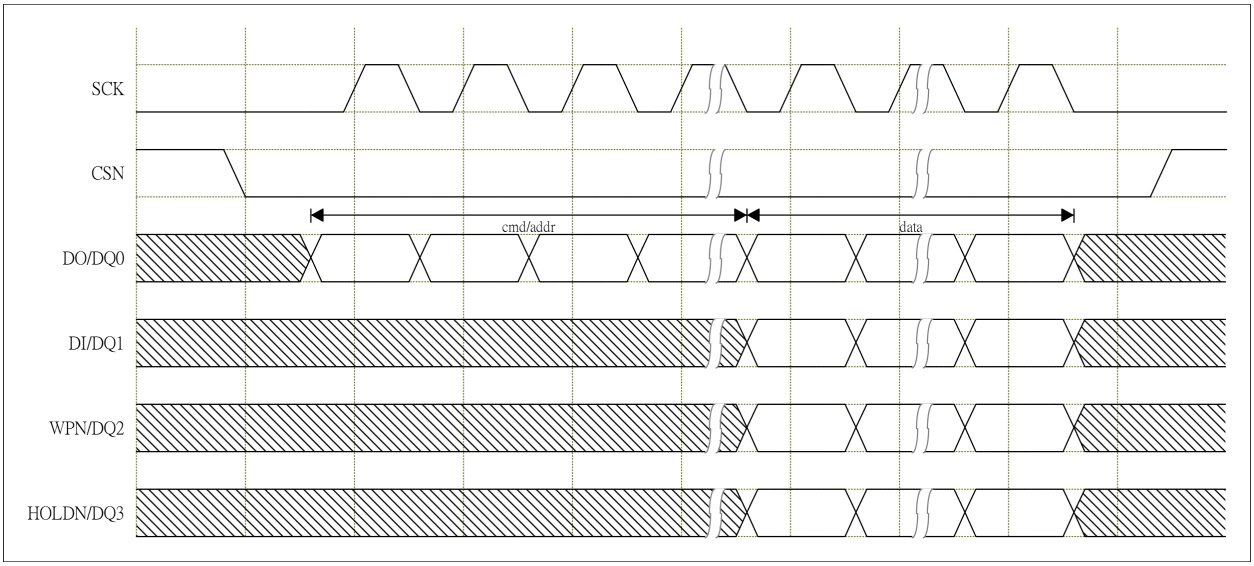


Diagram 16.5: Quad-Input SPI mode timing diagram

Timing description:

- command/address/dummy cycles are output on the DO line in single-bit serial mode.
- Data is input (read) in DO/DI/WPN/HOLDN in 4 Bits mode.

16.3.1.5 Quad-IO SPI Interface Mode

Quad IO SPI interface mode, parallel 4bit data lines in the address output and data input stages. *Quad-IO SPI mode timing diagram* is the Quad IO SPI interface mode operation sequence diagram.

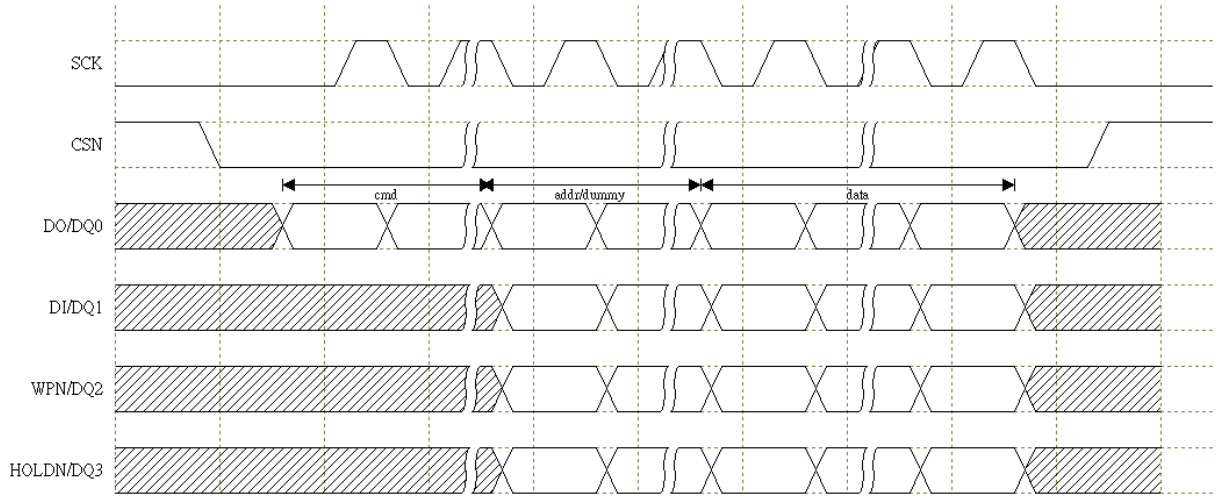


Diagram 16.6: Quad-IO SPI mode timing diagram

Timing description:

- Command is output on the DO line in single-bit serial mode.
- address/dummy cycles/Data is output (write) or input (read) in DO/DI/WPN/HOLDN in 4 Bits mode.

16.3.2 Boot Function

The SPI NOR Boot data is located at the chip address 0x1000_0000~0x1FFF_FFFF, which is directly mapped to the continuous address space 0x0000_0000~0x0FFF_FFFF of the SPI NOR Flash. SPI_NOR Flash can support up to 256MB. If you need to use SPI_NOR Flash larger than 16MB, you need to use the 4 bytes address mode. The reset state of the chip is 3bytes address mode, and the 4bytes address mode needs to be enabled through configuration, so SPI_NOR Flash needs to support 3bytes/4bytes address mode.

16.3.3 Register Operations

The software configures operation-related registers, such as operation commands, addresses, etc., and finally configures the reg_go_busy register to issue operations. The controller issues operations to the device based on the software configuration value.

16.3.4 DMA Operations

- DMMR read mode

  When the SPI_NOR Flash Controller is in DMMR mode, the SPI_NOR Flash space is directly mapped to the chip address space 0x1000_0000~0x1FFF_FFFF. System DMA can use memory-to-memory mode to move SPI_NOR data to DDR.

- Non-DMMR read and write mode

  Instructions, addresses and data need to be sent and received through FF_PORT. It is necessary to configure the controller register to select read instructions, write instructions, instruction length, and data length, then write FF_PORT through CPU or DMA to issue instructions and addresses, and write/read FF_PORT to send/receive data.

16.4 Work Process

16.4.1 Initialization Process

- Step 1. If you need to adjust the Timing parameters, configure the SPI Clock Divider according to the device.
- Step 2. Configure the interrupt control register.

16.4.2 Device Status Register Operation

- Step 1. Configure the transmission data length.
- Step 2. Transmission mode related configuration.
- Step 3. Configure reg_go_busy.
- Step 4. Write the transfer to cache.
- Step 5. Detect INT_STS and wait for the operation to complete.

16.4.3 SPI NOR Flash Address Mode Switching Process

For SPI NOR Flash devices, it supports two Flash address modes of 3 Byte and 4 Byte. The address mode can be dynamically switched through the configuration register after the chip is started. The steps to switch the Flash address mode after the chip starts up are as follows:

- Step 1. No Flash operation or ensure that the previous operation on the Flash device is completed.
- Step 2. According to the device requirements, use the register operation mode to configure the relevant registers of the device and send specific commands to configure the Flash to enter 4 Byte mode.
- Step 3. Configure the SPI NOR Flash controller [reg_byte4en] format to 4 Byte mode and complete the switch from 3 Byte mode to 4 Byte mode.

16.4.4 DMA Read Operation Process

- Step 1. Disable dmmr mode, disable dma_en.
- Step 2. Write 1 to FF_PT to clear the FIFO and reset the read and write indicators.
- Step 3. Enable dmmr mode.
- Step 4. Configure the system DMA for mem-to-mem migration. DST_TR_WIDTH = 0x2 (transaction width is 32bit), DST_MSIZE = 0x0 (burst transaction length =1), BLOCK_TS = TRAN_NUM/4 -1. In fact, BLOCK_TS/ DST_TR_WIDTH/ DST_MSIZE needs to be configured appropriately according to the actual transmission length.
- Step 5. Enable system DMA for selected channels. Start moving.
- Step 6. Wait for DMA corresponding channel interrupt, which indicates that DMA reading is completed.

16.4.5 DMA Write Operation Process

- Step 1. Disable dmmr mode, disable dma_en.
- Step 2. Write 1 to FF_PT to clear the FIFO and reset the read and write indicators.
- Step 3. Configure the system DMA channel mapping to map the selected DMA channel to 39:dma_req_spi_nor.
- Step 4. Configure the system DMA for peri-to-mem migration. DST_TR_WIDTH = 0x2 (transaction width is 32bit), DST_MSIZE = 0x0 (burst transaction length =1), BLOCK_TS = TRAN_NUM/4 -1. In fact, BLOCK_TS/ DST_TR_WIDTH/ DST_MSIZE needs to be configured appropriately according to the actual transmission length.
- Step 5. Enable system DMA for selected channels.
- Step 6. Configure the SPI_NOR register TRAN_NUM, which does not contain instructions and addresses.
- Step 7. Configure SPI_NOR register TRAN_CSR, tran_mode = 0x2 (tx only), fast_mode, bus_width, addr_bn, dma_en=0 and reg_go_busy. Ex: TRAN_CSR = 0x0000BC2A.
- Step 8. Write command and address to SPI_NOR register FF_PORT.
- Step 9. Check the SPI_NOR register rdat_ff_pt for 0, and ensure that the command and address have been sent.
- Step 10. Configure SPI_NOR register TRAN_CSR to enable dma_en.
- Step 11. Detect the SPI_NOR register INT_STS and wait for the operation to complete, indicating that the buffer content has been written to the device.

16.4.6 Other things to note

- Do not change the relevant register configuration before the device operation is completed, otherwise it may cause abnormal operation.

16.5 Register Overview

Base Address 0x10000000

Table 16.1: Registers Overview

Name	Address Offset	Description
SPI_CTRL	0x000	SPI_NOR Operation control
CE_CTRL	0x004	CE Operation control
DLY_CTRL	0x008	Delay control
DMMR_CTRL	0x00c	DMMR Mode control
TRAN_CSR	0x010	transport control
TRAN_NUM	0x014	Number of frames transmitted
FF_PORT	0x018	FIFO write read port
FF_PT	0x020	FIFO indicator status
INT_STS	0x028	interrupt status
INT_EN	0x02c	Interrupt enable

16.6 Register Description

16.6.1 SPI_CTRL

Table 16.2: SPI_CTRL, Offset Address: 0x000

Bits	Name	Access	Description	Reset
10:0	sck_div	R/W	SPI Clock Divider SCK frequency = HCLK frequency / (2(SckDiv+ 1))	0x9
11	Reserved			
12	cpha	R/W	Clock Phase 0: After the chip select is valid, starts sampling data at the first clock edge of SCK. 1: After the chip select is valid, starts sampling data at the second clock edge of SCK.	0x0
13	cpol	R/W	Clock Polarity 0: SCK Low level when idle 1: SCK High level when idle	0x0
14	hold_o	R/W	HOLD Pin output level	0x1
15	wp_o	R/W	WP Pin output level	0x1
19:16	frame_len	R/W	Frame length sent and received. 0 indicates that the frame length is 16 bits; a frame length of 1 is not supported.	0x8
20	lsb_first	R/W	LSBF: Least Significant Bit First 0: Frame MSB first 1: Frame LSB first	0x0
21	srst	R/W	Write 1 to reset each state machine and interrupt flag bit	0x0
31:22	Reserved			

16.6.2 CE_CTRL

Table 16.3: CE_CTRL, Offset Address: 0x004

Bits	Name	Access	Description	Reset
0	ce_manual	R/W	Control the level value of CE pin	0x0
1	ce_manual_en	R/W	CE Manual Enable 0: The level value of the CE pin is controlled by the hardware state machine. 1: The level value of CE pin is controlled by CEManual register	0x0
31:2	Reserved			

16.6.3 DLY_CTRL

Table 16.4: DLY_CTRL, Offset Address: 0x008

Bits	Name	Access	Description	Reset
3:0	frame_interval	R/W	Control the frame spacing between two adjacent frames of data : T= TSCK * FmIntvl (no SCK pulse within the frame spacing). The frame spacing between two adjacent frames of data. When it is 0, there is no frame spacing.	0x0
7:4	Reserved			
11:8	cet	R/W	CET controls the time that CE is valid in advance relative to the first clock edge of SCK before a transmission starts and the time that it continues to be valid relative to the last clock edge of SCK after the transmission ends. This time is calculated as T= TSCK * (CET+1)	0x3
13:12	smp_en_dly	R/W	Receive sampling delay option. Delay sample cycle (IP working clock) behind the positive edge of SCK to perform sampling.	0x0
14	rx_pipe_ctrl	R/W	Receive sample clock edge option. 0: Normal sampling 1: Adopt SCK negative edge sampling to achieve high-speed transmission	0x0
31:15	Reserved			

16.6.4 DMMR_CTRL

Table 16.5: DMMR_CTRL, Offset Address: 0x00c

Bits	Name	Access	Description	Reset
0	dmmr_mode	R/W	dmmr_mode. When this bit is 1, the read address on the AHB will be directly mapped to the SPI Flash, and the controller automatically reads data from the corresponding address of the SPI Flash without the need for software to set related commands and addresses. At this time, the SPI Flash can be used as a ROM. When DMMR is 1, the registers in the IP can be written but not read.	0x1
31:1	Reserved			

16.6.5 TRAN_CSR

Table 16.6: TRAN_CSR, Offset Address: 0x010

Bits	Name	Access	Description	Reset
1:0	tran_mode	R/W	Transfer Mode 00: No Tx, No Rx 01: Rx only 10: Tx only 11: Tx and Rx TranMode indicates the sending and receiving mode of transmitted data except commands and addresses.	0x0
2	Reserved			
3	fast_mode	R/W	FastMode: 0: Normal Mode 1: Fast Mode	0x0
5:4	bus_width	R/W	Bus Width 00: 1 bit bus 01: 2 bit bus 10: 4 bit bus 11: Reserved	0x0
6	dma_en	R/W	0: DMA Disable 1: DMA Enable When TranMode is 11 (sending and receiving at the same time), DMA transmission is not supported.	0x0
7	miso_cked	R/W	miso_i pin level value	0x0
10:8	addr_bn	R/W	Address Byte Number Indicates the number of bytes in the current Flash transmission address field, 0 indicates no address field.	0x3
11	with_cmd	R/W	With Command 0: The current transfer is without a command. 1: The current transport carries the command.	0x1
13:12	ff_trg_lvl	R/W	FFTrgLvl controls the conditions under which the FIFO generates interrupts and DMA requests. 00: 1 Byte 01: 2 Bytes 10: 4 Bytes 11: 8 Bytes For Transmit, interrupts and DMA requests are generated when the number of free Bytes in the FIFO is greater than or equal to the number of Bytes defined by FFTrgLvl; For Receive, interrupts and DMA requests are generated when the number of valid Bytes in the FIFO is greater than or equal to the number of Bytes defined by FFTrgLvl.	0x3
14	Reserved			
15	go_busy	R/W	Writing 0 to this bit has no effect, writing 1 sets this bit to 1 and starts a transfer, and the bit is automatically cleared after the transfer is completed. Before initiating a new transfer, the software should query this register. Only when the register is 0 can a new transfer be initiated.	0x0
19:16	dummy_cyc	R/W	dummy cycle count	0x0
20	byte4en	R/W	4 bytes address cycle enable in dmmr_mode	0x0
21	byte4cmd	R/W	4 bytes address cmd enable in dmmr_mode	0x0
31:22	Reserved			

16.6.6 TRAN_NUM

Table 16.7: TRAN_NUM, Offset Address: 0x014

Bits	Name	Access	Description	Reset
15:0	rdat_tran_num	R/W	In non-dmmr_mode, TRAN_NUM is the number of frames sent and received in one transmission.	0x0
31:16	Reserved			

16.6.7 FF_PORT

Table 16.8: FF_PORT, Offset Address: 0x018

Bits	Name	Access	Description	Reset
31:0	rdat_ff_port	R/W	FIFO write read address	0x0

16.6.8 FF_PT

Table 16.9: FF_PT, Offset Address: 0x020

Bits	Name	Access	Description	Reset
3:0	rdat_ff_pt	R/W	Read to get the number of valid data bytes in the FIFO, write to clear the FIFO.	0x0
7:4	Reserved			
9:8	wrcnt	R/W	Current fifo, write byte offset indicator status.	0x0
12:10	rdpt	R/W	Current fifo, read byte offset indicator status	0x0
31:13	Reserved			

16.6.9 INT_STS

Table 16.10: INT_STS, Offset Address: 0x028

Bits	Name	Access	Description	Reset
0	tran_done_int	R/W	This interrupt is generated every time a frame of data is successfully sent.	0x0
1	Reserved			
2	rdff_int	R/W	This interrupt is generated every time a frame of data is successfully received.	0x0
3	wrff_int	R/W	After receiving the interrupt, the CPU writes frame data to the FIFO.	0x0
4	rx_frame_int	R/W	After receiving the interrupt, the CPU reads the frame data from the FIFO.	0x0
5	tx_frame_int	R/W	This interrupt marks the completion of a transfer.	0x0
31:6	Reserved			

16.6.10 INT_EN

Table 16.11: INT_EN, Offset Address: 0x02c

Bits	Name	Access	Description	Reset
0	tran_done_int_en	R/W	Enable Interrupt tran_done_int	0x0
1	Reserved			
2	rdff_int_en	R/W	Enable Interrupt rdff_int	0x0
3	wrff_int_en	R/W	Enable Interrupt wrff_int	0x0
4	rx_frame_int_en	R/W	Enable Interrupt rx_frame_int	0x0
5	tx_frame_int_en	R/W	Enable Interrupt tx_frame_int	0x0
31:6	Reserved			

CHAPTER

SEVENTEEN

SPI NAND FLASH CONTROLLER

17.1 Overview

Provides off-chip SPI NAND Flash data access.

17.2 Features

- Supports external 1 chip select.
- Support SPI NAND Flash x1/x2/x4 read and write operations.
- Supports various specifications of SPI NAND Flash devices.
  - Supports 2KB and 4KB devices.
  - Supports 64Pages/Block, 128Pages/Block devices.
- Supports BOOT function of SPI NAND.

17.3 Function Description

17.3.1 Interface Description

The SPI NAND Flash controller can support three SPI NAND interface types, namely Standard SPI, X2 interface mode and X4 interface mode.

17.3.1.1 Standard SPI Interface Mode

Standard SPI interface mode has 1-bit data input line and 1-bit data output line. The chart *Standard SPI interface mode write operation timing* is the write operation timing diagram of the Standard SPI interface mode, and the diagram *Standard SPI interface mode read operation timing* is the read operation timing diagram of the Standard SPI interface mode.

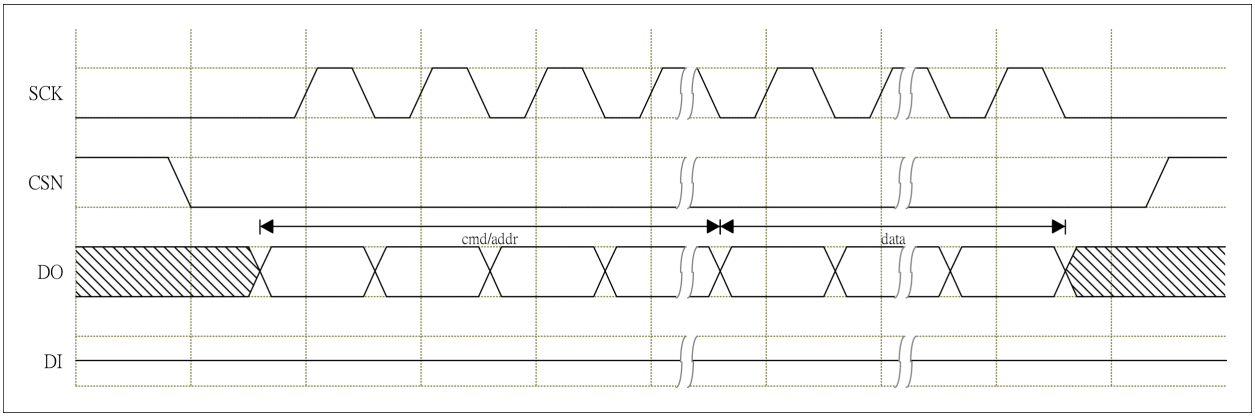


Diagram 17.1: Standard SPI interface mode write operation timing

Timing description:

- command/address/dummy cycles are output on the DO line in single-bit serial mode.
- Data is output on the DO line in single-bit serial mode.

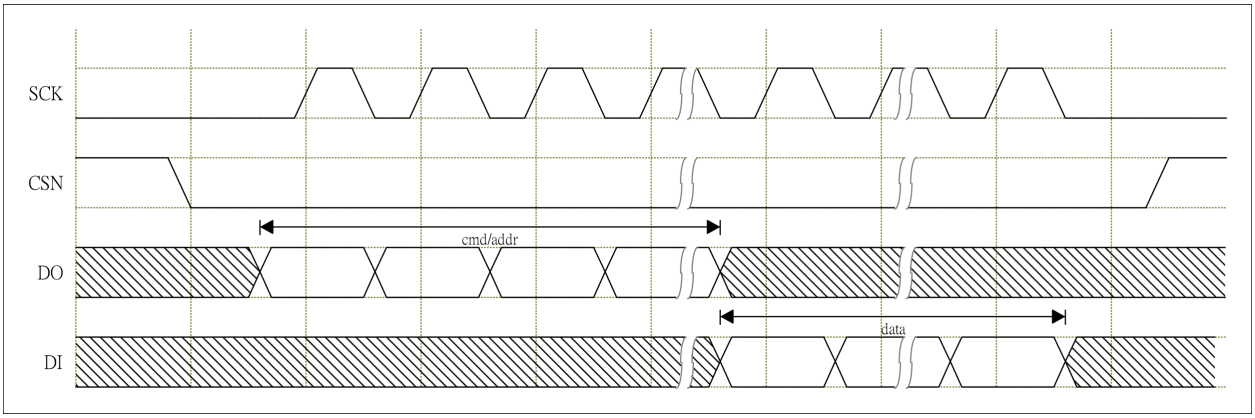


Diagram 17.2: Standard SPI interface mode read operation timing

Timing description:

- command/address/dummy cycles are output on the DO line in single-bit serial mode.
- Data is input on the DI line in single-bit serial mode.

17.3.1.2 X2 Interface Mode

X2 interface mode shares 2-bit data input and output lines. Diagram *SPI Nand X2 interface mode operation timing* is the X2 interface mode operation sequence diagram.

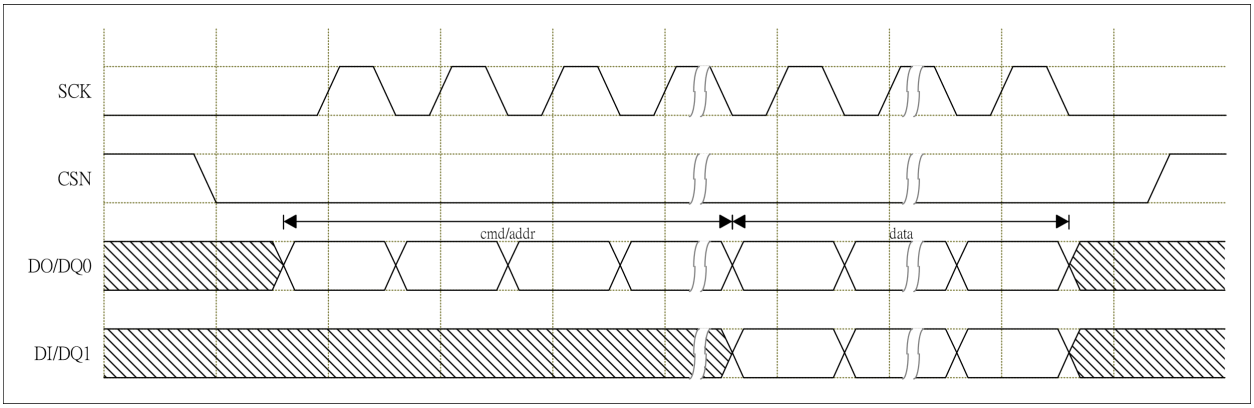


Diagram 17.3: SPI Nand X2 interface mode operation timing

Timing description:

- command/address/dummy cycles are output on the DO line in single-bit serial mode.
- Data is output (written) or input (read) on the DO/DI line in 2 Bits mode.

17.3.1.3 X4 Interface Mode

X4 interface mode shares 4-bit data input and output lines. Diagram *SPI Nand X4 interface mode operation timing* is the X4 interface mode operation sequence diagram.

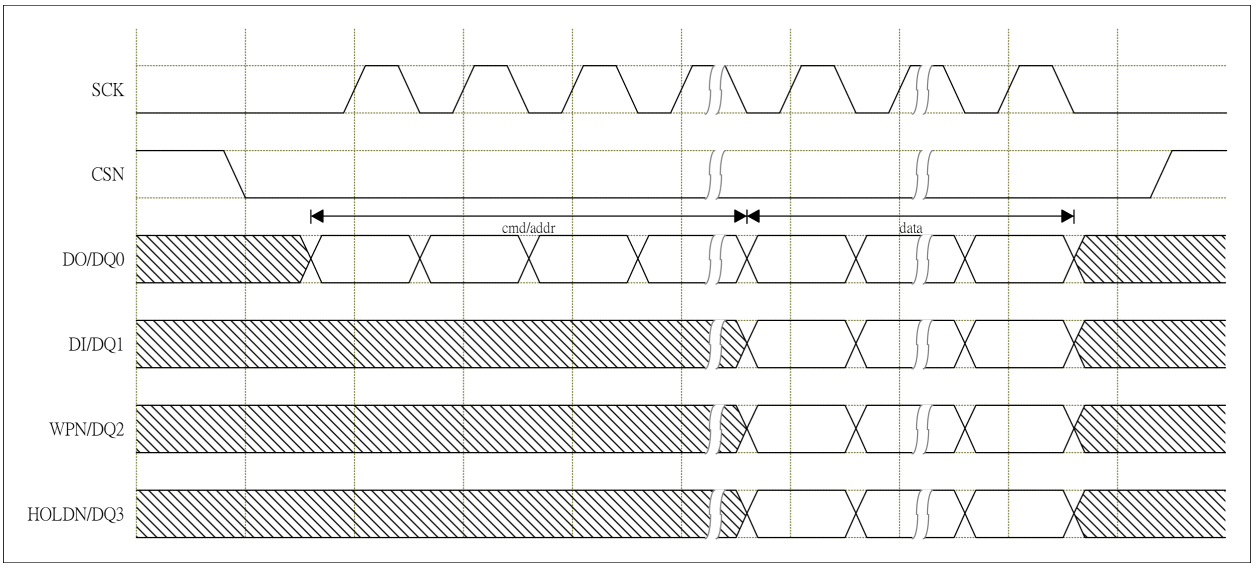


Diagram 17.4: SPI Nand X4 interface mode operation timing

Timing description:

- command/address/dummy cycles are output on the DO line in single-bit serial mode.
- Data is output (written) or input (read) on the DO/DI/WPN/HOLDN line in 4 Bits mode.

17.3.2 SPI NAND FLASH Address Description

When issuing SPI NAND Flash read and write operations, the column address is issued according to the specific operation.

- Write operation: Column address is configured during LOAD operation, row address is configured during PROGRAM operation.
- Read operation: configure the row address during PAGE READ TO CACHE operation and configure the column address during READ operation.
- Address issuance is completed by the controller, and the software needs to configure reg_trx_cmd_idx, and address configuration reg_trx_cmd_cnt0, reg_trx_cmd_cnt1 according to the operation instructions.

17.3.3 Boot Function

Since the SPI NAND Flash address space is discontinuous and there is the possibility of bad blocks, Boot data cannot be directly mapped to Flash.

Supports adaptive Boot function, which can automatically adapt hardware information based on Block0 data. The controller requires that physical Block0 must be a good block, and other blocks can be automatically skipped if they are bad blocks.

17.3.4 Register Operations

The software configures operation-related registers, such as operation commands, addresses, etc., and finally configures the reg_trx_start register to issue operations. The controller issues operations to the device based on the software configuration value. If data needs to be transferred to the device, an internal DMA operation is used to transfer the data.

17.3.5 Built-in DMA Operation Mode

Supports built-in system DMA mode for read and write operations to increase access speed. In this way, on-chip or off-chip memory space can be directly accessed through the bus.

- Step 1: Configure the DMA channel to use.
- Step 2: Configure source and destination addresses.
- Step 3: Configure the transmission format and data length.

17.3.6 TIMEOUT Function

The software sets a maximum 1 second TIMEOUT mechanism as a protection when the device does not respond normally.

17.4 Work Process

17.4.1 Initialization Process

- Step 1: (If the Timing parameters need to be adjusted) Configure the timing registers reg_trx_sck_h and reg_trx_sck_l according to the device.
- Step 2: Configure the interrupt control register reg_trx_done_int_en.

17.4.2 Device Register Operation Flow

- Step 1: Configure the transmission data length reg_trx_cmd_cont_size and reg_trx_data_size.
- Step 2: Configure the device instructions and their related contents reg_trx_cmd_id, reg_trx_cmd_cont0 and reg_trx_cmd_cont1.
- Step 3: Configure the reg_trx_start register issuing operation.
- Step 4: reg_trx_done_int is detected, indicating that the operation is completed.

17.4.3 Erase Operation Process

For flash operations, erasing must be performed before programming operations, and WREN operations must be completed before erasing operations.

- Step 1: Configure the transmission data length reg_trx_cmd_cont_size.
- Step 2: Configure the device instructions and their related contents reg_trx_cmd_id, reg_trx_cmd_cont0.
- Step 3: Configure the reg_trx_start register issuing operation.
- Step 4: reg_trx_done_int is detected, indicating that the operation is completed.

17.4.4 Built-in DMA Read Operation Process

- Step 1: Configure system DMA registers, refer to *DMAC Resisters*.
- Step 2: Configure the transmission data length reg_trx_cmd_cont_size and reg_trx_data_size.
- Step 3: Configure the device instructions and their related contents reg_trx_cmd_id, reg_trx_cmd_cont0 and reg_trx_cmd_cont1.
- Step 4: Configure the reg_trx_start register to issue operations.
- Step 5: reg_trx_done_int is detected, indicating that the device content has been read and written to the buffer.

17.4.5 Built-in DMA Write Operation Process

- Step 1: Configure system DMA registers, refer to *DMAC Resisters*.
- Step 2: Configure the transmission data length reg_trx_cmd_cont_size and reg_trx_data_size.
- Step 3: Configure the device instructions and their related contents reg_trx_cmd_id, reg_trx_cmd_cont0 and reg_trx_cmd_cont1.
- Step 4: Configure the eg_trx_start register issuing operation.
- Step 5: reg_trx_done_int is detected, indicating that the buffer content has been written to the device cache.

17.4.6 Other things to note

- Some SPI NAND Flash devices require a RESET operation of the device before use or after an abnormal reset; therefore, for device compatibility considerations, the first transmission command after starting use or an abnormal reset must be RESET.
- Do not change the relevant register configuration before the device operation is completed, otherwise it may cause abnormal operation.

17.5 Data Structure (NAND Flash/SPI NAND Flash)

17.5.1 2KB page_size

For a 2KB page_size configuration, the common size of the redundant area available to the software is 64Byte. The structure of data in Buffer and Flash is as follows. The size of the redundant area is related to the actual device used.

Table 17.1: The structure of data in Buffer and Flash (2KB page_size)

User Data	Data(2048)	OOB(64)

17.5.2 4KB page_size

For a 4KB page_size configuration, the common size of the redundant area available to the software is 256Byte. The structure of data in Buffer and Flash is as follows. The size of the redundant area is related to the actual device used.

Table 17.2: The structure of data in Buffer and Flash (4KB page_size)

User Data	Data(4096)	OOB(256)

17.6 Register Overview

Table 17.3: SPI NANDFLASH Registers Overview

Name	Address Offset	Description
reg_ctrl	0x000	transmission control
reg_timing_ctrl	0x004	timing control
reg_trx_size	0x008	number of content size
reg_int_en	0x010	interrupt enable
reg_int_clr	0x014	interrupt clear
reg_int_sts	0x01c	interrupt status
reg_cont0	0x030	content 0
reg_cont1	0x034	content 1
reg_cmplt_cnt	0x058	number of transferred bytes
reg_tx_data	0x060	tx data
reg_rx_data	0x064	rx data

17.7 Register Description

17.7.1 reg_ctrl

Table 17.4: reg_ctrl, Offset Address: 0x000

Bits	Name	Access	Description	Reset
0	reg_trx_start	W1T	trigger spi transmission start	
31:1	Reserved			

17.7.2 reg_timing_ctrl

Table 17.5: reg_timing_ctrl, Offset Address: 0x004

Bits	Name	Access	Description	Reset
1:0	reg_trx_time_start	R/W	time for cs assert to 1st command bit unit: sck period	0x0
3:2	Reserved			
7:4	reg_trx_time_end	R/W	time for last data bit to cs de-assert unit: sck period	0x0
15:8	Reserved			
19:16	reg_trx_sck_h	R/W	time for sck high unit: source clock period	0x0
23:20	reg_trx_sck_l	R/W	time for sck low unit: source clock period	0x1
31:24	Reserved			

17.7.3 reg_trx_size

Table 17.6: reg_trx_size, Offset Address: 0x008

Bits	Name	Access	Description	Reset
2:0	reg_trx_cmd_cont_size	R/W	numbers of command content byte	0x0
3	Reserved			
5:4	reg_trx_dummy_size	R/W	numbers of dummy byte	0x0
15:6	Reserved			
28:16	reg_trx_data_size	R/W	numbers of data byte	0x0
31:29	Reserved			

17.7.4 reg_int_en

Table 17.7: reg_int_en, Offset Address: 0x010

Bits	Name	Access	Description	Reset
0	reg_trx_done_int_en	R/W	trx_done interrupt enable	0x1
31:1	Reserved			

17.7.5 reg_int_clr

Table 17.8: reg_int_clr, Offset Address: 0x014

Bits	Name	Access	Description	Reset
0	reg_trx_done_int_clr	W1T	trx_done interrupt clear	
31:1	Reserved			

17.7.6 reg_int_sts

Table 17.9: reg_int_sts, Offset Address: 0x01c

Bits	Name	Access	Description	Reset
0	reg_trx_done_int	RO	trx_done interrpt	
31:1	Reserved			

17.7.7 reg_cont0

Table 17.10: reg_cont0, Offset Address: 0x030

Bits	Name	Access	Description	Reset
7:0	reg_trx_cmd_idx	R/W	spi flash command value	0x0
31:8	reg_trx_cmd_cont0	R/W	spi flash address, or other contents	0x0

17.7.8 reg_cont1

Table 17.11: reg_cont1, Offset Address: 0x034

Bits	Name	Access	Description	Reset
31:0	reg_trx_cmd_cont1	R/W	spi flash address, or other contents	0x0

17.7.9 reg_cmplt_cnt

Table 17.12: reg_cmplt_cnt, Offset Address: 0x058

Bits	Name	Access	Description	Reset
12:0	reg_cmplt_cnt	RO	number of transferred bytes	
31:13	Reserved			

17.7.10 reg_tx_data

Table 17.13: reg_tx_data, Offset Address: 0x060

Bits	Name	Access	Description	Reset
31:0	reg_tx_data	RO	spi tx data	

17.7.11 reg_rx_data

Table 17.14: reg_rx_data, Offset Address: 0x064

Bits	Name	Access	Description	Reset
31:0	reg_rx_data	RO	spi rx data	

CHAPTER

EIGHTEEN

NETWORK INTERFACE

18.1 Ethernet MAC

18.1.1 Overview

The chip supports 1 Ethernet MAC to receive and send network data.

This Ethernet MAC is equipped with a built-in 10/100Mbps Fast Ethernet Transceiver, which can operate in 10/100Mbps, full-duplex or half-duplex mode.

18.1.2 Function Description

The Ethernet module has the following features:

- Ethernet MAC0 with built-in 10 / 100 Mbps Fast Ethernet Transceiver with built-in Ethernet PHY supports 10/100 Mbit/s rate.
- Support full-duplex or half-duplex working mode.
- Support CRC check for input frames.
- Support adding CRC check to output frames.
- Support short frame filling function.
- Support inner loopback in port full-duplex mode.
- Support statistical counting of received and sent frames.
- Support sending and receiving packet caching.
- Support COE (Checksum Offload Engine) checksum offload engine function.

18.1.3 Overall Data Flow

The overall data flow of an ethernet switched interface is shown in the diagram *eMAC overall data flow*.

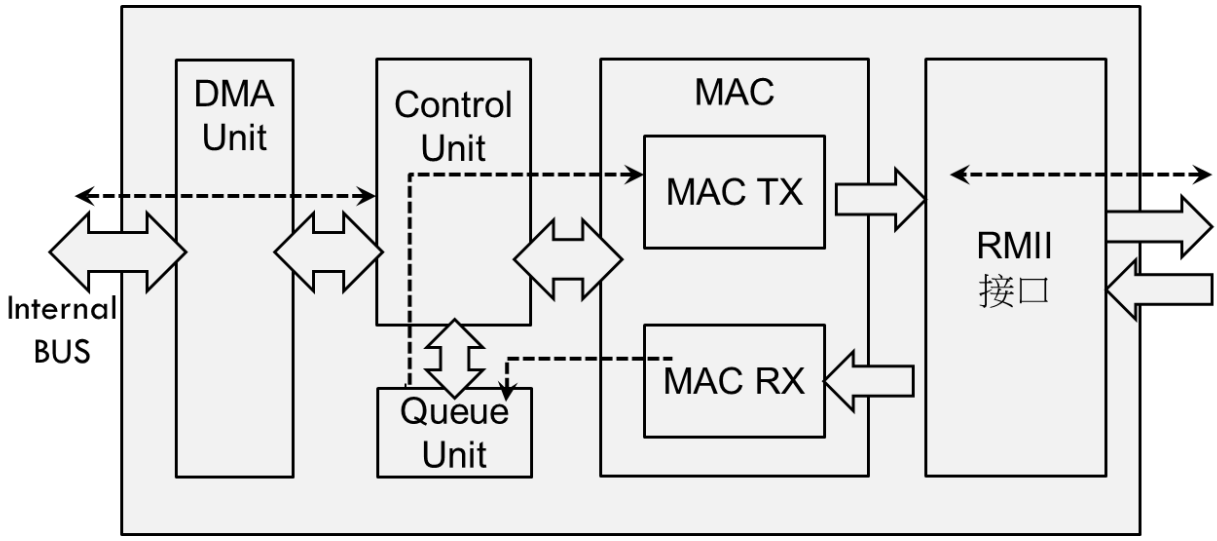


Diagram 18.1: eMAC overall data flow

18.1.4 Single Network Port Function Configuration Description

18.1.4.1 Ethernet Transceiver Frame Management Function

The CPU first configures the Ethernet MAC to receive and send the Descriptor List buffer area and the Descriptor List content, for example, the settings of the sending and receiving frame addresses and packet type and size parameters.

When receiving, the Ethernet MAC receives various received data packets, and receives Descriptor List information according to the CPU configuration, such as packet cache information, including packet cache starting address, packet cache depth, etc., and stores the received packets in the DDR. Then notify the CPU to perform subsequent processing actions.

When sending, the Ethernet MAC sends the packet cache information of the Descriptor List according to the CPU configuration, such as the packet cache starting address, packet length and other packet information, etc., moves the packets stored in the DDR, assembles them into packets by itself, and then sends to the network interface. Then notify the CPU that the packet has been transmitted.

18.1.5 Ethernet Packet Receiving Interrupt Management Function

18.1.5.1 Interrupt Generation

Set the receive direction interrupt, configure Re_Int_Enable bit[6] = 1, and the CPU queries the receive interrupt status Reg_Int_Status bit[6].

18.1.5.2 Interrupt Clear

CPU query receive interrupt status Reg_Int_Status bit[6], write 1 to clear the interrupt status.

18.1.6 Configure PHY Chip Working Status

Ethernet MAC provides MDIO interface settings for the PHY chip. The MDIO interface is divided into read operations and write operations. The registers that mainly control the MDIO interface are Reg_MdioAddr and Reg_MdioData.

- The configuration steps for read operations are as follows:
  - Configure the following settings in the MDIO control register:
    - Reg_MdioAddr bit[15:11] sets the PHY chip address. Please plan according to the PHY chip or board end.
    - Reg_MdioAddr bit[10:6] sets the PHY internal register address to be read and written.
    - Reg_MdioAddr bit[1] is written to 0 (read action command).
  - Finally, set Reg_MdioAddr bit[0] = 1 to start the read action.

  The MDIO interface will receive the read data into Reg_MdioData bit[15:0], and change Reg_Mdi0Addr bit[0] to 0.
- The configuration steps for write operations are as follows:
  - Configure the following settings in the MDIO control register:
    - Reg_MdioAddr bit[15:11] sets the PHY chip address. Please plan according to the PHY chip or board end.
    - Reg_MdioAddr bit[10:6] sets the PHY internal register address to be read and written.
    - Reg_MdioAddr bit[1] is written to 1 (write action command).
  - Finally, set Reg_MdioAddr bit[0] = 1 to start the writing action.

  The MDIO interface will change Reg_Mdi0Addr bit[0] to 0 after the writing action is completed.

18.1.7 Working Mode Switch

Ethernet MAC working mode: Ethernet MAC0 supports built-in EPHY function. The working mode adopted is RMII (10/100M).

The speed and mode switching register settings are as follows:

- Configure ETH0 Reg_MacConfig bit[14] = (100M:1, 10M:0);

Note: This configuration cannot be performed when the chip is working normally. It is recommended to configure it during initialization.

18.1.8 Typical Application

18.1.9 Register Offset Address Description

Ethernet MAC 0 register offset address space:

- ETH0_MAC : 0x0451_000~0x0451_FFFF

18.1.10 GMAC Register Overview

Table 18.1: GMAC Registers Overview

Name	Address Offset	Description
Reg_MacConfig	0x000	Local MAC Operation Configuration Register
Reg_MdioAddr	0x010	MDIO Operation Register
Reg_MdioData	0x014	MDIO Data Read and Write Register
Reg_MacAddr0_High	0x040	Local MAC Address Register#0 High 16bit
Reg_MacAddr0_Low	0x044	Local MAC Address Register#0 Low 32bit
Reg_MacAddr1_High	0x048	Local MAC Address Register#1 High 16bit
Reg_MacAddr1_Low	0x04c	Local MAC Address Register#1 Low 32bit
Reg_Tx_Packet_Num_Good_Bad	0x118	Tranmit Good and Bad Packet Num Count Register
Reg_Tx_Bcast_Packets_Good	0x11c	Tranmit Good Bad Packet Broadcast Count Register
Reg_Tx_Mcast_Packets_Good	0x120	Tranmit Good Bad Packet Multicast Count Register
Reg_Tx_Ucast_Packets_Good_Bad	0x13c	Tranmit Good and Bad Packet Unicast Count Register
Reg_Tx_Mcast_Packets_Good_Bad	0x140	Tranmit Good and Bad Packet Multicast Count Register
Reg_Tx_Bcast_Packets_Good_Bad	0x144	Tranmit Good and Bad Packet Broadcast Count Register
Reg_Rx_Packets_Num_Good_Bad	0x180	Receive Good and Bad Packet Count Register
Reg_Rx_Bcast_Packets_Good	0x18c	Receive Good Packet Broadcast Count Register
Reg_Rx_Mcast_Packets_Good	0x190	Receive Good Packet Multicast Count Register
Reg_Rx_CRC_Error_Packets	0x194	Receive CRC Error Packet Count Register
Reg_Rx_Ucast_Packets_Good	0x1c4	Receive Good Packet Unicast Count Register
Reg_Int_Enable	0x101c	Interrupt Enable Register
Reg_Int_Status	0x1014	Interrupt Status Register

18.1.11 GMAC Register Description

18.1.11.1 Reg_MacConfig

Table 18.2: Reg_MacConfig, Offset Address: 0x000

Bits	Name	Access	Description	Reset
1:0	Reserved			
2	RX_EN	R/W	MAC Receive Enable Register	0x0
3	TX_EN	R/W	MAC Transmit Enable Register	0x0
6:4	Reserved			
7	APCS_EN	R/W	Automatic Pad/CRC Strip Control Enable Register	0x0
9:8	Reserved			
10	CHKS_EN	R/W	IP checksum Auxiliary Enable Register	0x0
11	DUPLEX_MODE	R/W	Full/Half Duplex Mode Register(1enables full duplex mode)	0x0
12	LPBK_MODE	R/W	Loopback Mode Control Register	0x0
13	Reserved			
14	SPEED_MODE	R/W	Speed Mode Register 1'b1:100M, 1'b0:10M	0x0
16:15	Reserved			
19:17	IPG_VAL	R/W	Inter-packet Gap Value Control Register	0x0
22:20	Reserved			
23	WD_DISABLE	R/W	Watchdog Disable Register	0x0
24	Reserved			
25	CRC_STRIP_EN	R/W	CRC Strip Type Control Enable Register	0x0
31:26	Reserved			

18.1.11.2 Reg_MdioAddr

Table 18.3: Reg_MdioAddr, Offset Address: 0x010

Bits	Name	Access	Description	Reset
0	GO	R/W	MDIO Operation Completion Indicator 1: Start operation 0: Operation completed	0x0
1	CMD	R/W	MDIO Operation Command Type Write (1'b1), Read (1'b0)	0x0
5:2	Reserved			
10:6	RegAddr	R/W	External PHY Address Configuration Register	0x0
15:11	PhyAddr	R/W	PHY Device Internal Address Register	0x0
31:16	Reserved			

18.1.11.3 Reg_MdioData

Table 18.4: Reg_MdioData, Offset Address: 0x014

Bits	Name	Access	Description	Reset
15:0	MdioData	R/W	MDIO Data Register Write or Read Data from PHY Register	0x0
31:16	Reserved			

18.1.11.4 Reg_MacAddr0_High

Table 18.5: Reg_MacAddr0_High, Offset Address: 0x040

Bits	Name	Access	Description	Reset
15:0	Addr0_High	R/W	MAC Address Register#0 bit[47:32]	0x0
30:16	Reserved			
31	Addr0_EN	R/W	Addr0 Enable	0x0

18.1.11.5 Reg_MacAddr0_Low

Table 18.6: Reg_MacAddr0_Low, Offset Address: 0x044

Bits	Name	Access	Description	Reset
31:0	Addr0_Low	R/W	MAC Address Register#0 bit[31:0]	0x0

18.1.11.6 Reg_MacAddr1_High

Table 18.7: Reg_MacAddr1_High, Offset Address: 0x048

Bits	Name	Access	Description	Reset
15:0	Addr1_High	R/W	MAC Address Register#1 bit[47:32]	0x0
23:16	Reserved			
29:24	Addr1_MASK	R/W	Addr1 Mask Byte	0x0
30	Addr1_TYPE	R/W	1'b1: compare SA 1'b0: compare DA	0x0
31	Addr1_EN	R/W	Addr1 Enable	0x0

18.1.11.7 Reg_MacAddr1_Low

Table 18.8: Reg_MacAddr1_Low, Offset Address: 0x04c

Bits	Name	Access	Description	Reset
31:0	Addr1_Low	R/W	MAC Address Register#1 bit[31:0]	0x0

18.1.11.8 Reg_Tx_Packet_Num_Good_Bad

Table 18.9: Reg_Tx_Packet_Num_Good_Bad, Offset Address: 0x118

Bits	Name	Access	Description	Reset
31:0	TxPktNumGB	RO	Tranmit Good and Bad Packet Count Register	

18.1.11.9 Reg_Tx_Bcast_Packets_Good

Table 18.10: Reg_Tx_Bcast_Packets_Good, Offset Address: 0x11c

Bits	Name	Access	Description	Reset
31:0	TxBcG	RO	Tranmit Good and Bad Packet Unicast Count Register	

18.1.11.10 Reg_Tx_Mcast_Packets_Good

Table 18.11: Reg_Tx_Mcast_Packets_Good, Offset Address: 0x120

Bits	Name	Access	Description	Reset
31:0	TxMcG	RO	Tranmit Good and Bad Packet Multicast Count Register	

18.1.11.11 Reg_Tx_Ucast_Packets_Good_Bad

Table 18.12: Reg_Tx_Ucast_Packets_Good_Bad, Offset Address: 0x13c

Bits	Name	Access	Description	Reset
31:0	TxUcGB	RO	Tranmit Good and Bad Packet Unicast Count Register	

18.1.11.12 Reg_Tx_Mcast_Packets_Good_Bad

Table 18.13: Reg_Tx_Mcast_Packets_Good_Bad, Offset Address: 0x140

Bits	Name	Access	Description	Reset
31:0	TxMcGB	RO	Tranmit Good and Bad Packet Multicast Count Register	

18.1.11.13 Reg_Tx_Bcast_Packets_Good_Bad

Table 18.14: Reg_Tx_Bcast_Packets_Good_Bad, Offset Address: 0x144

Bits	Name	Access	Description	Reset
31:0	TxBcGB	RO	Tranmit Good and Bad Packet Broadcast Count Register	

18.1.11.14 Reg_Rx_Packets_Num_Good_Bad

Table 18.15: Reg_Rx_Packets_Num_Good_Bad, Offset Address: 0x180

Bits	Name	Access	Description	Reset
31:0	RxPktGB	RO	Receive Good and Bad Packet Count Register	

18.1.11.15 Reg_Rx_Bcast_Packets_Good

Table 18.16: Reg_Rx_Bcast_Packets_Good, Offset Address: 0x18c

Bits	Name	Access	Description	Reset
31:0	RxBcG	RO	Receive Good Packet Broadcast Count Register	

18.1.11.16 Reg_Rx_Mcast_Packets_Good

Table 18.17: Reg_Rx_Mcast_Packets_Good, Offset Address: 0x190

Bits	Name	Access	Description	Reset
31:0	RxMcG	RO	Receive Good Packet Multicast Count Register	

18.1.11.17 Reg_Rx_CRC_Error_Packets

Table 18.18: Reg_Rx_CRC_Error_Packets, Offset Address: 0x194

Bits	Name	Access	Description	Reset
31:0	RxCrcERR	RO	Receive CRC Error Packet Count Register	

18.1.11.18 Reg_Rx_Ucast_Packets_Good

Table 18.19: Reg_Rx_Ucast_Packets_Good, Offset Address: 0x1c4

Bits	Name	Access	Description	Reset
31:0	RxUcG	RO	Receive Good Packet Unicast Count Register	

18.1.11.19 Reg_Int_Enable

Table 18.20: Reg_Int_Enable, Offset Address: 0x101c

Bits	Name	Access	Description	Reset
0	TxInt_EN0	R/W	Transmit Interrupt Enable Register	0x0
5:1	Reserved			
6	RxInt_EN0	R/W	Receive Interrupt Enable Register	0x0
31:7	Reserved			

18.1.11.20 Reg_Int_Status

Table 18.21: Reg_Int_Status, Offset Address: 0x1014

Bits	Name	Access	Description	Reset
0	TxInt_ST0	RO	Transmit Interrupt Status Register	
5:1	Reserved			
6	RxInt_ST0	RO	Receive Interrupt Status Register	
31:7	Reserved			

18.2 EthernetPHY

18.2.1 Overview

The chip provides a built-in Ethernet 10/100 Base-TX compliant PHY interface.

18.2.2 Function Description

- Supports IEEE 802.3 10/100 Base-TX compliant.
- Supports automatic negotiation of transmission speed and full-half-duplex working mode.
- Supports the automatic movement and flip function of Ethernet line crossover lines or direct lines.
- Supports Ethernet WOL (Wake on Lan) function.

18.2.3 Function Configuration Diagram Description

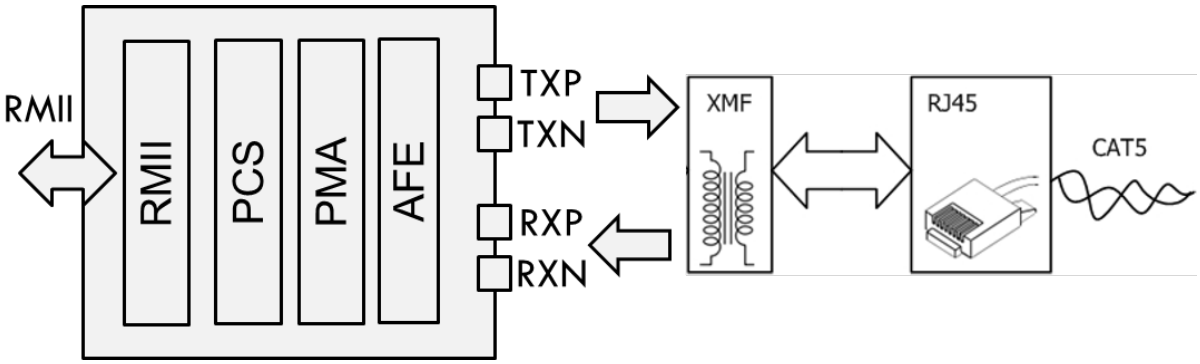


Diagram 18.2: Built-in 10/100 Ethernet PHY block diagram

The 10/100Mbps transmission and reception functions are on standard category 5 (CAT5) twisted pair cables, and the transmission and reception signals are connected to the RJ45 standard interface through a transformer.

CHAPTER

NINETEEN

VIDEO INTERFACE

19.1 VI

19.1.1 Overview

The Video Input (VI) unit is a module on chip which is responsile receiving camera video data. VI can support MIPI Rx (including MIPI, Sub-LVDS, HiSPi) interface or BT.656, BT.601, BT.1120 interface and DC (Digital Camera) to receive video data, to processe it and send it to the next-level video processing module (ISP). The functional block diagram of a VI is shown in the diagram *VI functional block diagram*.

VI is divided into two physical sub-modules, consisting of the mobile industry processor interface receiver module MIPI Rx and the video input processing module VI Proc. The MIPI Rx module receives and processes different video data, while the VI Proc module integrates video signals in different formats into a single video signal required by the ISP module and sends it out.

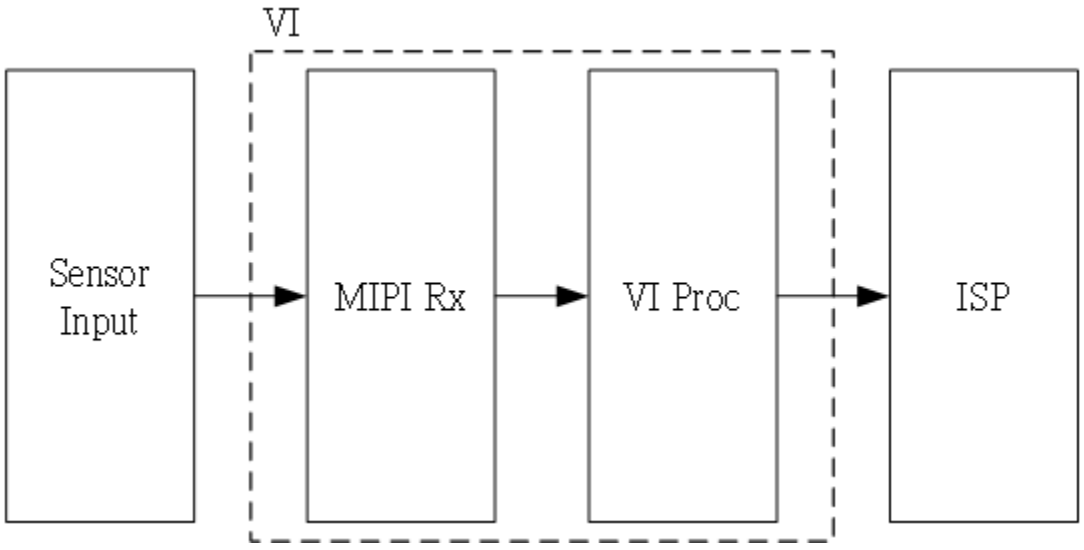


Diagram 19.1: VI functional block diagram

19.1.2 Features

- MIPI Rx supports up to two sensor data inputs at the same time.
  - Single sensor supports up to 5M (2688x1944, 2880x1620) @60fps HDR input or @30fps linear input.
  - Dual-channel sensor supports up to FHD (1920x1080) @60fps HDR or linear input.
- MIPI Rx input supports a maximum data bit width of 12bit.
- Supports BT.656, BT.601, BT.1120 (only progressive mode is supported).
- Supports BT.656/BT.1120 multi-channel fusion input (1,2,4 channels, only supports progressive mode).

DC interface

- Support MIPI CSI-2, Sub-LVDS, HiSPi interface.
- Support MIPI interface input YUV422 format.
- Supports two-frame high dynamic range (HDR) image input.

19.1.3 Mode Function Description

19.1.3.1 Typical Application

VI can support multiple timing inputs and different interfaces, and perform video input collection for different encoding methods. The system can use registers to configure different functional modes to adapt to different video interfaces.

The VI module can support up to three inputs. Typical inputs are as follows:

- 1 channel 5M (2688x1944, 2880x1620) @60fps HDR input or linear @30fps input.
- 2 channels FHD (1920x1080) @60fps HDR or line input + 1 channel BT input (BT.656, BT.601 or BT.1120).
- 1 channel 5M (2688x1944, 2880x1620) @60fps HDR or @30fps linear input + 1 channel BT input.

19.1.3.2 Functional Principle

19.1.3.3 BT.1120 Interface Timing

VI supports Y/C separated input BT.1120 interface timing. Before transmitting video signals, a synchronization code will be transmitted. The synchronization code uses special bytes SAV and EAV in the data stream to represent the start and end of valid row data respectively. After the synchronization code, 16 bits are used to transmit the video signal, of which 8 bits are used to transmit brightness and the other 8 bits are used to transmit chroma, as shown in the chart *BT.1120 horizontal interface timing* and the chart *BT.1120 vertical interface timing*.

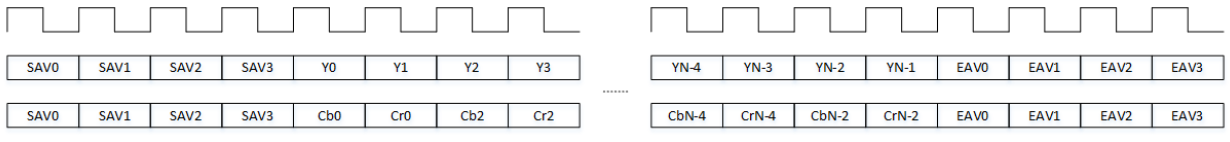


Diagram 19.2: BT.1120 horizontal interface timing

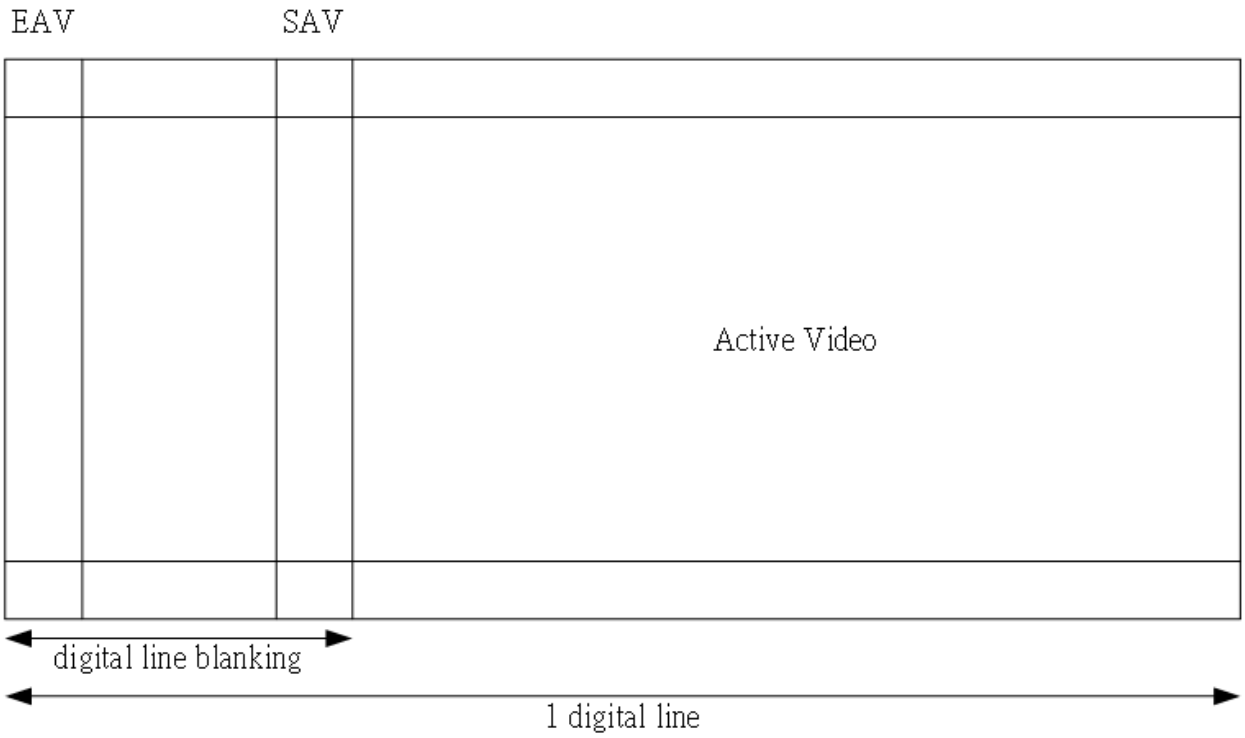


Diagram 19.3: BT.1120 vertical interface timing

The synchronization code format is a combination of 4 bytes, in order {0xFF, 0x00, 0x00, EAV/SAV}. The fourth byte is detailed as follows. SG2002 only supports progressive progressive scan format (Progressive). Therefore, the value of bit 6 is 0.

Table 19.1: Sync code format

	SAV/EAV bit			Protection Bit				
7 (Fixed)	6 (F)	5 (V)	4 (H)	3 (P3)	2 (P2)	1 (P1)	0 (P0)	Comment
1	0	0	0	0	0	0	0	SAV_VLD
1	0	0	1	1	1	0	1	EAV_VLD
1	0	1	0	1	0	1	1	SAV_BLK
1	0	1	1	0	1	1	0	EAV_BLK

SAV_VLD: Valid line area, line synchronization signal ends, and valid pixels begin.

EVA_VLD: Effective line area, line synchronization signal starts, and effective pixel ends.

SAV_BLK: Blanking line area, end of line synchronization signal.

EAV_BLK: Blanking row area, horizontal sync signal starts.

19.1.3.4 BT.656 Interface Timing

VI also supports Y/C combined input BT.656 interface timing. During transmission, it also uses synchronization codes SAV and EAV to indicate the start and end of valid line data, but only uses 8 bits to transmit video signals, and uses time sharing to transmit brightness and Chromaticity, as shown in diagram *BT.656 horizontal interface timing*.

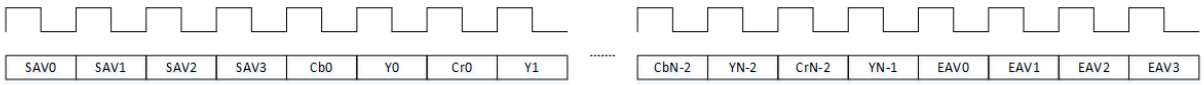


Diagram 19.4: BT.656 horizontal interface timing

The only difference between BT.656 and BT.1120 is that 16 bit (BT.1120) and 8 bit (BT.656) are used for image transmission. The rest of the vertical timing and synchronization code formats are the same.

19.1.3.5 BT.601 Interface Timing

In addition to utilizing synchronization codes BT.1120 and BT.656, VI supports BT.601 interface timing utilizing a variety of different synchronization signals for transmission. The actual video data can be set to the 16-bit mode of Y/C separate input or the 8-bit mode of Y/C combined time-sharing input using the register. The synchronization mode can be selected by the register to be vhs, vde, or vsde. The detailed timing is as shown below.

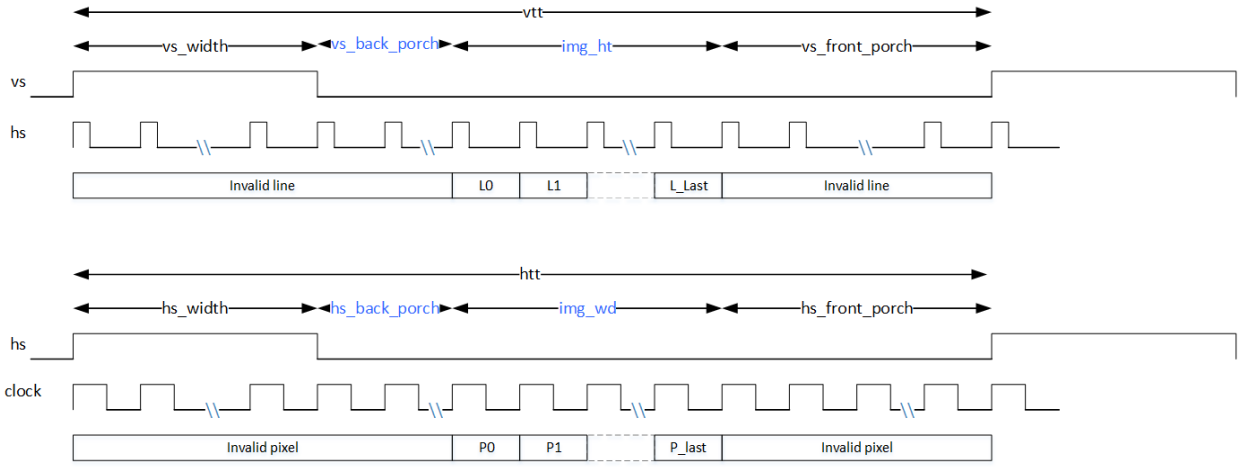


Diagram 19.5: BT.601 vhs synchronous mode

The input synchronization signal in vhs mode is frame synchronization signal (vs), horizontal synchronization signal (hs), the system must set the number of blanking lines after the frame (vs_back_porch), the image height (img_ht), the number of blanking pixels after the line (hs_back_porch) and Image width (img_wd).

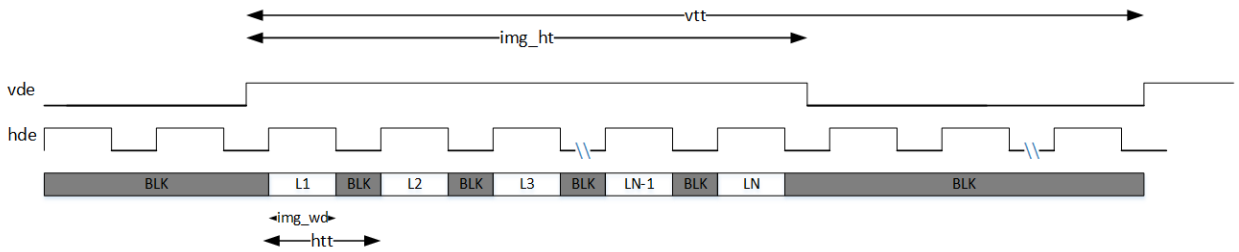


Diagram 19.6: BT.601 vde synchronous mode

The vde mode synchronization signals are the row valid signal (vde) and the row valid signal (hde). In this mode, the system does not need to set parameters related to timing and phase sequence. The VI module will receive data based on the hde/vde signal and perform frame updates based on the vde signal.

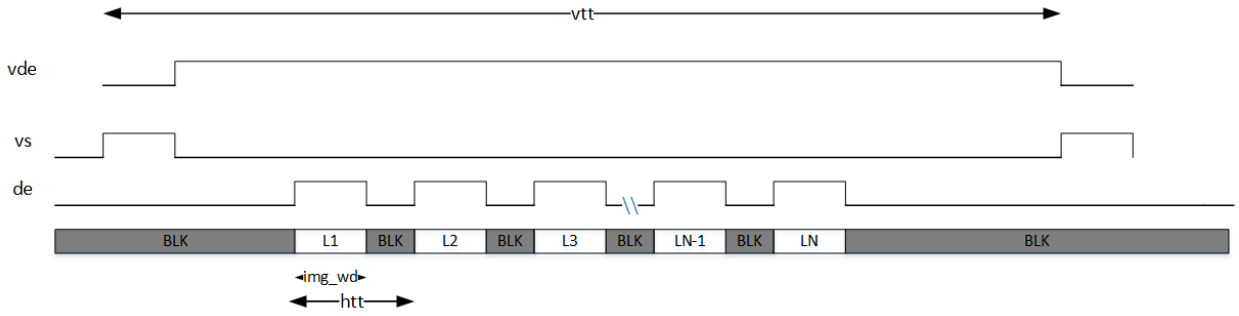


Diagram 19.7: BT.601 vsde synchronous mode

The vsde mode sync signals are the frame sync signal (vs) and the valid pixel flag (de). In this mode, the system does not need to set parameters related to timing and phase sequence. The VI module will receive data based on the de signal and perform frame updates based on the vs signal.

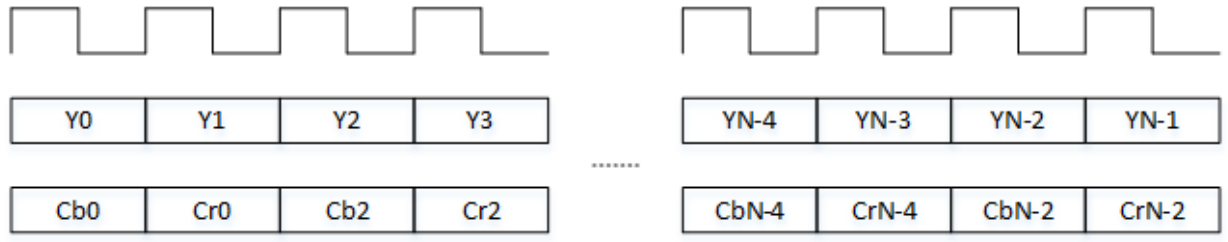


Diagram 19.8: BT.601 Y/C separated 16bit mode

Diagram 19.9: BT.601 Y/C combined 8bit mode

19.1.3.6 Digital Camera (DC) Interface Timing

VI supports digital camera (DC) interface timing that transmits RAW format and simulates BT transmission. The DC interface can support four different modes: 8bit, 10bit, 12bit, and 16bit. Register settings can also be used to receive synchronization codes or similar to BT. 601's three different synchronization modes to receive video signals.

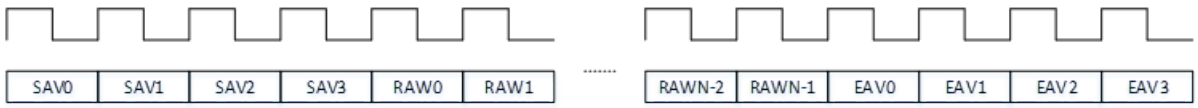


Diagram 19.10: DC sync code mode

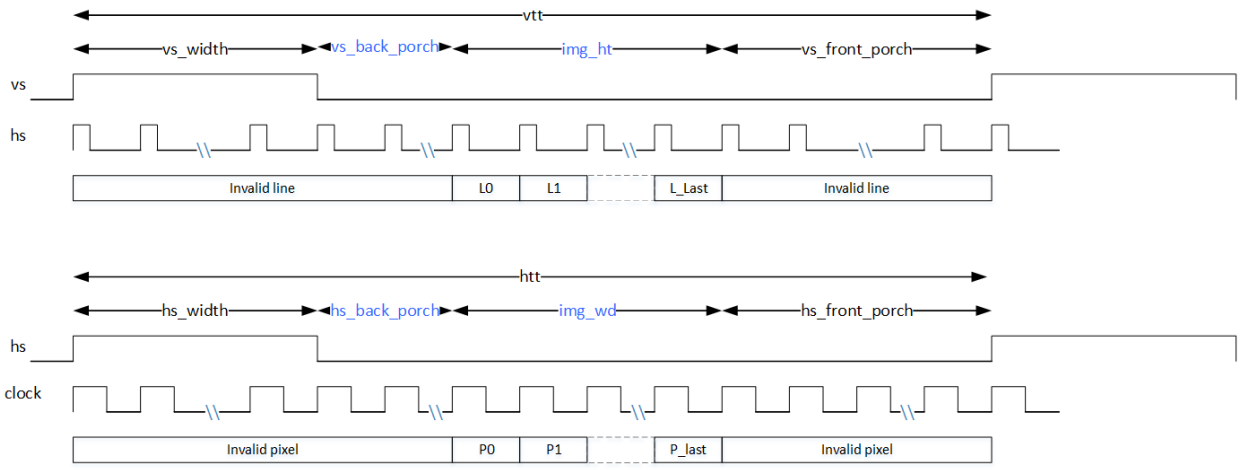


Diagram 19.11: DC sync signal mode -vhs mode

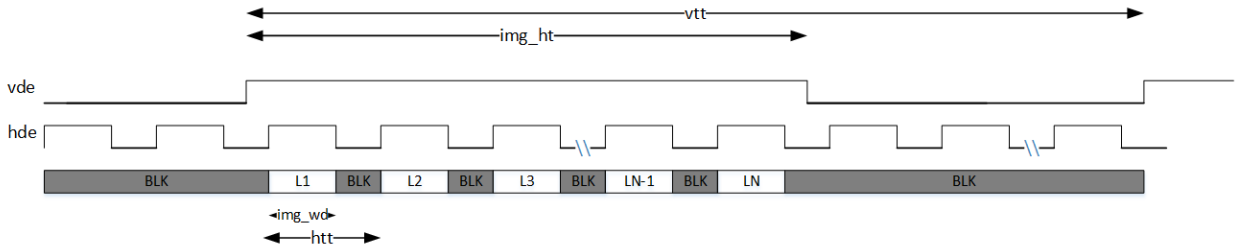


Diagram 19.12: DC sync signal mode -VDE mode

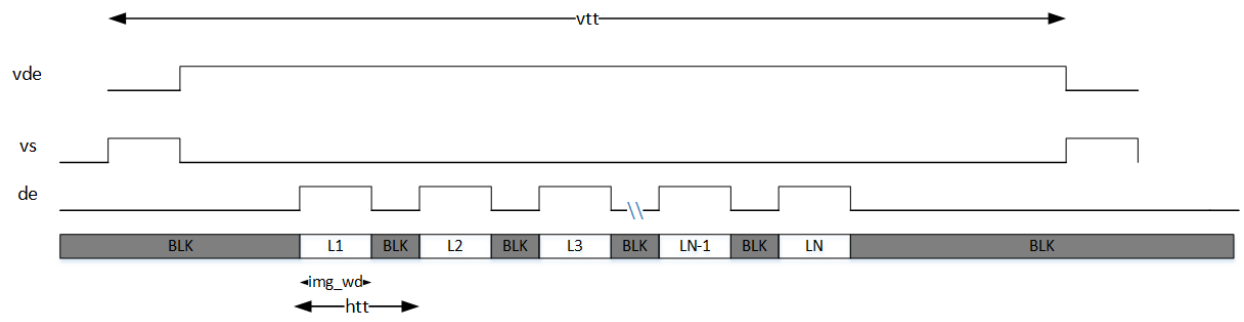


Diagram 19.13: DC sync signal mode -VSDE mode

19.1.4 Image Storage Mode

Images stored in DRAM are divided into two formats: Bayer 12bit and YCbCr 8bit. Among them, Y/Cb/Cr are stored separately in three different DRAM locations. The arrangement of the two (12bit/8bit) format images in DRAM is as shown in the figure below.

Bayer 12bit		Pixel 0	Pixel 1	Pixel 2	Pixel 3				
		Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
DRAM Address	Addr N+0	4	5	6	7	8	9	10	11
	Addr N+1	4	5	6	7	8	9	10	11
	Addr N+2	0	1	2	3	0	1	2	3
	Addr N+3	4	5	6	7	8	9	10	11
	Addr N+4	4	5	6	7	8	9	10	11
	Addr N+5	0	1	2	3	0	1	2	3

Diagram 19.14: Bayer 12 bit image storage method

YCbCr 8bit		Pixel 0	Pixel 1	Pixel 2	Pixel 3				
		Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
DRAM Address	Addr N+0	0	1	2	3	4	5	6	7
	Addr N+1	0	1	2	3	4	5	6	7
	Addr N+2	0	1	2	3	4	5	6	7
	Addr N+3	0	1	2	3	4	5	6	7

Diagram 19.15: YCbCr 8bit image storage method

19.1.5 VI Register Overview

There are two identical sets of VI modules in the chip, and the internal register offset addresses are the same, and the base addresses are 0x0A0C2000 and 0x0A0C4000 respectively. There is also a set of VI modules that only support the BT interface, with the base address 0x0A0C6000.

Table 19.2: VI Registers Overview

Name	Address Offset	Description
REG_00	0x000	MODE
REG_10	0x010	TTL_MODE_0
REG_14	0x014	TTL_MODE_1
REG_18	0x018	TTL_MODE_2
REG_1C	0x01c	TTL_MODE_3
REG_20	0x020	TTL_MODE_4
REG_24	0x024	TTL_MODE_5
REG_28	0x028	TTL_MODE_6
REG_30	0x030	TTL_MODE_7
REG_40	0x040	HDR_MODE_0
REG_44	0x044	HDR_MODE_1
REG_48	0x048	HDR_MODE_2
REG_50	0x050	BLC_MODE
REG_54	0x054	BLC_MODE_0
REG_58	0x058	BLC_MODE_1
REG_60	0x060	VI_PINMUX_0
REG_64	0x064	VI_PINMUX_1
REG_68	0x068	VI_PINMUX_2
REG_6C	0x06c	VI_PINMUX_3
REG_70	0x070	VI_PINMUX_4
REG_74	0x074	VI_PINMUX_5
REG_80	0x080	BT_PATH_0
REG_88	0x088	BT_PATH_2
REG_8C	0x08c	BT_PATH_3
REG_90	0x090	BT_PATH_4
REG_94	0x094	BT_PATH_5
REG_98	0x098	BT_PATH_6
REG_9C	0x09c	BT_PATH_7
REG_A0	0x0a0	BT_PATH_8
REG_A4	0x0a4	BT_PATH_A
REG_B0	0x0b0	CROP_0
REG_B4	0x0b4	CROP_1
REG_D0	0x0d0	MODE_CTRL
REG_D4	0x0d4	SYNC_CODE_0
REG_D8	0x0d8	SYNC_CODE_1
REG_DC	0x0dc	SYNC_CODE_2
REG_E0	0x0e0	SYNC_CODE_3
REG_E4	0x0e4	SYNC_CODE_4
REG_E8	0x0e8	SYNC_CODE_5
REG_EC	0x0ec	SYNC_CODE_6
REG_F0	0x0f0	SYNC_CODE_7
REG_F4	0x0f4	SYNC_CODE_8
REG_F8	0x0f8	SYNC_CODE_9

continues on next page

Table 19.2 – continued from previous page

Name	Address Offset	Description
REG_FC	0x0fc	VS_GEN
REG_100	0x100	SYNC_CODE_A
REG_104	0x104	SYNC_CODE_B
REG_108	0x108	HDR_PATTEN_2
REG_110	0x110	HISPI_MODE_CTRL_0
REG_114	0x114	HISPI_MODE_CTRL_1
REG_118	0x118	HISPI_MODE_CTRL_2
REG_11C	0x11c	HISPI_MODE_CTRL_3
REG_120	0x120	HISPI_MODE_CTRL_4
REG_124	0x124	HISPI_MODE_CTRL_5

19.1.6 VI Register Description

19.1.6.1 REG_00

Table 19.3: REG_00, Offset Address: 0x000

Bits	Name	Access	Description	Reset
2:0	reg_sensor_mac_mode	R/W	Sensor mode 3'b000: Disable 3'b001: CSI 3'b010: Sub-LVDS 3'b011: TTL	0x0
3	reg_bt_demux_enable	R/W	BT Demux enable	0x0
4	reg_csi_ctrl_enable	R/W	CSI controller enable	0x0
5	reg_csi_vs_inv	R/W	CSI VS inverse	0x1
6	reg_csi_hs_inv	R/W	CSI HS inverse	0x1
7	Reserved			
8	reg_sublvds_ctrl_enable	R/W	Sub-LVDS controller enable	0x0
9	reg_sublvds_vs_inv	R/W	Sub-LVDS VS inverse	0x1
10	reg_sublvds_hs_inv	R/W	Sub-LVDS HS inverse	0x1
11	reg_sublvds_hdr_inv	R/W	Sub-LVDS HDR inverse	0x1
31:12	Reserved			

19.1.6.2 REG_10

Table 19.4: REG_10, Offset Address: 0x010

Bits	Name	Access	Description	Reset
0	reg_ttl_ip_en	R/W	TTL enable	0x0
2:1	reg_ttl_sensor_bit	R/W	TTL bit mode 2'b00: 8-bit 2'b01: 10-bit 2'b10: 12-bit 2'b11: 16-bit	0x0
3	Reserved			
5:4	reg_ttl_bt_fmt_out	R/W	TTL BT output format 2'b00: {Cb,Y},{Cr,Y} 2'b01: {Cr,Y},{Cb,Y} 2'b10: {Y,Cb},{Y,Cr} 2'b11: {Y,Cr},{Y,Cb}	0x2
7:6	Reserved			
11:8	reg_ttl_fmt_in	R/W	TTL input format 4'b0000: bt_2x with sync pattern, 9-bit BT656 4'b0001: bt_1x with sync pattern, 17-bit BT1120 4'b0010: bt_2x without sync pattern, 11-bit BT601 (vhs mode) 4'b0011: bt_1x without sync pattern, 19-bit BT601 (vhs mode) 4'b0100: bt_2x without sync pattern, 11-bit BT601 (vde mode) 4'b0101: bt_1x without sync pattern, 19-bit BT601 (vde mode) 4'b0110: bt_2x without sync pattern, 11-bit BT601 (vsde mode) 4'b0111: bt_1x without sync pattern, 19-bit BT601 (vsde mode) 4'b100x: sensor with sync pattern 4'b101x: sensor without sync pattern, use vs + hs (vhs mode) 4'b110x: sensor without sync pattern, use vde + hde (vde mode) 4'b111x: sensor without sync pattern, use vs + hde (vsde mode)	0x0
13:12	reg_ttl_bt_data_seq	R/W	TTL bt data sequence 2'b00: Cb0-Y0-Cr0-Y1 2'b01: Cr0-Y0-Cb0-Y1 2'b10: Y0-Cb0-Y1-Cr0 2'b11: Y0-Cr0-Y1-Cb0	0x0
14	reg_ttl_vs_inv	R/W	TTL vs inverse	0x0
15	reg_ttl_hs_inv	R/W	TTL hs inverse	0x0
31:16	Reserved			

19.1.6.3 REG_14

Table 19.5: REG_14, Offset Address: 0x014

Bits	Name	Access	Description	Reset
11:0	reg_ttl_vs_bp	R/W	TTL vsync back porch setting	0x0
15:12	Reserved			
27:16	reg_ttl_hs_bp	R/W	TTL hsync back porch setting	0x0
31:28	Reserved			

19.1.6.4 REG_18

Table 19.6: REG_18, Offset Address: 0x018

Bits	Name	Access	Description	Reset
11:0	reg_ttl_img_wd	R/W	TTL image width setting	0x0
15:12	Reserved			
27:16	reg_ttl_img_ht	R/W	TTL image height setting	0x0
31:28	Reserved			

19.1.6.5 REG_1C

Table 19.7: REG_1C, Offset Address: 0x01c

Bits	Name	Access	Description	Reset
15:0	reg_ttl_sync_0	R/W	TTL sync code 0	0x0
31:16	reg_ttl_sync_1	R/W	TTL sync code 1	0x0

19.1.6.6 REG_20

Table 19.8: REG_20, Offset Address: 0x020

Bits	Name	Access	Description	Reset
15:0	reg_ttl_sync_2	R/W	TTL sync code 2	0x0
31:16	Reserved			

19.1.6.7 REG_24

Table 19.9: REG_24, Offset Address: 0x024

Bits	Name	Access	Description	Reset
15:0	reg_ttl_sav_vld	R/W	TTL valid line SAV	0x0
31:16	reg_ttl_sav_blk	R/W	TTL blanking line SAV	0x0

19.1.6.8 REG_28

Table 19.10: REG_28, Offset Address: 0x028

Bits	Name	Access	Description	Reset
15:0	reg_ttl_eav_vld	R/W	TTL valid line EAV	0x0
31:16	reg_ttl_eav_blk	R/W	TTL blanking line EAV	0x0

19.1.6.9 REG_30

Table 19.11: REG_30, Offset Address: 0x030

Bits	Name	Access	Description	Reset
2:0	reg_vi_sel	R/W	VI input mode select 3'h1: RAW 3'h2: BT601 3'h3: BT656 3'h4: BT1120 else: reserved	0x0
3	reg_vi_from	R/W	VI input from VI0 or VI1 1'b0: from VI0 1'b1: from VI1	0x0
4	reg_vi_clk_inv	R/W	VI clock inverse	0x0
5	reg_vi_v_sel_vs	R/W	1'b1: vs_in signal as vs 1'b0: vs_in signal as vde	0x1
6	reg_vi_vs_dbg	R/W	vsync source select	0x0
7	Reserved			
8	reg_pad_vi0_clk_inv	R/W	vi0 clk inverse	0x0
9	reg_pad_vi1_clk_inv	R/W	vi1 clk inverse	0x0
10	reg_pad_vi2_clk_inv	R/W	vi2 clk inverse	0x0
31:11	Reserved			

19.1.6.10 REG_40

Table 19.12: REG_40, Offset Address: 0x040

Bits	Name	Access	Description	Reset
0	reg_sensor_mac_hdr_en	R/W	Sensor mac hdr manual mode enable Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
1	reg_sensor_mac_hdr_vsinv	R/W	Sensor mac vsync output inverse Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
2	reg_sensor_mac_hdr_hsinv	R/W	Sensor mac hsync output inverse Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
3	reg_sensor_mac_hdr_deinv	R/W	Sensor mac de output inverse Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
4	reg_sensor_mac_hdr_hdr0inv	R/W	Sensor mac hdr[0] output inverse Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
5	reg_sensor_mac_hdr_hdr1inv	R/W	Sensor mac hdr[1] output inverse Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
6	reg_sensor_mac_hdr_blcinv	R/W	Sensor mac blc output inverse Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
7	Reserved			
8	reg_sensor_mac_hdr_mode	R/W	Sensor mac hdr mode 1'b1 stands for HiSPi S-SP HDR mode, remove HDR blanking line Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
31:9	Reserved			

19.1.6.11 REG_44

Table 19.13: DMACREG_44_IDREG, Offset Address: 0x044

Bits	Name	Access	Description	Reset
12:0	reg_sensor_mac_hdr_shift	R/W	Sensor mac hdr long exposure shift (long exposure lines before 1st short exposure line) Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
15:13	Reserved			
28:16	reg_sensor_mac_hdr_vsize	R/W	Sensor mac hdr vsize Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
31:29	Reserved			

19.1.6.12 REG_48

Table 19.14: REG_48, Offset Address: 0x048

Bits	Name	Access	Description	Reset
12:0	reg_sensor_mac_info_line_num	R/W	Info line number Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x1
15:13	Reserved			
16	reg_sensor_mac_rm_info_line	R/W	Remove info line Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
31:17	Reserved			

19.1.6.13 REG_50

Table 19.15: REG_50, Offset Address: 0x050

Bits	Name	Access	Description	Reset
0	reg_sensor_mac_blc0_en	R/W	BLC0 mode enable	0x0
1	reg_sensor_mac_blc1_en	R/W	BLC1 mode enable	0x0
31:2	Reserved			

19.1.6.14 REG_54

Table 19.16: REG_54, Offset Address: 0x054

Bits	Name	Access	Description	Reset
12:0	reg_sensor_mac_blc0_start	R/W	BLC0 start line number	0x0
15:13	Reserved			
28:16	reg_sensor_mac_blc0_size	R/W	BLC0 line size	0x4
31:29	Reserved			

19.1.6.15 REG_58

Table 19.17: REG_58, Offset Address: 0x058

Bits	Name	Access	Description	Reset
12:0	reg_sensor_mac_blc1_start	R/W	BLC1 start line number	0x0
15:13	Reserved			
28:16	reg_sensor_mac_blc1_size	R/W	BLC1 line size	0x4
31:29	Reserved			

19.1.6.16 REG_60

Table 19.18: REG_60, Offset Address: 0x060

Bits	Name	Access	Description	Reset
5:0	reg_vi_vs_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
7:6	Reserved			
13:8	reg_vi_hs_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
15:14	Reserved			
21:16	reg_vi_vde_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
23:22	Reserved			
29:24	reg_vi_hde_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
31:30	Reserved			

19.1.6.17 REG_64

Table 19.19: REG_64, Offset Address: 0x064

Bits	Name	Access	Description	Reset
5:0	reg_vi_d0_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
7:6	Reserved			
13:8	reg_vi_d1_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
15:14	Reserved			
21:16	reg_vi_d2_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
23:22	Reserved			
29:24	reg_vi_d3_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
31:30	Reserved			

19.1.6.18 REG_68

Table 19.20: REG_68, Offset Address: 0x068

Bits	Name	Access	Description	Reset
5:0	reg_vi_d4_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
7:6	Reserved			
13:8	reg_vi_d5_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
15:14	Reserved			
21:16	reg_vi_d6_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
23:22	Reserved			
29:24	reg_vi_d7_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
31:30	Reserved			

19.1.6.19 REG_6C

Table 19.21: REG_6C, Offset Address: 0x06c

Bits	Name	Access	Description	Reset
5:0	reg_vi_d8_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
7:6	Reserved			
13:8	reg_vi_d9_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
15:14	Reserved			
21:16	reg_vi_d10_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
23:22	Reserved			
29:24	reg_vi_d11_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
31:30	Reserved			

19.1.6.20 REG_70

Table 19.22: REG_70, Offset Address: 0x070

Bits	Name	Access	Description	Reset
5:0	reg_vi_d12_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
7:6	Reserved			
13:8	reg_vi_d13_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
15:14	Reserved			
21:16	reg_vi_d14_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
23:22	Reserved			
29:24	reg_vi_d15_sel	R/W	vi pin select [5]: from VI1 or VI0 [4:0]: from which VI pad count	0x0
31:30	Reserved			

19.1.6.21 REG_74

Table 19.23: REG_74, Offset Address: 0x074

Bits	Name	Access	Description	Reset
2:0	reg_vi_bt_d0_sel	R/W	vi bt pin select from which VI2 pad count	0x0
3	Reserved			
6:4	reg_vi_bt_d1_sel	R/W	vi bt pin select from which VI2 pad count	0x1
7	Reserved			
10:8	reg_vi_bt_d2_sel	R/W	vi bt pin select from which VI2 pad count	0x2
11	Reserved			
14:12	reg_vi_bt_d3_sel	R/W	vi bt pin select from which VI2 pad count	0x3
15	Reserved			
18:16	reg_vi_bt_d4_sel	R/W	vi bt pin select from which VI2 pad count	0x4
19	Reserved			
22:20	reg_vi_bt_d5_sel	R/W	vi bt pin select from which VI2 pad count	0x5
23	Reserved			
26:24	reg_vi_bt_d6_sel	R/W	vi bt pin select from which VI2 pad count	0x6
27	Reserved			
30:28	reg_vi_bt_d7_sel	R/W	vi bt pin select from which VI2 pad count	0x7
31	Reserved			

19.1.6.22 REG_80

Table 19.24: REG_80, Offset Address: 0x080

Bits	Name	Access	Description	Reset
0	reg_bt_clr_sync_lost_1t	R/W	Clear sync_lost signal	0x0
1	reg_bt_ip_en	R/W	BT path enable	0x0
2	reg_bt_ddr_mode	R/W	BT DDR mode	0x0
3	reg_bt_hs_gate_by_vde	R/W	HS gating by VDE	0x0
4	reg_bt_vs_inv	R/W	vsync inverse	0x0
5	reg_bt_hs_inv	R/W	hsync inverse	0x0
6	reg_bt_vs_as_vde	R/W	input vsync as vde	0x0
7	reg_bt_hs_as_hde	R/W	input hsync as hde	0x0
14:8	reg_bt_sw_en_clk	R/W	Clock gating software enable [0]: delay control clock enable [1]: timing demultiplexer clock enable [2]: timing gen clock enable [3]: rx decode 0 clock enable [4]: rx decode 1 clock enable [5]: rx decode 2 clock enable [6]: rx decode 3 clock enable	0x0
15	Reserved			
17:16	reg_bt_demux_ch	R/W	Demux setting 2'h0: No demux 2'h1: Demux 2 2'h2: Demux 3 2'h3: Demux 4	0x0
19:18	Reserved			
22:20	reg_bt_fmt_sel	R/W	3'b000 : bt_2x with sync pattern, 9-bit BT656 (clock + 8-bit data) 3'b001 : bt_1x with sync pattern, 17-bit BT1120 (clock + 16-bit data) 3'b010 : bt_2x without sync pattern, 11-bit BT601 (clock + 8-bit data + vs + hs) (vhs_mode) 3'b011 : bt_1x without sync pattern, 19-bit BT601 (clock + 16-bit data + vs + hs) (vhs_mode) 3'b100 : bt_2x without sync pattern, 11-bit BT601 (clock + 8-bit data + vde + hde) (vde_mode) 3'b101 : bt_1x without sync pattern, 19-bit BT601 (clock + 16-bit data + vde + hde) (vde_mode) 3'b110 : bt_2x without sync pattern, 11-bit BT601 (clock + 8-bit data + vs + hde) (vsde_mode) 3'b111 : bt_1x without sync pattern, 19-bit BT601 (clock + 16-bit data + vs + hde) (vsde_mode)	0x0
31:23	Reserved			

19.1.6.23 REG_88

Table 19.25: REG_88, Offset Address: 0x088

Bits	Name	Access	Description	Reset
11:0	reg_bt_img_wd_m1	R/W	BT image width	0x0
15:12	Reserved			
27:16	reg_bt_img_ht_m1	R/W	BT image height	0x0
31:28	Reserved			

19.1.6.24 REG_8C

Table 19.26: REG_8C, Offset Address: 0x08c

Bits	Name	Access	Description	Reset
11:0	reg_bt_vs_bp_m1	R/W	BT vsync back porch	0x0
15:12	Reserved			
27:16	reg_bt_hs_bp_m1	R/W	BT hsync back porch	0x0
31:28	Reserved			

19.1.6.25 REG_90

Table 19.27: REG_90, Offset Address: 0x090

Bits	Name	Access	Description	Reset
7:0	reg_bt_vs_fp_m1	R/W	BT vsync front porch	0x0
15:8	reg_bt_hs_fp_m1	R/W	BT hsync front porch	0x0
31:16	Reserved			

19.1.6.26 REG_94

Table 19.28: REG_94, Offset Address: 0x094

Bits	Name	Access	Description	Reset
7:0	reg_bt_sync_0	R/W	BT sync code byte 0	0x0
15:8	reg_bt_sync_1	R/W	BT sync code byte 1	0x0
23:16	reg_bt_sync_2	R/W	BT sync code byte 2	0x0
31:24	Reserved			

19.1.6.27 REG_98

Table 19.29: REG_98, Offset Address: 0x098

Bits	Name	Access	Description	Reset
7:0	reg_bt_sav_vld_0	R/W	BT valid SAV sync code for demux 0	0x0
15:8	reg_bt_sav_blk_0	R/W	BT blank SAV sync code for demux 0	0x0
23:16	reg_bt_eav_vld_0	R/W	BT valid EAV sync code for demux 0	0x0
31:24	reg_bt_eav_blk_0	R/W	BT blank EAV sync code for demux 0	0x0

19.1.6.28 REG_9C

Table 19.30: REG_9C, Offset Address: 0x09c

Bits	Name	Access	Description	Reset
7:0	reg_bt_sav_vld_1	R/W	BT valid SAV sync code for demux 1	0x0
15:8	reg_bt_sav_blk_1	R/W	BT blank SAV sync code for demux 1	0x0
23:16	reg_bt_eav_vld_1	R/W	BT valid EAV sync code for demux 1	0x0
31:24	reg_bt_eav_blk_1	R/W	BT blank EAV sync code for demux 1	0x0

19.1.6.29 REG_A0

Table 19.31: REG_A0, Offset Address: 0x0a0

Bits	Name	Access	Description	Reset
7:0	reg_bt_sav_vld_2	R/W	BT valid SAV sync code for demux 2	0x0
15:8	reg_bt_sav_blk_2	R/W	BT blank SAV sync code for demux 2	0x0
23:16	reg_bt_eav_vld_2	R/W	BT valid EAV sync code for demux 2	0x0
31:24	reg_bt_eav_blk_2	R/W	BT blank EAV sync code for demux 2	0x0

19.1.6.30 REG_A4

Table 19.32: REG_A4, Offset Address: 0x0a4

Bits	Name	Access	Description	Reset
7:0	reg_bt_sav_vld_3	R/W	BT valid SAV sync code for demux 3	0x0
15:8	reg_bt_sav_blk_3	R/W	BT blank SAV sync code for demux 3	0x0
23:16	reg_bt_eav_vld_3	R/W	BT valid EAV sync code for demux 3	0x0
31:24	reg_bt_eav_blk_3	R/W	BT blank EAV sync code for demux 3	0x0

19.1.6.31 REG_B0

Table 19.33: REG_B0, Offset Address: 0x0b0

Bits	Name	Access	Description	Reset
12:0	reg_sensor_mac_crop_start_x	R/W	Pixels before reg_s en-sor_mac_crop_start_x will be cropped in each line if enable r eg_sensor_mac_crop_en.	0xFFF
15:13	Reserved			
28:16	reg_sensor_mac_crop_end_x	R/W	Pixels after reg _sensor_mac_crop_end_x will be cropped in each line if enable r eg_sensor_mac_crop_en.	0xFFF
30:29	Reserved			
31	reg_sensor_mac_crop_en	R/W	enable crop function	0x0

19.1.6.32 REG_B4

Table 19.34: REG_B4, Offset Address: 0x0b4

Bits	Name	Access	Description	Reset
12:0	reg_sensor_mac_crop_start_y	R/W	Lines before reg_s ensor_mac_crop_start_y will be cropped in each frame if enable r eg_sensor_mac_crop_en.	0xFFF
15:13	Reserved			
28:16	reg_sensor_mac_crop_end_y	R/W	Lines after reg _sensor_mac_crop_end_y will be cropped in each frame if enable r eg_sensor_mac_crop_en.	0xFFF
31:29	Reserved			

19.1.6.33 REG_D0

Table 19.35: REG_D0, Offset Address: 0x0d0

Bits	Name	Access	Description	Reset
0	reg_ttl_as_slvds_enable	R/W	Sub-LVDS lane enable for each lane	0x0
7:1	Reserved			
9:8	reg_ttl_as_slvds_bit_mode	R/W	Sub-LVDS bit mode 2'b00: 8-bit 2'b01: 10-bit 2'b10: 12-bit	0x2
10	reg_ttl_as_slvds_data_reverse	R/W	Sub-LVDS data packet bit inverse	0x0
11	Reserved			
12	reg_ttl_as_slvds_hdr_mode	R/W	Sub-LVDS HDR mode enable	0x0
13	reg_ttl_as_slvds_hdr_pattern	R/W	Sub-LVDS HDR pattern mode 1'b0: pattern 1 1'b1: pattern 2	0x0
31:14	Reserved			

19.1.6.34 REG_D4

Table 19.36: REG_D4, Offset Address: 0x0d4

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_slvds_sync_1st	R/W	Sub-LVDS SYNC code 1st word	0xFFF
15:12	Reserved			
27:16	reg_ttl_as_slvds_sync_2nd	R/W	Sub-LVDS SYNC code 2nd word	0x000
31:28	Reserved			

19.1.6.35 REG_D8

Table 19.37: v, Offset Address: 0x0d8

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_slvds_sync_3rd	R/W	Sub-LVDS SYNC code 3rd word	0x000
15:12	Reserved			
27:16	reg_ttl_as_slvds_norm_bk_sav	R/W	Normal mode blanking SAV	0xAB0
31:28	Reserved			

19.1.6.36 REG_DC

Table 19.38: REG_DC, Offset Address: 0x0dc

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_slvds_norm_bk_eav	R/W	Normal mode blanking EAV	0xB60
15:12	Reserved			
27:16	reg_ttl_as_slvds_norm_sav	R/W	Normal mode active SAV	0x800
31:28	Reserved			

19.1.6.37 REG_E0

Table 19.39: REG_E0, Offset Address: 0x0e0

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_slvds_norm_eav	R/W	Normal mode active EAV	0x9D0
15:12	Reserved			
27:16	reg_ttl_as_slvds_n0_bk_sav	R/W	HDR mode n0 blanking SAV	0x2B0
31:28	Reserved			

19.1.6.38 REG_E4

Table 19.40: REG_E4, Offset Address: 0x0e4

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_slvds_n0_bk_eav	R/W	HDR mode n0 blanking EAV	0x360
15:12	Reserved			
27:16	reg_ttl_as_slvds_n1_bk_sav	R/W	HDR mode n1 blanking SAV	0x6B0
31:28	Reserved			

19.1.6.39 REG_E8

Table 19.41: REG_E8, Offset Address: 0x0e8

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_slvds_n1_bk_eav	R/W	HDR mode n1 blanking EAV	0x760
15:12	Reserved			
27:16	reg_ttl_as_slvds_n0_lef_sav	R/W	Sub-LVDS mode: n0 long exposure sav Sub-LVDS 12-bit LEF SAV n0 (801) Sub-LVDS 10-bit LEF SAV n0 (004) HiSPi P-SP mode: SOL T1 (800)	0x801
31:28	Reserved			

19.1.6.40 REG_EC

Table 19.42: REG_EC, Offset Address: 0x0ec

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_slvds_n0_lef_eav	R/W	Sub-LVDS mode: n0 long exposure eav Sub-LVDS 12-bit LEF EAV n0 (9D1) Sub-LVDS 10-bit LEF EAV n0 (1D4) HiSPi P-SP mode: EOL T1 (A00)	0x9D1
15:12	Reserved			
27:16	reg_ttl_as_slvds_n0_sef_sav	R/W	Sub-LVDS mode: n0 short exposure sav Sub-LVDS 12-bit SEF SAV n0 (802) Sub-LVDS 10-bit SEF SAV n0 (008) HiSPi P-SP mode: SOL T2 (820)	0x802
31:28	Reserved			

19.1.6.41 REG_F0

Table 19.43: REG_F0, Offset Address: 0x0f0

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_slvds_n0_sef_eav	R/W	Sub-LVDS mode: n0 short exposure eav Sub-LVDS 12-bit SEF EAV n0 (9D2) Sub-LVDS 10-bit SEF EAV n0 (1d8) HiSPi P-SP mode: EOL T2 (A20)	0x9D2
15:12	Reserved			
27:16	reg_ttl_as_slvds_n1_lef_sav	R/W	Sub-LVDS mode: n1 long exposure sav Sub-LVDS 12-bit LEF SAV n1 (C01) Sub-LVDS 10-bit LEF SAV n1 (404) HiSPi P-SP mode: SOF T1 (C00)	0xC01
31:28	Reserved			

19.1.6.42 REG_F4

Table 19.44: REG_F4, Offset Address: 0x0f4

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_slvds_n1_lef_eav	R/W	Sub-LVDS mode: n1 long exposure eav Sub-LVDS 12-bit LEF EAV n1 (DD1) Sub-LVDS 10-bit LEF EAV n1 (5D4) HiSPi P-SP mode: EOF T1 (E00)	0xDD1
15:12	Reserved			
27:16	reg_ttl_as_slvds_n1_sef_sav	R/W	Sub-LVDS mode: n1 short exposure sav Sub-LVDS 12-bit SEF SAV n1 (C02) Sub-LVDS 10-bit SEF SAV n1 (408) HiSPi P-SP mode: SOF T2 (C20)	0xC02
31:28	Reserved			

19.1.6.43 REG_F8

Table 19.45: REG_F8, Offset Address: 0x0f8

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_slvds_n1_sef_eav	R/W	Sub-LVDS mode: n1 short exposure eav Sub-LVDS 12-bit SEF EAV n1 (DD2) Sub-LVDS 10-bit SEF EAV n1 (5D8) HiSPi P-SP mode: EOF T2 (E20)	0xDD2
31:12	Reserved			

19.1.6.44 REG_FC

Table 19.46: REG_FC, Offset Address: 0x0fc

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_slvds_vs_gen_sync_code	R/W	vs generate sync code value using scenario: HiSPi P-SP HDR	0xC00
12	reg_ttl_as_slvds_vs_gen_by_sync_code	R/W	vs generate by identical sync code using scenario: HiSPi P-SP HDR	0x0
31:13	Reserved			

19.1.6.45 REG_100

Table 19.47: REG_100, Offset Address: 0x100

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_slvds_n0_lsef_sav	R/W	SAV for n0 long & short exposure both exist line only used for pattern 2	0x803
15:12	Reserved			
27:16	reg_ttl_as_slvds_n0_lsef_eav	R/W	EAV for n0 long & short exposure both exist line only used for pattern 2	0x9D3
31:28	Reserved			

19.1.6.46 REG_104

Table 19.48: REG_104, Offset Address: 0x104

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_slvds_n1_lsef_sav	R/W	SAV for n1 long & short exposure both exist line only used for pattern 2	0xC03
15:12	Reserved			
27:16	reg_ttl_as_slvds_n1_lsef_eav	R/W	EAV for n1 long & short exposure both exist line only used for pattern 2	0xDD3
31:28	Reserved			

19.1.6.47 REG_108

Table 19.49: REG_108, Offset Address: 0x108

Bits	Name	Access	Description	Reset
13:0	reg_ttl_as_slvds_hdr_p2_hsize	R/W	Hsize for pattern 2	0xF0
15:14	Reserved			
29:16	reg_ttl_as_slvds_hdr_p2_hblank	R/W	Hblank size for pattern 2	0x14
31:30	Reserved			

19.1.6.48 REG_110

Table 19.50: REG_110, Offset Address: 0x110

Bits	Name	Access	Description	Reset
0	reg_ttl_as_hispi_mode	R/W	HiSPi mode enable 1'b0: Sub-LVDS 1'b1: HiSPi	0x0
1	reg_ttl_as_hispi_use_hsize	R/W	HiSPi DE de-assert by register count	0x0
3:2	Reserved			
4	reg_ttl_as_hispi_hdr_psp_mode	R/W	HiSPi P-SP HDR mode enable	0x0
31:5	Reserved			

19.1.6.49 REG_114

Table 19.51: REG_114, Offset Address: 0x114

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_hispi_norm_sof	R/W	HiSPi SOF sync code	0xC00
15:12	Reserved			
27:16	reg_ttl_as_hispi_norm_eof	R/W	HiSPi EOF sync code	0xE00
31:28	Reserved			

19.1.6.50 REG_118

Table 19.52: REG_118, Offset Address: 0x118

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_hispi_hdr_t1_sof	R/W	HiSPi HDR T1 SOF	0xC00
15:12	Reserved			
27:16	reg_ttl_as_hispi_hdr_t1_eof	R/W	HiSPi HDR T1 EOF	0xE00
31:28	Reserved			

19.1.6.51 REG_11C

Table 19.53: REG_11C, Offset Address: 0x11c

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_hispi_hdr_t1_sol	R/W	HiSPi HDR T1 SOL	0x800
15:12	Reserved			
27:16	reg_ttl_as_hispi_hdr_t1_eol	R/W	HiSPi HDR T1 EOL	0xA00
31:28	Reserved			

19.1.6.52 REG_120

Table 19.54: REG_120, Offset Address: 0x120

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_hispi_hdr_t2_sof	R/W	HiSPi HDR T2 SOF	0xC20
15:12	Reserved			
27:16	reg_ttl_as_hispi_hdr_t2_eof	R/W	HiSPi HDR T2 EOF	0xE20
31:28	Reserved			

19.1.6.53 REG_124

Table 19.55: REG_124, Offset Address: 0x124

Bits	Name	Access	Description	Reset
11:0	reg_ttl_as_hispi_hdr_t2_sol	R/W	HiSPi HDR T2 SOL	0x820
15:12	Reserved			
27:16	reg_ttl_as_hispi_hdr_t2_eol	R/W	HiSPi HDR T2 EOL	0xA20
31:28	Reserved			

19.2 VDP (Video Display Processor)

19.2.1 Overview

The VDP module can superimpose graphics data on video data and then output it through different display channels. Video data can be read from memory or receive video output from the VPSS module. Graphics data must be read from memory.

19.2.2 Architecture Description

The overall architecture of VDP is as shown in the figure:

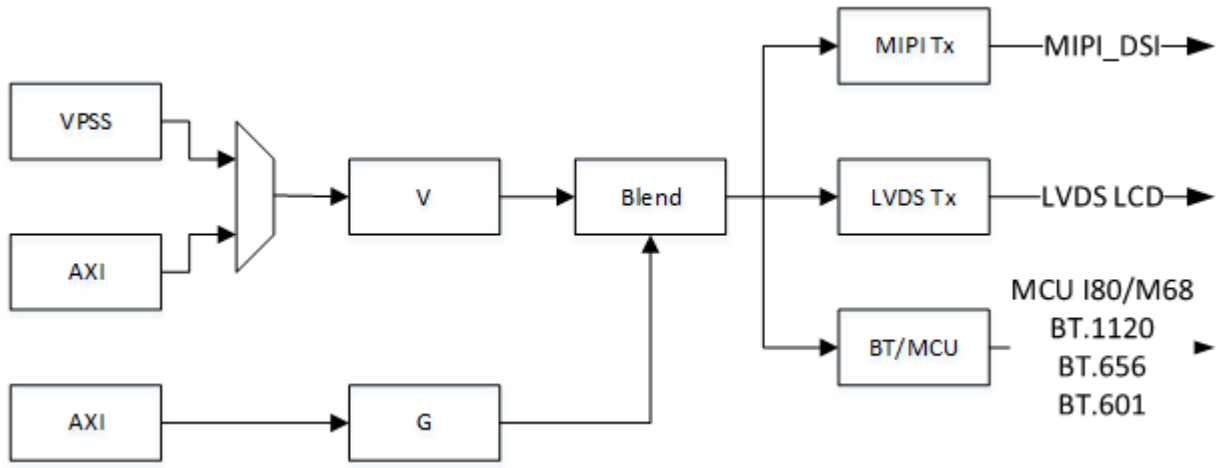


Diagram 19.16: VDP functional block diagram

The chipset only supports one BT.656/BT.601/BT.1120/MCU-I80/MCU-M68/LVDS/MIPI DSI/serial TTL output

- Bus data reading and data processing, including video layer V and graphics layer G.
- The video layer can receive different image formats (YUV422, YUV420, YUV444, RBG packet, NV12/NV21, YUYV-packet) and convert them into the timing and format required by the display channel.
- The graphics layer supports OSD in ARGB8888, ARGB4444, ARGB1555, 8-bit/4-bit LUT, font base and other formats.
- Support OSD compression format.
- Blend: Single layer graphics layer G video graphics overlay.

The characteristics of VDP are as follows:

- MIPI Tx output interface: supports up to 1080P@60fps RGB24-bit output.
- LVDS output: supports up to 720P@60fps RGB24-bit output.
- Digital output interface: (BT format only support progressive mode)
  - Support ITU-R BT.1120 output
  - Support ITU-R BT.656 output
  - Support ITU-R BT.601 output

  - MCU I80/M68 8-bit output
  - Support serial TTL parallel output
- Video layer: V layer.
- Graphic layer: G layer.
- Superposition characteristics: V/G 256-order linear superposition, and supports Gamma correction.
- VDP contains a display channel, a timing interrupt and a low-bandwidth interrupt.

19.2.3 Way of Working

19.2.3.1 Clock Configuration

VDP is equipped with a dedicated clock generator and the architecture is as follows:

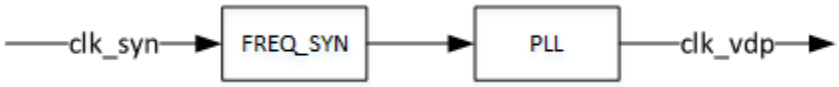


Diagram 19.17: VDP clock generator

- clk_syn : 1GHz or 1.2Ghz
- FREQ_SYN: 6-bit integer, 26-bit decimal frequency generator
- PLL: generates VDP clock and clock required for MIPI Tx/LVDS serial

19.2.3.2 Reset

VDP reset includes a hardware reset and a software reset.

Before performing an AXI bus reset:

- Turn off all AXI access.
- Confirm that the AXI access operation has ended, and then reset the configuration bus.

19.2.3.3 Output Interface

VDP supports the following three interface outputs

- MIPI DSI
- LVDS
- ITU-R BT.1120/ITU-R BT.656/ITU-R BT.601/MCU I80/MCU M68/Serial TTL output

19.2.3.4 Interrupt

VDP supports two types of interrupts

- Vertical timing interrupt
- Low bandwidth interrupt

19.2.3.5 Vertical Timing Interrupt

VDP contains a display channel, corresponding to a vertical timing interrupt:

- Support frame start/end interrupt.
- Support interrupt mask configurable.
- Each interrupt can be cleared by writing 1.

19.2.3.6 Low Bandwidth Interrupt

VDP supports polling mode to provide low bandwidth status:

- Interrupt mask configurable.
- Interrupt is cleared by writing 1.

19.2.4 Function Description

19.2.4.1 Video Layer Functionality

19.2.4.2 Video Layer V Properties

- Support input image formats: 400, planar-420, planar-422, planar-444, RGB packet.
- The minimum input resolution is 64x64, and the maximum input resolution is 1920x1080.
- The minimum output resolution is 64x64, and the maximum output resolution is 1920x1080.
- Support input data bit width: 8-bit.
- YUV420 horizontal/vertical resolution is a multiple of 2.
- YUV422 horizontal resolution is a multiple of 2.
- YUV400/Planar-444(RGB or YUV)/RGB packet has no resolution limit.
- The source starting address is configurable, and the address is 32-byte aligned.
  - Source stride configurable, 32-byte aligned.
- Support color space conversion and contrast/brightness adjustment.
- Support video layer BT.601, BT.709 color gamut conversion.
- Support configurable display position/size and can be displayed anywhere on the screen.

19.2.4.3 Graphics Layer G Properties

- Support input pixel formats: ARGB8888, ARGB4444, ARGB1555, LUT8, LUT4, LUT1.
- The minimum input resolution is 64x64, and the maximum input resolution is 1920x1080.
- The source starting address is configurable, and the address is 32-byte aligned.
  - Source stride configurable, 32-byte aligned.
  - Support color space conversion.
  - Support configurable display position and can be displayed anywhere on the screen.
  - Support 0~255 alpha.
  - Support colorkey processing.

19.2.4.4 Overlay Processing

VDP only supports one video layer and one graphics layer overlay.

19.2.4.5 Overlay Properties

Supports 8 window overlay.

19.2.4.6 Display Channel Characteristics

- Can be used as HD and SD output channels.
  - Only one output interface can be selected for output.
  - Supports typical 1080P@60fps, 720P@60fps output timing.

19.2.4.7 Timing Configuration

The VDP output interface can be connected to different chip interfaces and supports the configuration of various typical and atypical timing sequences.

All timing parameters must be configured before the interface is turned on.

19.2.4.8 HD Output Interface MIPI Tx

- Support RGB666, RGB565, RGB10-10-10 output.
- Support 1080P@60fps 4-channel display.
- Support 720P@60fps 2/4-channel display.

19.2.4.9 HD Output Interface BT.1120 Features

- Support YUV422 8bit output.
- Clip that supports data clamping. According to the interface protocol, the Y clamping range is 16~235, and the C clamping range is 16~240.
- Support the following typical output timings: 720P, 1080P.
- 16 bit data, Y is in the high bit by default, C is in the low bit, YC positions are interchangeable.
- Support color signal Cb/Cr timing interchange.
- Support output associated clock inversion.
- Only Progressive timing is supported.

19.2.4.10 SD output interface BT.656 features

- Support YUV422 8bit output.
- Clip that supports data clamping. According to the interface protocol, the Y clamping range is 16~235, and the C clamping range is 16~240.
- Support Y/C timing interchange and color signal Cb/Cr timing interchange.
- Only Progressive timing is supported.

19.2.4.11 BT.601 Features

- Support YUV422 8-bit output.
- YC data range 0 ~ 255.
- Support VS/HS sync signal output.
- Support Y/C timing interchange and color signal Cb/Cr timing interchange.
- Only Progressive timing is supported.

19.2.4.12 MCU Features

- Support 8-bit data, 4-bit control output.
- Support I80/M68 format output.

19.2.4.13 LCD LVDS Output Interface

- Support 1-link LVDS output.
- Support 6-bit/8-bit RGB serial output.
- Support VESA/JEIDA format output.

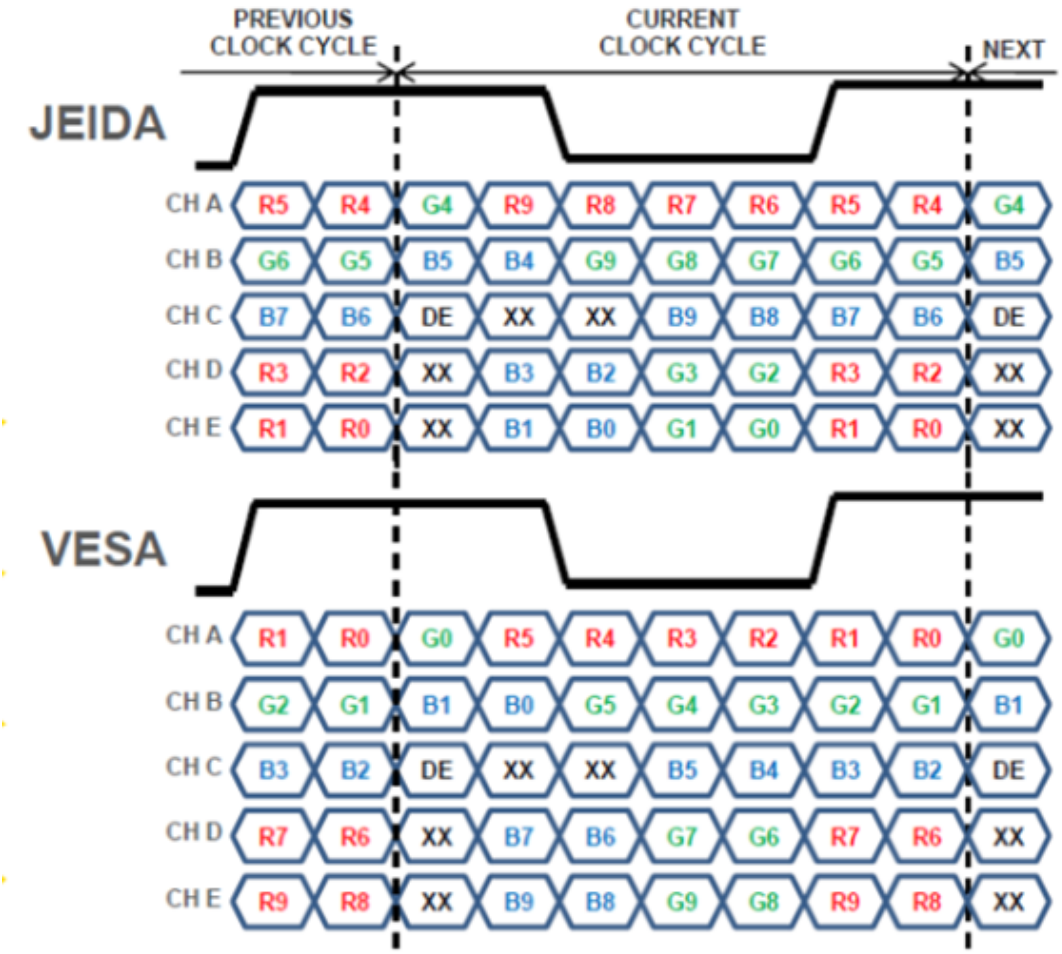


Diagram 19.18: LVDS JEIDA/VESA mode data format

19.2.5 VDP Register Overview

The VDP DISP register location is 0X0A08_8000 ~ 0x0A08_83FF.

Table 19.56: VDP DISP Registers Overview

Name	Address Offset	Description
REG_00	0x000	
REG_01	0x004	
REG_02	0x008	
REG_03	0x00c	
REG_04	0x010	
REG_05	0x014	
REG_06	0x018	
REG_07	0x01c	
REG_13	0x034	

continues on next page

Table 19.56 – continued from previous page

Name	Address Offset	Description
REG_14	0x038	
REG_15	0x03c	
REG_16	0x040	
REG_17	0x044	
REG_18	0x048	
REG_19	0x04c	
REG_20	0x050	
REG_21	0x054	
REG_22	0x058	
REG_23	0x05c	
REG_24	0x060	
REG_25	0x064	
REG_26	0x068	
REG_27	0x06c	
REG_28	0x070	
REG_29	0x074	
REG_30	0x078	
REG_31	0x07c	
REG_32	0x080	
REG_33	0x084	
REG_34	0x088	
REG_35	0x08c	
REG_36	0x090	
REG_37	0x094	
REG_39	0x09c	
REG_40	0x0a0	
REG_41	0x0a4	
REG_42	0x0a8	
REG_43	0x0ac	
REG_CATCH	0x0c0	
REG_OSD_FIFO_M0	0x130	
REG_OSD_FIFO_M1	0x134	
REG_OSD_FIFO_CNT0	0x138	
REG_OSD_FIFO_CNT1	0x13c	
REG_OSD_FIFO_CNT2	0x140	
REG_OSD_FIFO_CNT3	0x144	
REG_GAMMA_CTRL	0x180	
REG_GAMMA_WR_LUT	0x184	
REG_MCU_IF_CTRL	0x200	
REG_HW_MCU_AUTO	0x210	
REG_HW_MCU_CMD	0x214	
REG_HW_MCU_CMD_0	0x218	
REG_HW_MCU_CMD_1	0x21c	
REG_HW_MCU_CMD_2	0x220	
REG_HW_MCU_CMD_3	0x224	
REG_HW_MCU_CMD_4	0x228	
REG_HW_MCU_CMD_5	0x22c	
REG_HW_MCU_CMD_6	0x230	
REG_HW_MCU_CMD_7	0x234	

continues on next page

Table 19.56 – continued from previous page

Name	Address Offset	Description
REG_HW_MCU_OV	0x238	
REG_SRGB_CTRL	0x240	
COV_W0_CFG	0x280	
COV_W0_SIZE	0x284	
COV_W0_COLOR	0x288	
COV_W1_CFG	0x28c	
COV_W1_SIZE	0x290	
COV_W1_COLOR	0x294	
COV_W2_CFG	0x298	
COV_W2_SIZE	0x29c	
COV_W2_COLOR	0x2a0	
COV_W3_CFG	0x2a4	
COV_W3_SIZE	0x2a8	
COV_W3_COLOR	0x2ac	
REG_TGEN_LITE_SIZE	0x304	
REG_TGEN_LITE_VS	0x308	
REG_TGEN_LITE_HS	0x314	

The VDP OSD register location is 0X0A08_8800 ~ 0x0A08_89FF.

Table 19.57: VDP OSD Registers Overview

Name	Address Offset	Description
VGOP_REG_0	0x000	OW0_Setting_0
VGOP_REG_1	0x004	OW0_Setting_1
VGOP_REG_2	0x008	OW0_Setting_2
VGOP_REG_3	0x00c	OW0_Setting_3
VGOP_REG_4	0x010	OW0_Setting_3_1
VGOP_REG_5	0x014	OW0_Setting_4
VGOP_REG_6	0x018	OW0_Setting_5
VGOP_REG_10	0x020	OW1_Setting_0
VGOP_REG_11	0x024	OW1_Setting_1
VGOP_REG_12	0x028	OW1_Setting_2
VGOP_REG_13	0x02c	OW1_Setting_3
VGOP_REG_14	0x030	OW1_Setting_3_1
VGOP_REG_15	0x034	OW1_Setting_4
VGOP_REG_16	0x038	OW1_Setting_5
VGOP_REG_20	0x040	OW2_Setting_0
VGOP_REG_21	0x044	OW2_Setting_1
VGOP_REG_22	0x048	OW2_Setting_2
VGOP_REG_23	0x04c	OW2_Setting_3
VGOP_REG_24	0x050	OW2_Setting_3_1
VGOP_REG_25	0x054	OW2_Setting_4
VGOP_REG_26	0x058	OW2_Setting_5
VGOP_REG_30	0x060	OW3_Setting_0
VGOP_REG_31	0x064	OW3_Setting_1
VGOP_REG_32	0x068	OW3_Setting_2
VGOP_REG_33	0x06c	OW3_Setting_3
VGOP_REG_34	0x070	OW3_Setting_3_1
VGOP_REG_35	0x074	OW3_Setting_4

continues on next page

Table 19.57 – continued from previous page

Name	Address Offset	Description
VGOP_REG_36	0x078	OW3_Setting_5
VGOP_REG_40	0x080	OW4_Setting_0
VGOP_REG_41	0x084	OW4_Setting_1
VGOP_REG_42	0x088	OW4_Setting_2
VGOP_REG_43	0x08c	OW4_Setting_3
VGOP_REG_44	0x090	OW4_Setting_3_1
VGOP_REG_45	0x094	OW4_Setting_4
VGOP_REG_46	0x098	OW4_Setting_5
VGOP_REG_50	0x0a0	OW5_Setting_0
VGOP_REG_51	0x0a4	OW5_Setting_1
VGOP_REG_52	0x0a8	OW5_Setting_2
VGOP_REG_53	0x0ac	OW5_Setting_3
VGOP_REG_54	0x0b0	OW5_Setting_3_1
VGOP_REG_55	0x0b4	OW5_Setting_4
VGOP_REG_56	0x0b8	OW5_Setting_5
VGOP_REG_60	0x0c0	OW6_Setting_0
VGOP_REG_61	0x0c4	OW6_Setting_1
VGOP_REG_62	0x0c8	OW6_Setting_2
VGOP_REG_63	0x0cc	OW6_Setting_3
VGOP_REG_64	0x0d0	OW6_Setting_3_1
VGOP_REG_65	0x0d4	OW6_Setting_4
VGOP_REG_66	0x0d8	OW6_Setting_5
VGOP_REG_70	0x0e0	OW7_Setting_0
VGOP_REG_71	0x0e4	OW7_Setting_1
VGOP_REG_72	0x0e8	OW7_Setting_2
VGOP_REG_73	0x0ec	OW7_Setting_3
VGOP_REG_74	0x0f0	OW7_Setting_3_1
VGOP_REG_75	0x0f4	OW7_Setting_4
VGOP_REG_76	0x0f8	OW7_Setting_5
VGOP_REG_80	0x100	Common
VGOP_REG_81	0x104	idx_sram_ctl
VGOP_REG_82	0x108	idx_sram_ctl
VGOP_REG_83	0x10c	clr_key_content
VGOP_REG_84	0x110	const_clr
VGOP_REG_85	0x114	debug
VGOP_REG_86	0x120	fb_thr
VGOP_REG_87	0x124	fb0_setting
VGOP_REG_88	0x128	fb0_init_st
VGOP_REG_89	0x12c	fb0_st_ro
VGOP_REG_90	0x134	fb1_setting
VGOP_REG_91	0x138	fb1_init_st
VGOP_REG_92	0x13c	fb1_st_ro
BW_LIMIT	0x140	
VGOP_DEC_00	0x150	vgop_dec_ctrl
VGOP_DEC_01	0x154	vgop_dec_debug
VGOP_LUT16_0	0x160	vgop_lut16_0_1
VGOP_LUT16_1	0x164	vgop_lut16_2_3
VGOP_LUT16_2	0x168	vgop_lut16_4_5
VGOP_LUT16_3	0x16c	vgop_lut16_6_7

continues on next page

Table 19.57 – continued from previous page

Name	Address Offset	Description
VGOP_LUT16_4	0x170	vgop_lut16_8_9
VGOP_LUT16_5	0x174	vgop_lut16_10_11
VGOP_LUT16_6	0x178	vgop_lut16_12_13
VGOP_LUT16_7	0x17c	vgop_lut16_14_15

19.2.6 VDP Register Description

19.2.6.1 VDP DISP Registers

REG_00

Table 19.58: REG_00, Offset Address: 0x000

Bits	Name	Access	Description	Reset
2:0	reg_disp_sel	R/W	disp soruce selection 0 : from DMA 1 : from scaler_disp Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
8:3	Reserved			
9	reg_vs_pol_lite	R/W	vsync polarity for tgen_lite 0 : high active 1 : low active	0x0
10	reg_hs_pol_lite	R/W	hsync polarity for tgen lite 0 : high active 1 : low active	0x0
11	reg_tgen_en_lite	R/W	tgen lite enable signal 0 : disable 1 : enable	0x0
15:12	reg_fmt_sel	R/W	disp source from DMA 4'h0 : YUV420 planar 4'h1 : YUV422 planar 4'h2 : RGB888 planar 4'h3 : RGB packed {R,G,B} 4'h4 : RGB packed {B,G,R} 4'h5 : Y only 4'h6 : reserved 4'h7 : reserved 4'h8 : NV21 {V,U} 4'h9 : NV12 {U,V} 4'ha : YUV422-SP1{V,U} 4'hb : YUV422-SP2{U,V} 4'hc : YUV2-1 {U,Y,V,Y} 4'hd : YUV2-2 {V,Y,U,Y} 4'he : YUV2-3 {Y,U,Y,V} 4'hf : YUV2-4 {Y,V,Y,U} Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
3 1:16	Reserved			

REG_01

Table 19.59: REG_01, Offset Address: 0x004

Bits	Name	Access	Description	Reset
13:0	reg_vtt	R/W	vtotal Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x167
15:14	Reserved			
29:16	reg_htt	R/W	htotal Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x1df
31:30	Reserved			

REG_02

Table 19.60: REG_02, Offset Address: 0x008

Bits	Name	Access	Description	Reset
13:0	reg_vs_str	R/W	vsync start Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x1
15:14	Reserved			
29:16	reg_vs_stp	R/W	vsync end Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x3
31:30	Reserved			

REG_03

Table 19.61: REG_03, Offset Address: 0x00c

Bits	Name	Access	Description	Reset
13:0	reg_vfde_str	R/W	vfde start Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x6e
15:14	Reserved			
29:16	reg_vfde_stp	R/W	vfde end Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x15d
31:30	Reserved			

REG_04

Table 19.62: REG_04, Offset Address: 0x010

Bits	Name	Access	Description	Reset
13:0	reg_vmde_str	R/W	vmde start Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x6e
15:14	Reserved			
29:16	reg_vmde_stp	R/W	vmde end Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x15d
31:30	Reserved			

REG_05

Table 19.63: REG_05, Offset Address: 0x014

Bits	Name	Access	Description	Reset
13:0	reg_hs_str	R/W	hsync start Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x1
15:14	Reserved			
29:16	reg_hs_stp	R/W	hsync end Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0xa
31:30	Reserved			

REG_06

Table 19.64: REG_06, Offset Address: 0x018

Bits	Name	Access	Description	Reset
13:0	reg_hfde_str	R/W	hfde start Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x14
15:14	Reserved			
29:16	reg_hfde_stp	R/W	hfde end Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x153
31:30	Reserved			

REG_07

Table 19.65: REG_07, Offset Address: 0x01c

Bits	Name	Access	Description	Reset
13:0	reg_hmde_str	R/W	hmde start Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x14
15:14	Reserved			
29:16	reg_hmde_stp	R/W	hmde end Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x153
31:30	Reserved			

REG_13

Table 19.66: REG_13, Offset Address: 0x034

Bits	Name	Access	Description	Reset
31:0	reg_src_y_base_0	R/W	source Y/R/packet DRAM address Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0

REG_14

Table 19.67: REG_14, Offset Address: 0x038

Bits	Name	Access	Description	Reset
7:0	reg_src_y_base_1	R/W	Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
31:8	Reserved			

REG_15

Table 19.68: REG_15, Offset Address: 0x03c

Bits	Name	Access	Description	Reset
31:0	reg_src_u_base_0	R/W	source U/G DRAM address Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0

REG_16

Table 19.69: REG_16, Offset Address: 0x040

Bits	Name	Access	Description	Reset
7:0	reg_src_u_base_1	R/W	Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
31:8	Reserved			

REG_17

Table 19.70: REG_17, Offset Address: 0x044

Bits	Name	Access	Description	Reset
31:0	reg_src_v_base_0	R/W	source V/B DRAM address Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0

REG_18

Table 19.71: REG_18, Offset Address: 0x048

Bits	Name	Access	Description	Reset
7:0	reg_src_v_base_1	R/W	Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
31:8	Reserved			

REG_19

Table 19.72: REG_19, Offset Address: 0x04c

Bits	Name	Access	Description	Reset
23:0	reg_src_y_pitch	R/W	source Y/R/packet line pitch, must be 32-byte aligned Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
31:24	Reserved			

REG_20

Table 19.73: REG_20, Offset Address: 0x050

Bits	Name	Access	Description	Reset
23:0	reg_src_c_pitch	R/W	source UV/GB line pitch, must be 32-byte aligned Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
24	reg_64b_align	R/W	read address 64-byte align enable, only effective when reg_burst_ln se to 3,7,11 or 15	0x0
31:25	Reserved			

REG_21

Table 19.74: REG_21, Offset Address: 0x054

Bits	Name	Access	Description	Reset
15:0	reg_src_x_str	R/W	source crop x start position Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
31:16	reg_src_y_str	R/W	source crop y start position Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0

REG_22

Table 19.75: REG_22, Offset Address: 0x058

Bits	Name	Access	Description	Reset
15:0	reg_src_wd	R/W	effect when source is from DRAM Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
31:16	reg_src_ht	R/W	effect when source is from DRAM Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0

REG_23

Table 19.76: REG_23, Offset Address: 0x05c

Bits	Name	Access	Description	Reset
13:0	reg_out_csc_c00	R/W	display output csc coef C00, s.3.10 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x400
15:14	Reserved			
29:16	reg_out_csc_c01	R/W	display output csc coef C01, s.3.10 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
30	Reserved			
31	reg_out_csc_en	R/W	display output Color Space Convert Enable Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0

REG_24

Table 19.77: REG_24, Offset Address: 0x060

Bits	Name	Access	Description	Reset
13:0	reg_out_csc_c02	R/W	display output csc coef C02, s.3.10 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x59c
15:14	Reserved			
29:16	reg_out_csc_c10	R/W	display output csc coef C10, s.3.10 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x400
31:30	Reserved			

REG_25

Table 19.78: REG_25, Offset Address: 0x064

Bits	Name	Access	Description	Reset
13:0	reg_out_csc_c11	R/W	display output csc coef C11, s.3.10 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x2160
15:14	Reserved			
29:16	reg_out_csc_c12	R/W	display output csc coef C12, s.3.10 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x22db
31:30	Reserved			

REG_26

Table 19.79: REG_26, Offset Address: 0x068

Bits	Name	Access	Description	Reset
13:0	reg_out_csc_c20	R/W	display output csc coef C20, s.3.10 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x400
15:14	Reserved			
29:16	reg_out_csc_c21	R/W	display output csc coef C21, s.3.10 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x717
31:30	Reserved			

REG_27

Table 19.80: REG_27, Offset Address: 0x06c

Bits	Name	Access	Description	Reset
13:0	reg_out_csc_c22	R/W	display output csc coef C22, s.3.10 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
31:14	Reserved			

REG_28

Table 19.81: REG_28, Offset Address: 0x070

Bits	Name	Access	Description	Reset
7:0	reg_out_csc_sub_0	R/W	display output csc sub value, 0~255 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
15:8	reg_out_csc_sub_1	R/W	display output csc sub value, 0~255 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x80
23:16	reg_out_csc_sub_2	R/W	display output csc sub value, 0~255 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x80
31:24	Reserved			

REG_29

Table 19.82: REG_29, Offset Address: 0x074

Bits	Name	Access	Description	Reset
7:0	reg_out_csc_add_0	R/W	display output csc add value, 0~255 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
15:8	reg_out_csc_add_1	R/W	display output csc add value, 0~255 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
23:16	reg_out_csc_add_2	R/W	display output csc add value, 0~255 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
31:24	Reserved			

REG_30

Table 19.83: REG_30, Offset Address: 0x078

Bits	Name	Access	Description	Reset
13:0	reg_in_csc_c00	R/W	display input csc coef C00, s.3.10 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x400
15:14	Reserved			
29:16	reg_in_csc_c01	R/W	display input csc coef C01, s.3.10 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
30	Reserved			
31	reg_in_csc_en	R/W	display input Color Space Convert Enable Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0

REG_31

Table 19.84: REG_31, Offset Address: 0x07c

Bits	Name	Access	Description	Reset
13:0	reg_in_csc_c02	R/W	display input csc coef C02, s.3.10 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x59c
15:14	Reserved			
29:16	reg_in_csc_c10	R/W	display input csc coef C10, s.3.10 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x400
31:30	Reserved			

REG_32

Table 19.85: REG_32, Offset Address: 0x080

Bits	Name	Access	Description	Reset
13:0	reg_in_csc_c11	R/W	display input csc coef C11, s.3.10 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x2160
15:14	Reserved			
29:16	reg_in_csc_c12	R/W	display input csc coef C12, s.3.10 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x22db
31:30	Reserved			

REG_33

Table 19.86: REG_33, Offset Address: 0x084

Bits	Name	Access	Description	Reset
13:0	reg_in_csc_c20	R/W	display input csc coef C20, s.3.10 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x400
15:14	Reserved			
29:16	reg_in_csc_c21	R/W	display input csc coef C21, s.3.10 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x717
31:30	Reserved			

REG_34

Table 19.87: REG_34, Offset Address: 0x088

Bits	Name	Access	Description	Reset
13:0	reg_in_csc_c22	R/W	display input csc coef C22, s.3.10 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
31:14	Reserved			

REG_35

Table 19.88: REG_35, Offset Address: 0x08c

Bits	Name	Access	Description	Reset
7:0	reg_in_csc_sub_0	R/W	display input csc sub value, 0~255 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
15:8	reg_in_csc_sub_1	R/W	display input csc sub value, 0~255 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x80
23:16	reg_in_csc_sub_2	R/W	display input csc sub value, 0~255 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x80
31:24	Reserved			

REG_36

Table 19.89: REG_36, Offset Address: 0x090

Bits	Name	Access	Description	Reset
7:0	reg_in_csc_add_0	R/W	display input csc add value, 0~255 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
15:8	reg_in_csc_add_1	R/W	display input csc add value, 0~255 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
23:16	reg_in_csc_add_2	R/W	display input csc add value, 0~255 Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
31:24	Reserved			

REG_37

Table 19.90: REG_37, Offset Address: 0x094

Bits	Name	Access	Description	Reset
4:0	Reserved			
5	reg_fix_mc	R/W	mde window output fix color enable Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
31:6	Reserved			

REG_39

Table 19.91: REG_39, Offset Address: 0x09c

Bits	Name	Access	Description	Reset
15:0	Reserved			
25:16	reg_fde_r	R/W	fde window r color	0x0
31:26	Reserved			

REG_40

Table 19.92: REG_40, Offset Address: 0x0a0

Bits	Name	Access	Description	Reset
9:0	reg_fde_g	R/W	fde window g color	0x0
15:10	Reserved			
25:16	reg_fde_b	R/W	fde window b color	0x3ff
31:26	Reserved			

REG_41

Table 19.93: REG_41, Offset Address: 0x0a4

Bits	Name	Access	Description	Reset
9:0	reg_mde_r	R/W	mde window r color	0x3ff
15:10	Reserved			
25:16	reg_mde_g	R/W	mde window g color	0x0
31:26	Reserved			

REG_42

Table 19.94: REG_42, Offset Address: 0x0a8

Bits	Name	Access	Description	Reset
9:0	reg_mde_b	R/W	mde window b color	0x0
15:10	Reserved			
17:16	reg_out_bit	R/W	2 : 8-bit 3 : 6-bit others : 10-bit	0x2
19:18	reg_drop_md	R/W	2 : rounding 3 : direct drop others : noise dither	0x1
31:20	Reserved			

REG_43

Table 19.95: REG_43, Offset Address: 0x0ac

Bits	Name	Access	Description	Reset
0	reg_disp_bw_fail	RO	data not ready for display, Low Bandwidth interrupt	
1	reg_clr_disp_bw_fail	W1T	clear reg_disp_bw_fail	
2	reg_osd_bw_fail	RO	OSD data not ready for display, Low Bandwidth interrupt	
3	reg_clr_osd_bw_fail	W1T	clear reg_osd_bw_fail	
31:4	Reserved			

REG_CATCH

Table 19.96: REG_CATCH, Offset Address: 0x0c0

Bits	Name	Access	Description	Reset
1:0	Reserved			
2	reg_qos_sel_rr	R/W	qos criteria selection 0 : data in buffer + outstanding 1 : data in buffer only	0x1
31:3	Reserved			

REG_OSD_FIFO_M0

Table 19.97: REG_OSD_FIFO_M0, Offset Address: 0x130

Bits	Name	Access	Description	Reset
11:0	reg_osd_thres_0	R/W	osd fifo monitor threshold value 0	0x10
15:12	Reserved			
27:16	reg_osd_thres_1	R/W	osd fifo monitor threshold value 1	0x18
31:28	Reserved			

REG_OSD_FIFO_M1

Table 19.98: REG_OSD_FIFO_M1, Offset Address: 0x134

Bits	Name	Access	Description	Reset
11:0	reg_osd_thres_2	R/W	osd fifo monitor threshold value 2	0x20
15:12	Reserved			
27:16	reg_osd_thres_3	R/W	osd fifo monitor threshold value 3	0x28
31:28	Reserved			

REG_OSD_FIFO_CNT0

Table 19.99: REG_OSD_FIFO_CNT0, Offset Address: 0x138

Bits	Name	Access	Description	Reset
31:0	reg_osd_rd_cnt_0	RO	osd fifo data less than threshold value 0 count	

REG_OSD_FIFO_CNT1

Table 19.100: REG_OSD_FIFO_CNT1, Offset Address: 0x13c

Bits	Name	Access	Description	Reset
31:0	reg_osd_rd_cnt_1	RO	osd fifo data less than threshold value 1 count	

REG_OSD_FIFO_CNT2

Table 19.101: REG_OSD_FIFO_CNT2, Offset Address: 0x140

Bits	Name	Access	Description	Reset
31:0	reg_osd_rd_cnt_2	RO	osd fifo data less than threshold value 2 count	

REG_OSD_FIFO_CNT3

Table 19.102: REG_OSD_FIFO_CNT3, Offset Address: 0x144

Bits	Name	Access	Description	Reset
31:0	reg_osd_rd_cnt_3	RO	osd fifo data less than threshold value 3 count	

REG_GAMMA_CTRL

Table 19.103: REG_GAMMA_CTRL, Offset Address: 0x180

Bits	Name	Access	Description	Reset
0	reg_gamma_acc_lut	R/W	apb access gamma lut table	0x0
1	reg_gamma_acc_wr	R/W	0 : read , 1 : write	0x0
2	reg_gamma_en	R/W	gamma enable Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
3	reg_gamma_pre_osd	R/W	1 : gamma -> osd 0 : osd -> gamma Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
31:4	Reserved			

REG_GAMMA_WR_LUT

Table 19.104: REG_GAMMA_WR_LUT, Offset Address: 0x184

Bits	Name	Access	Description	Reset
7:0	reg_gamma_b_wdat	R/W	B gamma Lut write data	0x0
15:8	reg_gamma_g_wdat	R/W	G gamma Lut write data	0x0
23:16	reg_gamma_r_wdat	R/W	R gamma Lut write data	0x0
30:24	reg_gamma_addr	R/W	gamma lut address	0x0
31	reg_gamma_acc_w1t	W1T	write 1 to access lut (wrtie)	

REG_MCU_IF_CTRL

Table 19.105: REG_MCU_IF_CTRL, Offset Address: 0x200

Bits	Name	Access	Description	Reset
0	reg_i80_if_en	R/W	display S.W. mcu interface enable (read prepared data from DRAM)	0x0
1	reg_i80_sw_mode_en	R/W	mcu sw step by step mode enable	0x0
2	reg_i80_hw_if_en	R/W	display H.W. mcu interface enable (read RGB data fromDRAM) this mode is mutually exclusive with reg_i80_if_en, and the control is in REG_HW_MCU related register group	0x0
3	Reserved			
7:4	reg_i80_ctrl_ini	R/W	control signal initial value	0xf
9:8	Reserved			
10	reg_i80_ip_clr_w1t	W1T	write 1 to clear mcu interface IP	
11	reg_i80_run_w1t	W1T	write 1 to start mcu interface IP	
31:12	Reserved			

REG_HW_MCU_AUTO

Table 19.106: REG_HW_MCU_AUTO, Offset Address: 0x210

Bits	Name	Access	Description	Reset
0	reg_mcu_hw_trig	R/W	rising edge to start H.W MCU	0x0
1	reg_mcu_hw_stop	R/W	rising edge to stop H.W. MCU	0x0
3:2	reg_cs_h_hw_blk	R/W	csx signal state in video blanking duration 2'b00 : csx high in h-blanking & v-blanking 2'b01 : csx high in v-blanking 2'b1x : csx high when not sending video data	0x2
4	reg_mcu_565	R/W	H.W. MCU data out format 0 : RGB565, 2 TX cycle per pix 1 : RGB888, 3 TX cycle per pix (RGB666 also with this mode)	0x1
5	reg_pre_cmd_en	R/W	1 : insert cmd before H.W. MCU data send use reg_sw_tx related config, if this bit set to high a. please set reg_sw_tx related config b. set hfde_str number > reg_sw_tx_num	0x0
6	reg_hw_mcu_start_flag	RWC	H.W. MCU interface start to send data, write 1 to clear	
7	reg_hw_mcu_stop_flag	RWC	H.W. MCU interace stop to send data, write 1 to clear	
31:8	Reserved			

REG_HW_MCU_CMD

Table 19.107: REG_HW_MCU_CMD, Offset Address: 0x214

Bits	Name	Access	Description	Reset
0	reg_mcu_sw_trig	R/W	rising edge to trig S.W. command command number is set in reg_mcu_sw_tx_num (max 16 command per trig) command queue is set in reg_sw_tx0 ~ reg_sw_txf	0x0
1	reg_cs_h_sw_idle	R/W	csx signal state after S.W. trig finish 1'b0 : keep low 1'b1 : go high	0x0
2	Reserved			
3	reg_mcu_sw_tx_done	RWC	S.W. TX done flag, write 1 to clear	
7:4	reg_mcu_sw_tx_num	R/W	H.W. MCU engine send s.w. command number(0 ~ 15) -> (1~16)	0x0
11:8	reg_hw_mcu_state	RO	H.W. MCU status 0 : IDLE 1 : SW_TX 2 : WAIT_VS 3 : BLK 4 : HW_TX0 5 : HW_TX1 6 : HW_TX2	
15:12	reg_sw_tx_cnt_ro	RO	S.W. TX number by H.W. when finish, this value is set to 0	
31:16	Reserved			

REG_HW_MCU_CMD_0

Table 19.108: REG_HW_MCU_CMD_0, Offset Address: 0x218

Bits	Name	Access	Description	Reset
9:0	reg_sw_tx_0	R/W	S.W. TX data [9:8] : OP code 2'b00 : command write 2'b01 : data write 2'b10 : command read 2'b11 : data read [7:0] TX data	0x0
15:10	Reserved			
25:16	reg_sw_tx_1	R/W		0x0
31:26	Reserved			

REG_HW_MCU_CMD_1

Table 19.109: REG_HW_MCU_CMD_1, Offset Address: 0x21c

Bits	Name	Access	Description	Reset
9:0	reg_sw_tx_2	R/W		0x0
15:10	Reserved			
25:16	reg_sw_tx_3	R/W		0x0
31:26	Reserved			

REG_HW_MCU_CMD_2

Table 19.110: REG_HW_MCU_CMD_2, Offset Address: 0x220

Bits	Name	Access	Description	Reset
9:0	reg_sw_tx_4	R/W		0x0
15:10	Reserved			
25:16	reg_sw_tx_5	R/W		0x0
31:26	Reserved			

REG_HW_MCU_CMD_3

Table 19.111: REG_HW_MCU_CMD_3, Offset Address: 0x224

Bits	Name	Access	Description	Reset
9:0	reg_sw_tx_6	R/W		0x0
15:10	Reserved			
25:16	reg_sw_tx_7	R/W		0x0
31:26	Reserved			

REG_HW_MCU_CMD_4

Table 19.112: REG_HW_MCU_CMD_4, Offset Address: 0x228

Bits	Name	Access	Description	Reset
9:0	reg_sw_tx_8	R/W		0x0
15:10	Reserved			
25:16	reg_sw_tx_9	R/W		0x0
31:26	Reserved			

REG_HW_MCU_CMD_5

Table 19.113: REG_HW_MCU_CMD_5, Offset Address: 0x22c

Bits	Name	Access	Description	Reset
9:0	reg_sw_tx_a	R/W		0x0
15:10	Reserved			
25:16	reg_sw_tx_b	R/W		0x0
31:26	Reserved			

REG_HW_MCU_CMD_6

Table 19.114: REG_HW_MCU_CMD_6, Offset Address: 0x230

Bits	Name	Access	Description	Reset
9:0	reg_sw_tx_c	R/W		0x0
15:10	Reserved			
25:16	reg_sw_tx_d	R/W		0x0
31:26	Reserved			

REG_HW_MCU_CMD_7

Table 19.115: REG_HW_MCU_CMD_7, Offset Address: 0x234

Bits	Name	Access	Description	Reset
9:0	reg_sw_tx_e	R/W		0x0
15:10	Reserved			
25:16	reg_sw_tx_f	R/W		0x0
31:26	Reserved			

REG_HW_MCU_OV

Table 19.116: REG_HW_MCU_OV, Offset Address: 0x238

Bits	Name	Access	Description	Reset
0	reg_i80_wrx_sw_ov	R/W	S.W. over-write i80 WRX enable	0x0
1	reg_i80_rdx_sw_ov	R/W		0x0
2	reg_i80_cdx_sw_ov	R/W		0x0
3	reg_i80_csx_sw_ov	R/W		0x0
4	reg_i80_dat_sw_ov	R/W		0x0
7:5	Reserved			
8	reg_i80_wrx_sw_dat	R/W		0x0
9	reg_i80_rdx_sw_dat	R/W		0x0
10	reg_i80_cdx_sw_dat	R/W		0x0
11	reg_i80_csx_sw_dat	R/W		0x0
15:12	Reserved			
23:16	reg_i80_dat_sw_dat	R/W		0x0
31:24	Reserved			

REG_SRGB_CTRL

Table 19.117: REG_SRGB_CTRL, Offset Address: 0x240

Bits	Name	Access	Description	Reset
0	reg_srgb_ttl_en	R/W	serial RGB TTL interface enable	0x0
1	reg_srgb_ttl_4t	R/W	serial RGB output cycle per pixel 0 : 3T 1 : 4T	0x0
3:2	Reserved			
5:4	reg_srgb_ttl_g_num	R/W	sw overite serial RGB out G pixel idx, effective when reg_srgb_ttl_sw_seq = 1	0x0
7:6	reg_srgb_ttl_b_num	R/W	sw overite serial RGB out B pixel idx, effective when reg_srgb_ttl_sw_seq = 1	0x0
8	reg_srgb_ttl_sw_seq	R/W	sw redefine serial RGB output squence	0x0
31:9	Reserved			

COV_W0_CFG

Table 19.118: COV_W0_CFG, Offset Address: 0x280

Bits	Name	Access	Description	Reset
11:0	reg_cover_w0_x_str	R/W	cover window 0 horizontal start (include this point) Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
15:12	Reserved			
27:16	reg_cover_w0_y_str	R/W	cover window 0 verticlal start (include this point) Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
30:28	Reserved			
31	reg_cover_w0_enable	R/W	cover window 0 enable Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0

COV_W0_SIZE

Table 19.119: COV_W0_SIZE, Offset Address: 0x284

Bits	Name	Access	Description	Reset
11:0	reg_cover_w0_wd_m1	R/W	cover window 0 widh minus 1 value Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
15:12	Reserved			
27:16	reg_cover_w0_ht_m1	R/W	cover window 0 height minus 1 value Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
31:28	Reserved			

COV_W0_COLOR

Table 19.120: COV_W0_COLOR, Offset Address: 0x288

Bits	Name	Access	Description	Reset
7:0	reg_cover_w0_r	R/W	cover window 0 r color	0x0
15:8	reg_cover_w0_g	R/W	cover window 0 g color	0x0
23:16	reg_cover_w0_b	R/W	cover window 0 b color	0x0
31:24	Reserved			

COV_W1_CFG

Table 19.121: COV_W1_CFG, Offset Address: 0x28c

Bits	Name	Access	Description	Reset
11:0	reg_cover_w1_x_str	R/W	cover window1 horizontal start (include this point) Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
15:12	Reserved			
27:16	reg_cover_w1_y_str	R/W	cover window1 verticlal start (include this point) Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
30:28	Reserved			
31	reg_cover_w1_enable	R/W	cover window 1 enable Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0

COV_W1_SIZE

Table 19.122: COV_W1_SIZE, Offset Address: 0x290

Bits	Name	Access	Description	Reset
11:0	reg_cover_w1_wd_m1	R/W	cover window 1 widh minus 1 value Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
15:12	Reserved			
27:16	reg_cover_w1_ht_m1	R/W	cover window 1 height minus 1 value Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
31:28	Reserved			

COV_W1_COLOR

Table 19.123: COV_W1_COLOR, Offset Address: 0x294

Bits	Name	Access	Description	Reset
7:0	reg_cover_w1_r	R/W	cover window 1 r color	0x0
15:8	reg_cover_w1_g	R/W	cover window 1 g color	0x0
23:16	reg_cover_w1_b	R/W	cover window 1 b color	0x0
31:24	Reserved			

COV_W2_CFG

Table 19.124: COV_W2_CFG, Offset Address: 0x298

Bits	Name	Access	Description	Reset
11:0	reg_cover_w2_x_str	R/W	cover window 2 horizontal start (include this point) Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
15:12	Reserved			
27:16	reg_cover_w2_y_str	R/W	cover window 2 verticlal start (include this point) Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
30:28	Reserved			
31	reg_cover_w2_enable	R/W	cover window 2 enable Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0

COV_W2_SIZE

Table 19.125: COV_W2_SIZE, Offset Address: 0x29c

Bits	Name	Access	Description	Reset
11:0	reg_cover_w2_wd_m1	R/W	cover window 2 widh minus 1 value Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
15:12	Reserved			
27:16	reg_cover_w2_ht_m1	R/W	cover window 2 height minus 1 value Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
31:28	Reserved			

COV_W2_COLOR

Table 19.126: COV_W2_COLOR, Offset Address: 0x2a0

Bits	Name	Access	Description	Reset
7:0	reg_cover_w2_r	R/W	cover window 2 r color	0x0
15:8	reg_cover_w2_g	R/W	cover window 2 g color	0x0
23:16	reg_cover_w2_b	R/W	cover window 2 b color	0x0
31:24	Reserved			

COV_W3_CFG

Table 19.127: COV_W3_CFG, Offset Address: 0x2a4

Bits	Name	Access	Description	Reset
11:0	reg_cover_w3_x_str	R/W	cover window 3 horizontal start (include this point) Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
15:12	Reserved			
27:16	reg_cover_w3_y_str	R/W	cover window 3 verticlal start (include this point) Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
30:28	Reserved			
31	reg_cover_w3_enable	R/W	cover window 3 enable Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0

COV_W3_SIZE

Table 19.128: COV_W3_SIZE, Offset Address: 0x2a8

Bits	Name	Access	Description	Reset
11:0	reg_cover_w3_wd_m1	R/W	cover window 3 widh minus 1 value Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
15:12	Reserved			
27:16	reg_cover_w3_ht_m1	R/W	cover window 3 height minus 1 value Shadow: Yes Shadow Ctrl: up_1t Shadow Read Select: shrd_sel	0x0
31:28	Reserved			

COV_W3_COLOR

Table 19.129: COV_W3_COLOR, Offset Address: 0x2ac

Bits	Name	Access	Description	Reset
7:0	reg_cover_w3_r	R/W	cover window 3 r color	0x0
15:8	reg_cover_w3_g	R/W	cover window 3 g color	0x0
23:16	reg_cover_w3_b	R/W	cover window 3 b color	0x0
31:24	Reserved			

REG_TGEN_LITE_SIZE

Table 19.130: REG_TGEN_LITE_SIZE, Offset Address: 0x304

Bits	Name	Access	Description	Reset
13:0	reg_vtt_lite	R/W	vtotal Shadow: Yes Shadow Ctrl: up_1t_lite Shadow Read Select: shrd_sel	0x167
15:14	Reserved			
29:16	reg_htt_lite	R/W	htotal Shadow: Yes Shadow Ctrl: up_1t_lite Shadow Read Select: shrd_sel	0x1df
31:30	Reserved			

REG_TGEN_LITE_VS

Table 19.131: REG_TGEN_LITE_VS, Offset Address: 0x308

Bits	Name	Access	Description	Reset
13:0	reg_vs_str_lite	R/W	vsync start Shadow: Yes Shadow Ctrl: up_1t_lite Shadow Read Select: shrd_sel	0x1
15:14	Reserved			
29:16	reg_vs_stp_lite	R/W	vsync end Shadow: Yes Shadow Ctrl: up_1t_lite Shadow Read Select: shrd_sel	0x3
31:30	Reserved			

REG_TGEN_LITE_HS

Table 19.132: REG_TGEN_LITE_HS, Offset Address: 0x314

Bits	Name	Access	Description	Reset
13:0	reg_hs_str_lite	R/W	hsync start Shadow: Yes Shadow Ctrl: up_1t_lite Shadow Read Select: shrd_sel	0x1
15:14	Reserved			
29:16	reg_hs_stp_lite	R/W	hsync end Shadow: Yes Shadow Ctrl: up_1t_lite Shadow Read Select: shrd_sel	0xa
31:30	Reserved			

19.2.6.2 VDP_OSD Register Description

VGOP_REG_0

Table 19.133: VGOP_REG_0, Offset Address: 0x000

Bits	Name	Access	Description	Reset
3:0	reg_ow0_format	R/W	OW0 Format 4'b0000: ARGB8888 4'b0100: ARGB4444 4'b0101: ARGB1555 4'b1000: 256LUT-ARGB4444 4'b1010: 16-LUT-ARGB4444 4'b1100: Font-base Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:4	Reserved			

VGOP_REG_1

Table 19.134: VGOP_REG_1, Offset Address: 0x004

Bits	Name	Access	Description	Reset
11:0	reg_ow0_h_start	R/W	OW0 H start pixel, unit: 1pixel Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
15:12	Reserved			
27:16	reg_ow0_h_end	R/W	OW0 H end pixel, unit: 1pixel Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:28	Reserved			

VGOP_REG_2

Table 19.135: VGOP_REG_2, Offset Address: 0x008

Bits	Name	Access	Description	Reset
11:0	reg_ow0_v_start	R/W	OW0 V start pixel, unit: 1line Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
15:12	Reserved			
27:16	reg_ow0_v_end	R/W	OW0 V end pixel, unit: 1line Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:28	Reserved			

VGOP_REG_3

Table 19.136: VGOP_REG_3, Offset Address: 0x00c

Bits	Name	Access	Description	Reset
3:0	Reserved			
31:4	reg_ow0_dram_str_adr_l	R/W	OW0 DRAM Start address[31:0] Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0

VGOP_REG_4

Table 19.137: VGOP_REG_4, Offset Address: 0x010

Bits	Name	Access	Description	Reset
7:0	reg_ow0_dram_str_adr_h	R/W	OW0 DRAM Start address[39:32] Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:8	Reserved			

VGOP_REG_5

Table 19.138: VGOP_REG_5, Offset Address: 0x014

Bits	Name	Access	Description	Reset
3:0	Reserved			
15:4	reg_ow0_dram_strip	R/W	OW0 DRAM Strip Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
22:16	reg_ow0_crop_pixels	R/W	OW0 crop pixels (per line) Format ARGB8888, crop pixels valid value: 0~3 Format ARGB4444/ARGB1555, crop pixels valid value: 0~7 Format 256LUT, crop pixels valid value: 0~15 Format 16-LUT, crop pixels valid value: 0~31 Format Font-base, crop pixels valid value: 0~127 Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:23	Reserved			

VGOP_REG_6

Table 19.139: VGOP_REG_6, Offset Address: 0x018

Bits	Name	Access	Description	Reset
3:0	Reserved			
13:4	reg_ow0_dram_hsize	R/W	OW0 DRAM Hsize ** while reg_odec_en=1, odec_stream_size = {reg_ow_dram_vs ize,reg_ow_dram_hsize} Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
15:14	Reserved			
27:16	reg_ow0_dram_vsize	R/W	OW0 DRAM Vsize Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:28	Reserved			

VGOP_REG_10

Table 19.140: VGOP_REG_10, Offset Address: 0x020

Bits	Name	Access	Description	Reset
3:0	reg_ow1_format	R/W	OW1 Format 4'b0000: ARGB8888 4'b0100: ARGB4444 4'b0101: ARGB1555 4'b1000: 256LUT-ARGB4444 4'b1010: 16-LUT-ARGB4444 4'b1100: Font-base Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:4	Reserved			

VGOP_REG_11

Table 19.141: VGOP_REG_11, Offset Address: 0x024

Bits	Name	Access	Description	Reset
11:0	reg_ow1_h_start	R/W	OW1 H start pixel, unit: 1pixel Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
15:12	Reserved			
27:16	reg_ow1_h_end	R/W	OW1 H end pixel, unit: 1pixel Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:28	Reserved			

VGOP_REG_12

Table 19.142: VGOP_REG_12, Offset Address: 0x028

Bits	Name	Access	Description	Reset
11:0	reg_ow1_v_start	R/W	OW1 V start pixel, unit: 1line Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
15:12	Reserved			
27:16	reg_ow1_v_end	R/W	OW1 V end pixel, unit: 1line Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:28	Reserved			

VGOP_REG_13

Table 19.143: VGOP_REG_13, Offset Address: 0x02c

Bits	Name	Access	Description	Reset
3:0	Reserved			
31:4	reg_ow1_dram_str_adr_l	R/W	OW1 DRAM Start address[31:0] Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0

VGOP_REG_14

Table 19.144: VGOP_REG_14, Offset Address: 0x030

Bits	Name	Access	Description	Reset
7:0	reg_ow1_dram_str_adr_h	R/W	OW1 DRAM Start address[39:32] Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:8	Reserved			

VGOP_REG_15

Table 19.145: VGOP_REG_15, Offset Address: 0x034

Bits	Name	Access	Description	Reset
3:0	Reserved			
15:4	reg_ow1_dram_strip	R/W	OW1 DRAM Strip Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
22:16	reg_ow1_crop_pixels	R/W	OW1 crop pixels (per line) Format ARGB8888, crop pixels valid value: 0~3 Format ARGB4444/ARGB1555, crop pixels valid value: 0~7 Format 256LUT, crop pixels valid value: 0~15 Format 16-LUT, crop pixels valid value: 0~31 Format Font-base, crop pixels valid value: 0~127 Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:23	Reserved			

VGOP_REG_16

Table 19.146: VGOP_REG_16, Offset Address: 0x038

Bits	Name	Access	Description	Reset
3:0	Reserved			
13:4	reg_ow1_dram_hsize	R/W	OW1 DRAM Hsize ** while reg_odec_en=1, odec_stream_size = {reg_ow_dram_vs ize,reg_ow_dram_hsize} Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
15:14	Reserved			
27:16	reg_ow1_dram_vsize	R/W	OW1 DRAM Vsize Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:28	Reserved			

VGOP_REG_20

Table 19.147: VGOP_REG_20, Offset Address: 0x040

Bits	Name	Access	Description	Reset
3:0	reg_ow2_format	R/W	OW2 Format 4'b0000: ARGB8888 4'b0100: ARGB4444 4'b0101: ARGB1555 4'b1000: 256LUT-ARGB4444 4'b1010: 16-LUT-ARGB4444 4'b1100: Font-base Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:4	Reserved			

VGOP_REG_21

Table 19.148: VGOP_REG_21, Offset Address: 0x044

Bits	Name	Access	Description	Reset
11:0	reg_ow2_h_start	R/W	OW2 H start pixel, unit: 1pixel Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
15:12	Reserved			
27:16	reg_ow2_h_end	R/W	OW2 H end pixel, unit: 1pixel Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:28	Reserved			

VGOP_REG_22

Table 19.149: VGOP_REG_22, Offset Address: 0x048

Bits	Name	Access	Description	Reset
11:0	reg_ow2_v_start	R/W	OW2 V start pixel, unit: 1line Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
15:12	Reserved			
27:16	reg_ow2_v_end	R/W	OW2 V end pixel, unit: 1line Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:28	Reserved			

VGOP_REG_23

Table 19.150: VGOP_REG_23, Offset Address: 0x04c

Bits	Name	Access	Description	Reset
3:0	Reserved			
31:4	reg_ow2_dram_str_adr_l	R/W	OW2 DRAM Start address[31:0] Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0

VGOP_REG_24

Table 19.151: VGOP_REG_24, Offset Address: 0x050

Bits	Name	Access	Description	Reset
7:0	reg_ow2_dram_str_adr_h	R/W	OW2 DRAM Start address[39:32] Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:8	Reserved			

VGOP_REG_25

Table 19.152: VGOP_REG_25, Offset Address: 0x054

Bits	Name	Access	Description	Reset
3:0	Reserved			
15:4	reg_ow2_dram_strip	R/W	OW2 DRAM Strip Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
22:16	reg_ow2_crop_pixels	R/W	OW2 crop pixels (per line) Format ARGB8888, crop pixels valid value: 0~3 Format ARGB4444/ARGB1555, crop pixels valid value: 0~7 Format 256LUT, crop pixels valid value: 0~15 Format 16-LUT, crop pixels valid value: 0~31 Format Font-base, crop pixels valid value: 0~127 Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:23	Reserved			

VGOP_REG_26

Table 19.153: VGOP_REG_26, Offset Address: 0x058

Bits	Name	Access	Description	Reset
3:0	Reserved			
13:4	reg_ow2_dram_hsize	R/W	OW2 DRAM Hsize ** while reg_odec_en=1, odec_stream_size = {reg_ow_dram_vs ize,reg_ow_dram_hsize} Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
15:14	Reserved			
27:16	reg_ow2_dram_vsize	R/W	OW2 DRAM Vsize Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:28	Reserved			

VGOP_REG_30

Table 19.154: VGOP_REG_30, Offset Address: 0x060

Bits	Name	Access	Description	Reset
3:0	reg_ow3_format	R/W	OW3 Format 4'b0000: ARGB8888 4'b0100: ARGB4444 4'b0101: ARGB1555 4'b1000: 256LUT-ARGB4444 4'b1010: 16-LUT-ARGB4444 4'b1100: Font-base Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:4	Reserved			

VGOP_REG_31

Table 19.155: VGOP_REG_31, Offset Address: 0x064

Bits	Name	Access	Description	Reset
11:0	reg_ow3_h_start	R/W	OW3 H start pixel, unit: 1pixel Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
15:12	Reserved			
27:16	reg_ow3_h_end	R/W	OW3 H end pixel, unit: 1pixel Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:28	Reserved			

VGOP_REG_32

Table 19.156: VGOP_REG_32, Offset Address: 0x068

Bits	Name	Access	Description	Reset
11:0	reg_ow3_v_start	R/W	OW3 V start pixel, unit: 1line Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
15:12	Reserved			
27:16	reg_ow3_v_end	R/W	OW3 V end pixel, unit: 1line Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:28	Reserved			

VGOP_REG_33

Table 19.157: VGOP_REG_33, Offset Address: 0x06c

Bits	Name	Access	Description	Reset
3:0	Reserved			
31:4	reg_ow3_dram_str_adr_l	R/W	OW3 DRAM Start address[31:0] Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0

VGOP_REG_34

Table 19.158: VGOP_REG_34, Offset Address: 0x070

Bits	Name	Access	Description	Reset
7:0	reg_ow3_dram_str_adr_h	R/W	OW3 DRAM Start address[39:32] Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:8	Reserved			

VGOP_REG_35

Table 19.159: VGOP_REG_35, Offset Address: 0x074

Bits	Name	Access	Description	Reset
3:0	Reserved			
15:4	reg_ow3_dram_strip	R/W	OW3 DRAM Strip Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
22:16	reg_ow3_crop_pixels	R/W	OW3 crop pixels (per line) Format ARGB8888, crop pixels valid value: 0~3 Format ARGB4444/ARGB1555, crop pixels valid value: 0~7 Format 256LUT, crop pixels valid value: 0~15 Format 16-LUT, crop pixels valid value: 0~31 Format Font-base, crop pixels valid value: 0~127 Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:23	Reserved			

VGOP_REG_36

Table 19.160: VGOP_REG_36, Offset Address: 0x078

Bits	Name	Access	Description	Reset
3:0	Reserved			
13:4	reg_ow3_dram_hsize	R/W	OW3 DRAM Hsize ** while reg_odec_en=1, odec_stream_size = {reg_ow_dram_vs ize,reg_ow_dram_hsize} Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
15:14	Reserved			
27:16	reg_ow3_dram_vsize	R/W	OW3 DRAM Vsize Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:28	Reserved			

VGOP_REG_40

Table 19.161: VGOP_REG_40, Offset Address: 0x080

Bits	Name	Access	Description	Reset
3:0	reg_ow4_format	R/W	OW4 Format 4'b0000: ARGB8888 4'b0100: ARGB4444 4'b0101: ARGB1555 4'b1000: 256LUT-ARGB4444 4'b1010: 16-LUT-ARGB4444 4'b1100: Font-base Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:4	Reserved			

VGOP_REG_41

Table 19.162: VGOP_REG_41, Offset Address: 0x084

Bits	Name	Access	Description	Reset
11:0	reg_ow4_h_start	R/W	OW4 H start pixel, unit: 1pixel Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
15:12	Reserved			
27:16	reg_ow4_h_end	R/W	OW4 H end pixel, unit: 1pixel Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:28	Reserved			

VGOP_REG_42

Table 19.163: VGOP_REG_42, Offset Address: 0x088

Bits	Name	Access	Description	Reset
11:0	reg_ow4_v_start	R/W	OW4 V start pixel, unit: 1line Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
15:12	Reserved			
27:16	reg_ow4_v_end	R/W	OW4 V end pixel, unit: 1line Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:28	Reserved			

VGOP_REG_43

Table 19.164: VGOP_REG_43, Offset Address: 0x08c

Bits	Name	Access	Description	Reset
3:0	Reserved			
31:4	reg_ow4_dram_str_adr_l	R/W	OW4 DRAM Start address[31:0] Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0

VGOP_REG_44

Table 19.165: VGOP_REG_44, Offset Address: 0x090

Bits	Name	Access	Description	Reset
7:0	reg_ow4_dram_str_adr_h	R/W	OW4 DRAM Start address[39:32] Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:8	Reserved			

VGOP_REG_45

Table 19.166: VGOP_REG_45, Offset Address: 0x094

Bits	Name	Access	Description	Reset
3:0	Reserved			
15:4	reg_ow4_dram_strip	R/W	OW4 DRAM Strip Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
22:16	reg_ow4_crop_pixels	R/W	OW4 crop pixels (per line) Format ARGB8888, crop pixels valid value: 0~3 Format ARGB4444/ARGB1555, crop pixels valid value: 0~7 Format 256LUT, crop pixels valid value: 0~15 Format 16-LUT, crop pixels valid value: 0~31 Format Font-base, crop pixels valid value: 0~127 Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:23	Reserved			

VGOP_REG_46

Table 19.167: VGOP_REG_46, Offset Address: 0x098

Bits	Name	Access	Description	Reset
3:0	Reserved			
13:4	reg_ow4_dram_hsize	R/W	OW4 DRAM Hsize ** while reg_odec_en=1, odec_stream_size = {reg_ow_dram_vs ize,reg_ow_dram_hsize} Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
15:14	Reserved			
27:16	reg_ow4_dram_vsize	R/W	OW4 DRAM Vsize Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:28	Reserved			

VGOP_REG_50

Table 19.168: VGOP_REG_50, Offset Address: 0x0a0

Bits	Name	Access	Description	Reset
3:0	reg_ow5_format	R/W	OW5 Format 4'b0000: ARGB8888 4'b0100: ARGB4444 4'b0101: ARGB1555 4'b1000: 256LUT-ARGB4444 4'b1010: 16-LUT-ARGB4444 4'b1100: Font-base Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:4	Reserved			

VGOP_REG_51

Table 19.169: VGOP_REG_51, Offset Address: 0x0a4

Bits	Name	Access	Description	Reset
11:0	reg_ow5_h_start	R/W	OW5 H start pixel, unit: 1pixel Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
15:12	Reserved			
27:16	reg_ow5_h_end	R/W	OW5 H end pixel, unit: 1pixel Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:28	Reserved			

VGOP_REG_52

Table 19.170: VGOP_REG_52, Offset Address: 0x0a8

Bits	Name	Access	Description	Reset
11:0	reg_ow5_v_start	R/W	OW5 V start pixel, unit: 1line Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
15:12	Reserved			
27:16	reg_ow5_v_end	R/W	OW5 V end pixel, unit: 1line Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:28	Reserved			

VGOP_REG_53

Table 19.171: VGOP_REG_53, Offset Address: 0x0ac

Bits	Name	Access	Description	Reset
3:0	Reserved			
31:4	reg_ow5_dram_str_adr_l	R/W	OW5 DRAM Start address[31:0] Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0

VGOP_REG_54

Table 19.172: VGOP_REG_54, Offset Address: 0x0b0

Bits	Name	Access	Description	Reset
7:0	reg_ow5_dram_str_adr_h	R/W	OW5 DRAM Start address[39:32] Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:8	Reserved			

VGOP_REG_55

Table 19.173: VGOP_REG_55, Offset Address: 0x0b4

Bits	Name	Access	Description	Reset
3:0	Reserved			
15:4	reg_ow5_dram_strip	R/W	OW5 DRAM Strip Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
22:16	reg_ow5_crop_pixels	R/W	OW5 crop pixels (per line) Format ARGB8888, crop pixels valid value: 0~3 Format ARGB4444/ARGB1555, crop pixels valid value: 0~7 Format 256LUT, crop pixels valid value: 0~15 Format 16-LUT, crop pixels valid value: 0~31 Format Font-base, crop pixels valid value: 0~127 Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:23	Reserved			

VGOP_REG_56

Table 19.174: VGOP_REG_56, Offset Address: 0x0b8

Bits	Name	Access	Description	Reset
3:0	Reserved			
13:4	reg_ow5_dram_hsize	R/W	OW5 DRAM Hsize ** while reg_odec_en=1, odec_stream_size = {reg_ow_dram_vs ize,reg_ow_dram_hsize} Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
15:14	Reserved			
27:16	reg_ow5_dram_vsize	R/W	OW5 DRAM Vsize Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:28	Reserved			

VGOP_REG_60

Table 19.175: VGOP_REG_60, Offset Address: 0x0c0

Bits	Name	Access	Description	Reset
3:0	reg_ow6_format	R/W	OW6 Format 4'b0000: ARGB8888 4'b0100: ARGB4444 4'b0101: ARGB1555 4'b1000: 256LUT-ARGB4444 4'b1010: 16-LUT-ARGB4444 4'b1100: Font-base Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:4	Reserved			

VGOP_REG_61

Table 19.176: VGOP_REG_61, Offset Address: 0x0c4

Bits	Name	Access	Description	Reset
11:0	reg_ow6_h_start	R/W	OW6 H start pixel, unit: 1pixel Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
15:12	Reserved			
27:16	reg_ow6_h_end	R/W	OW6 H end pixel, unit: Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:28	Reserved			

VGOP_REG_62

Table 19.177: VGOP_REG_62, Offset Address: 0x0c8

Bits	Name	Access	Description	Reset
11:0	reg_ow6_v_start	R/W	OW6 V start pixel, unit: 1line Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
15:12	Reserved			
27:16	reg_ow6_v_end	R/W	OW6 V end pixel, unit: 1line Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:28	Reserved			

VGOP_REG_63

Table 19.178: VGOP_REG_63, Offset Address: 0x0cc

Bits	Name	Access	Description	Reset
3:0	Reserved			
31:4	reg_ow6_dram_str_adr_l	R/W	OW6 DRAM Start address[31:0] Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0

VGOP_REG_64

Table 19.179: VGOP_REG_64, Offset Address: 0x0d0

Bits	Name	Access	Description	Reset
7:0	reg_ow6_dram_str_adr_h	R/W	OW6 DRAM Start address[39:32] Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:8	Reserved			

VGOP_REG_65

Table 19.180: VGOP_REG_65, Offset Address: 0x0d4

Bits	Name	Access	Description	Reset
3:0	Reserved			
15:4	reg_ow6_dram_strip	R/W	OW6 DRAM Strip Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
22:16	reg_ow6_crop_pixels	R/W	OW6 crop pixels (per line) Format ARGB8888, crop pixels valid value: 0~3 Format ARGB4444/ARGB1555, crop pixels valid value: 0~7 Format 256LUT, crop pixels valid value: 0~15 Format 16-LUT, crop pixels valid value: 0~31 Format Font-base, crop pixels valid value: 0~127 Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:23	Reserved			

VGOP_REG_66

Table 19.181: VGOP_REG_66, Offset Address: 0x0d8

Bits	Name	Access	Description	Reset
3:0	Reserved			
13:4	reg_ow6_dram_hsize	R/W	OW6 DRAM Hsize ** while reg_odec_en=1, odec_stream_size = {reg_ow_dram_vs ize,reg_ow_dram_hsize} Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
15:14	Reserved			
27:16	reg_ow6_dram_vsize	R/W	OW6 DRAM Vsize Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:28	Reserved			

VGOP_REG_70

Table 19.182: VGOP_REG_70, Offset Address: 0x0e0

Bits	Name	Access	Description	Reset
3:0	reg_ow7_format	R/W	OW7 Format 4'b0000: ARGB8888 4'b0100: ARGB4444 4'b0101: ARGB1555 4'b1000: 256LUT-ARGB4444 4'b1010: 16-LUT-ARGB4444 4'b1100: Font-base Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:4	Reserved			

VGOP_REG_71

Table 19.183: VGOP_REG_71, Offset Address: 0x0e4

Bits	Name	Access	Description	Reset
11:0	reg_ow7_h_start	R/W	OW7 H start pixel, unit: 1pixel Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
15:12	Reserved			
27:16	reg_ow7_h_end	R/W	OW7 H end pixel, unit: 1pixel Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:28	Reserved			

VGOP_REG_72

Table 19.184: VGOP_REG_72, Offset Address: 0x0e8

Bits	Name	Access	Description	Reset
11:0	reg_ow7_v_start	R/W	OW7 V start pixel, unit: 1line Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
15:12	Reserved			
27:16	reg_ow7_v_end	R/W	OW7 V end pixel, unit: 1line Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:28	Reserved			

VGOP_REG_73

Table 19.185: VGOP_REG_73, Offset Address: 0x0ec

Bits	Name	Access	Description	Reset
3:0	Reserved			
31:4	reg_ow7_dram_str_adr_l	R/W	OW7 DRAM Start address[31:0] Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0

VGOP_REG_74

Table 19.186: VGOP_REG_74, Offset Address: 0x0f0

Bits	Name	Access	Description	Reset
7:0	reg_ow7_dram_str_adr_h	R/W	OW7 DRAM Start address[39:32] Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:8	Reserved			

VGOP_REG_75

Table 19.187: VGOP_REG_75, Offset Address: 0x0f4

Bits	Name	Access	Description	Reset
3:0	Reserved			
15:4	reg_ow7_dram_strip	R/W	OW7 DRAM Strip Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
22:16	reg_ow7_crop_pixels	R/W	OW7 crop pixels (per line) Format ARGB8888, crop pixels valid value: 0~3 Format ARGB4444/ARGB1555, crop pixels valid value: 0~7 Format 256LUT, crop pixels valid value: 0~15 Format 16-LUT, crop pixels valid value: 0~31 Format Font-base, crop pixels valid value: 0~127 Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:23	Reserved			

VGOP_REG_76

Table 19.188: VGOP_REG_76, Offset Address: 0x0f8

Bits	Name	Access	Description	Reset
3:0	Reserved			
13:4	reg_ow7_dram_hsize	R/W	OW7 DRAM Hsize ** while reg_odec_en=1, odec_stream_size = {reg_ow_dram_vs ize,reg_ow_dram_hsize} Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
15:14	Reserved			
27:16	reg_ow7_dram_vsize	R/W	OW7 DRAM Vsize Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:28	Reserved			

VGOP_REG_80

osd_common_ctrl

Table 19.189: VGOP_REG_80, Offset Address: 0x100

Bits	Name	Access	Description	Reset
0	reg_ow0_en	R/W	OSD Window0 enable Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
1	reg_ow1_en	R/W	OSD Window1 enable Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
2	reg_ow2_en	R/W	OSD Window2 enable Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
3	reg_ow3_en	R/W	OSD Window3 enable Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
4	reg_ow4_en	R/W	OSD Window4 enable Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
5	reg_ow5_en	R/W	OSD Window5 enable Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
6	reg_ow6_en	R/W	OSD Window6 enable Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
7	reg_ow7_en	R/W	OSD Window7 enable Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
8	reg_vgop_hscal	R/W	VGOP H scale up (x2) Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
9	reg_vgop_vscal	R/W	VGOP V scale up (x2) Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
10	reg_clr_key_en	R/W	Color Key enable Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
11	Reserved			

To be continued ……

Table 19.190: VGOP_REG_80, Offset Address: 0x100 (continued)

Bits	Name	Access	Description	Reset
15:12	reg_vgop_arlen	R/W	AXI-R burst length (INCR mode) Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
30:16	Reserved			
31	reg_vgop_sw_rst	R/W	vgop software reset Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0

VGOP_REG_81

LUT256 SRAM

Table 19.191: VGOP_REG_81, Offset Address: 0x104

Bits	Name	Access	Description	Reset
15:0	reg_idx_wdata	R/W	Idx sram wdata (ARGB4444)	0x0
23:16	reg_idx_adr	R/W	Idx sram address	0x0
31:24	Reserved			

VGOP_REG_82

LUT256 SRAM

Table 19.192: VGOP_REG_82, Offset Address: 0x108

Bits	Name	Access	Description	Reset
15:0	reg_idx_rdata	R	idx sram rdata	
16	reg_idx_wr	R/W	idx sram write	0x0
17	reg_idx_rd	R/W	idx sram read	0x0
18	reg_vgop_db_clr	R/W	vgop debug flag clear	0x0
31:19	Reserved			

VGOP_REG_83

color key content

Table 19.193: VGOP_REG_83, Offset Address: 0x10c

Bits	Name	Access	Description	Reset
23:0	reg_clr_key	R/W	Color Key (RGB888) Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:24	Reserved			

VGOP_REG_84

constant color

Table 19.194: VGOP_REG_84, Offset Address: 0x110

Bits	Name	Access	Description	Reset
15:0	reg_const_argb0	R/W	Constant0 ARGB4444 for Font-base Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:16	reg_const_argb1	R/W	Constant1 ARGB4444 for Font-base Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0

VGOP_REG_85

debug

Table 19.195: VGOP_REG_85, Offset Address: 0x114

Bits	Name	Access	Description	Reset
31:0	reg_vgop_debug	R	debug	

VGOP_REG_86

fb_thr

Table 19.196: VGOP_REG_86, Offset Address: 0x120

Bits	Name	Access	Description	Reset
5:0	reg_fb_clr_hi_thr	R/W	font_box brightness strong threshold , unit: 4 levels Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x30
7:6	Reserved			
13:8	reg_fb_clr_lo_thr	R/W	font_box brightness wrak threshold , unit: 4 levels Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x20
15:14	Reserved			
16	reg_fb_init	R/W	fb strong status init Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
17	reg_fb_font_is_dark	R/W	fb inv control 1 : font color is dark 0 : font color is light Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
19:18	Reserved			
25:20	reg_fb_diff_fnum	R/W	font_box brightness detect interval 0: stop detection update 1: always detection update 2~63: detection update if n-1 continous frames detection value is all difference Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x1
31:26	Reserved			

VGOP_REG_87

fb0_setting

Table 19.197: VGOP_REG_87, Offset Address: 0x124

Bits	Name	Access	Description	Reset
6:0	reg_fb0_width	R/W	font_box_0 width , unit: 1 pixel, 1~128 pixels (reg_fb_width+1) * (reg_fb_num+1) must be equal to attached ow width Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0f
7	Reserved			
12:8	reg_fb0_pix_thr	R/W	font_box_0 strong pixel pixel threshold, unit: 8 pixels Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x10
14:13	reg_fb0_sample_rate	R/W	font_box_0 sample rqtio 0: per pixel 1: 2 pixels 2: 4 pixels 3: 8 pixels Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x1
15	Reserved			
20:16	reg_fb0_num	R/W	font_box_0 box numbers , 1~32 boxes Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x1f
23:21	Reserved			
26:24	reg_fb0_attached_idx	R/W	font_box_0 attached ow number Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
27	Reserved			
28	reg_fb0_en	R/W	font_box_0 enable (active while attached ow is alive and set as font base fromat) Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:29	Reserved			

VGOP_REG_88

fb0_init_st

Table 19.198: VGOP_REG_88, Offset Address: 0x128

Bits	Name	Access	Description	Reset
31:0	reg_fb0_init_st	R/W	font_box_0 init strong value Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0

VGOP_REG_89

fb0_st_ro

Table 19.199: VGOP_REG_89, Offset Address: 0x12c

Bits	Name	Access	Description	Reset
31:0	reg_fb0_record	R	font_box_0 strong value record	

VGOP_REG_90

fb1_setting

Table 19.200: VGOP_REG_90, Offset Address: 0x134

Bits	Name	Access	Description	Reset
6:0	reg_fb1_width	R/W	font_box_1 width , unit: 1 pixel, 1~128 pixels (reg_fb_width+1) * (reg_fb_num+1) must be equal to attached ow width Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0f
7	Reserved			
12:8	reg_fb1_pix_thr	R/W	font_box_1 strong pixel pixel threshold, unit: 8 pixels Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x10
14:13	reg_fb1_sample_rate	R/W	font_box_1 sample rqtio 0: per pixel 1: 2 pixels 2: 4 pixels 3: 8 pixels Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x1
15	Reserved			
20:16	reg_fb1_num	R/W	font_box_1 box numbers , 1~32 boxes Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x1f
23:21	Reserved			
26:24	reg_fb1_attached_idx	R/W	font_box_1 attached ow number Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x7
27	Reserved			
28	reg_fb1_en	R/W	font_box_1 enable (active while attached ow is alive and set as font base fromat) Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:29	Reserved			

VGOP_REG_91

fb1_init_st

Table 19.201: VGOP_REG_91, Offset Address: 0x138

Bits	Name	Access	Description	Reset
31:0	reg_fb1_init_st	R/W	font_box_1 init strong value Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0

VGOP_REG_92

fb1_st_ro

Table 19.202: VGOP_REG_92, Offset Address: 0x13c

Bits	Name	Access	Description	Reset
31:0	reg_fb1_record	R	font_box_1 strong value record	

BW_LIMIT

Table 19.203: BW_LIMIT, Offset Address: 0x140

Bits	Name	Access	Description	Reset
9:0	reg_bwl_win	R/W	B.W. limit window period, unit : clk_axi cycle	0x0
15:10	Reserved			
25:16	reg_bwl_vld	R/W	B.W. limit valid number in reg_bwl_win, unit : Byte	0x0
30:26	Reserved			
31	reg_bwl_en	R/W	B.W. limit enable	0x0

VGOP_DEC_00

vgop_dec_ctrl

Table 19.204: VGOP_DEC_00, Offset Address: 0x150

Bits	Name	Access	Description	Reset
0	reg_odec_en	R/W	vgop decoder enable 0: bypass 1: decoder mode Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
1	reg_odec_int_en	R/W	vgop_decoder INT enable	0x0
2	reg_odec_int_clr	R/W	vgop_decoder INT clear	0x0
3	reg_odec_wdt_en	R/W	vgop wdt enable	0x0
7:4	reg_odec_dbg_ridx	R/W	vgop_decoder debug idx	0x0
8	reg_odec_done	R	vgop_decoder done	
11:9	Reserved			
14:12	reg_odec_attached_idx	R/W	vgop decoder attached ow number Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
15	Reserved			
18:16	reg_odec_wdt_fdiv_bit	R/W	wdt precision	0x0
23:19	Reserved			
31:24	reg_odec_int_vec	R	vgop_decoder INT vector 1: odec done 2: Watch Dog time out 4: detect reduntant data input	

VGOP_DEC_01

vgop_dec_debug

Table 19.205: VGOP_DEC_01, Offset Address: 0x154

Bits	Name	Access	Description	Reset
31:0	reg_odec_dbg_rdata	R	vgop decoder debug rdata	

VGOP_LUT16_0

vgop_lut16_0_1

Table 19.206: VGOP_LUT16_0, Offset Address: 0x160

Bits	Name	Access	Description	Reset
15:0	reg_lut16_cnt0	R/W	LUT 16 content 0 (ARGB4444) Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:16	reg_lut16_cnt1	R/W	LUT 16 content 1 (ARGB4444) Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0

VGOP_LUT16_1

vgop_lut16_2_3

Table 19.207: VGOP_LUT16_1, Offset Address: 0x164

Bits	Name	Access	Description	Reset
15:0	reg_lut16_cnt2	R/W	LUT 16 content 2 (ARGB4444) Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:16	reg_lut16_cnt3	R/W	LUT 16 content 3(ARGB4444) Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0

VGOP_LUT16_2

vgop_lut16_4_5

Table 19.208: VGOP_LUT16_2, Offset Address: 0x168

Bits	Name	Access	Description	Reset
15:0	reg_lut16_cnt4	R/W	LUT 16 content 4 (ARGB4444) Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:16	reg_lut16_cnt5	R/W	LUT 16 content 5 (ARGB4444) Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0

VGOP_LUT16_3

vgop_lut16_6_7

Table 19.209: VGOP_LUT16_3, Offset Address: 0x16c

Bits	Name	Access	Description	Reset
15:0	reg_lut16_cnt6	R/W	LUT 16 content 6 (ARGB4444) Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:16	reg_lut16_cnt7	R/W	LUT 16 content 7 (ARGB4444) Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0

VGOP_LUT16_4

vgop_lut16_8_9

Table 19.210: VGOP_LUT16_4, Offset Address: 0x170

Bits	Name	Access	Description	Reset
15:0	reg_lut16_cnt8	R/W	LUT 16 content 8 (ARGB4444) Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:16	reg_lut16_cnt9	R/W	LUT 16 content 9 (ARGB4444) Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0

VGOP_LUT16_5

vgop_lut16_10_11

Table 19.211: VGOP_LUT16_5, Offset Address: 0x174

Bits	Name	Access	Description	Reset
15:0	reg_lut16_cnt10	R/W	LUT 16 content 10 (ARGB4444) Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:16	reg_lut16_cnt11	R/W	LUT 16 content 11 (ARGB4444) Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0

VGOP_LUT16_6

vgop_lut16_12_13

Table 19.212: VGOP_LUT16_6, Offset Address: 0x178

Bits	Name	Access	Description	Reset
15:0	reg_lut16_cnt12	R/W	LUT 16 content 12 (ARGB4444) Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:16	reg_lut16_cnt13	R/W	LUT 16 content 13 (ARGB4444) Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0

VGOP_LUT16_7

vgop_lut16_14_15

Table 19.213: VGOP_LUT16_7, Offset Address: 0x17c

Bits	Name	Access	Description	Reset
15:0	reg_lut16_cnt14	R/W	LUT 16 content 14 (ARGB4444) Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0
31:16	reg_lut16_cnt15	R/W	LUT 16 content 15 (ARGB4444) Shadow: Yes Shadow Ctrl: shdw_update Shadow Read Select: shdw_read_sel	0x0

19.3 MIPI Rx

19.3.1 Overview

The main function of the MIPI Rx (Mobile Industry Processor Interface Receiver) module is to receive video data transmitted by the CMOS sensor. It supports MIPI D-PHY, sub-LVDS (Low-Voltage Differential Signal), HiSPi (High-Speed Serial Pixel Interface), etc. Different serial video signal inputs are processed and converted into internal video timing, which is passed to the next-level video processing module (ISP).

The MIPI Rx module can be subdivided into two parts: PHY and Controller. The PHY module integrates analog and digital parts and mainly converts serial signals into parallel signals. The Controller module is responsible for decoding different video data formats and transmitting them to Back-end video processing module (ISP). The functional block diagram and its location in the system are shown in the diagram *MIPI Rx functional block diagram and location in the system*.

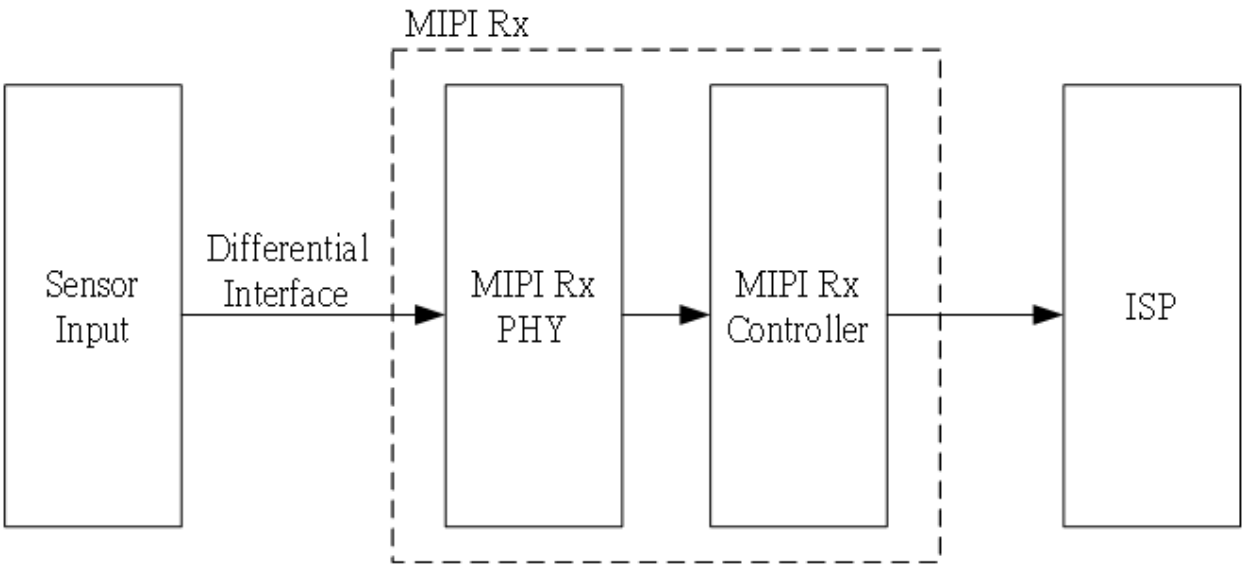


Diagram 19.19: MIPI Rx functional block diagram and location in the system

19.3.2 Features

- Support MIPI DPHY-ver2.1
- Can support 2 sensor inputs at the same time
- A single sensor supports up to 5M (2688x1944, 2880x1620) @30fps HDR or @60fps linear input
- Dual-channel sensor supports maximum FHD(1920x1080) @60fps HDR or linear input
- A single channel supports up to 4-Lane MIPI D-PHY interface and a maximum of 1.5Gbps/Lane
- A single channel supports up to 4-Lane sub-LVDS/HiSPi interfaces and a maximum of 1.5Gbps/Lane
- Supports parsing of RAW8/RAW10/RAW12/RAW16 data types
- Supports YUV422 8-bit / YUV422 10-bit data type parsing
- Supports up to 2 frames of WDR and supports multiple WDR timings
- Supports sub-LVDS/HiSPi mode pixel/sync code big and small endian configuration

- Supports configurable Lane number and Lane order

19.3.3 Function Description

19.3.3.1 Typical Application

In applications using image sensors, the MIPI Rx module register can be set according to different interface selections (MIPI/Sub-LVDS/HiSPi). At the same time, MIPI Rx also supports transmission requirements of different speeds and different resolutions, and is compatible with a variety of images. Sensor format.

The MIPI Rx module contains 1 group of D-PHY, each group has six differential pairs. One D-PHY can support a pair of differential clocks and up to four pairs of data differential signals, or support two groups of one pair of differential clocks at the same time. The upper two pairs of data differential signals, so MIPI Rx can support 2 Sensor inputs at the same time. In addition, MIPI Rx can support different differential pair sequencing and clock differential pair positions. The source of the clock and the differential pair sequencing method can be configured through registers.

MIPI Rx only targets the timing conversion and decoding of the interface, and does not handle the image processing part. Therefore, any resolution and frame rate can be supported under the premise of meeting the bandwidth. The bandwidth of MIPI Rx is limited by two parts: the PHY's interface data rate and the internal processing speed. The input interface supports a maximum of 1.5Gbps/Lane, and the internal processing speed is a maximum of 600M*1pixels/s.

Table 19.214: MIPI Rx Interface Type

	Common mode voltage	Differential mode voltage	Maximum clock frequency	Maximum data rate per lane
MIPI DPHY	200mV	200mV	750MHz	1.5Gbps
Sub-LVDS	900mV	150mV	750MHz	1.5Gbps
HiS Pi(HiVCM)	900mV	280mV	750MHz	1.5Gbps
HiS Pi(SLVDS)	200mV	200mV	750MHz	1.5Gbps

19.3.3.2 MIPI Interface Data Format

MIPI specifications are developed and maintained by different working groups, corresponding to applications in different fields. MIPI Rx supports D-PHY and CSI-2 (Camera Serial Interface). D-PHY specifies the transmission specification of the physical layer, and CSI-2 specifies the format and protocol of the Camera output data packet.

- D-PHY

  D-PHY is a high-speed physical layer standard released by the MIPI Alliance, which specifies the physical characteristics and transmission protocols of the interface layer. D-PHY uses 200mV source-synchronous low-voltage differential signaling technology, and the data green rate range of each Lane supports up to 2500Mbps. D-PHY can operate in two modes: Low Power (LP) and High Speed (HS).

- CSI-2

  CSI-2 is a data protocol for cameras, which specifies the data packet format for communication between the host and peripherals.

  CSI-2 can support image applications with different pixel formats, and the minimum granularity of data transmission is bytes. To increase the performance of CSI-2, you can choose the number of data Lanes. The CSI-2 protocol specifies the mechanism for the sender to pack pixel data into bytes, and specifies how multiple data Lanes are allocated and managed. Bytes of data are organized in packets, which are transmitted between SoT and EoT. The receiving end parses the corresponding data packet according to the protocol and recovers the original pixel data.

MIPI Rx supports parsing of RAW8/RAW10/RAW12/RAW16/YUV422-8bit/YUV422-10bit data types.

CSI-2 data packets are divided into long packets and short packets, which contain check codes and can perform error correction and error detection.

Both long packets and short packets are transmitted between SoT and EoT. During the gap between data transmission, D-PHY is in LP mode.

The transmission mechanism of CSI-2 packets is shown in the figure. PH and PF stand for Packet Header and Packet Footer respectively.

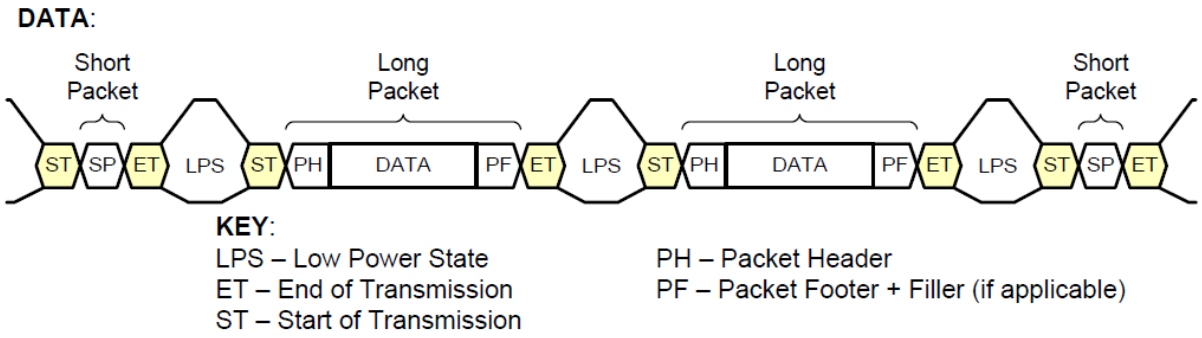


Diagram 19.20: Data packet transmission mechanism

Long packets are used to transmit valid pixel data and are divided into five parts: Data ID, Word Count, ECC, PAYLOAD, CHECKSUM.

- Data ID contains Virtual Channel and Data Type. Virtual Channel controls the channel used for transmission and can use different channels to transmit different data. Data Type specifies the type of data to be transmitted.
- Word Count indicates the amount of data that the receiving end needs to receive.
- ECC is an error correction code that can correct or detect errors in Data Type and Word Count.
- PAYLOAD is the actual pixel data that needs to be transmitted.
- CHECKSUM is a checksum generated by a linear feedback shift register and is used to verify PAYLOAD data.

The structure of the long package is shown in the diagram *CSI-2 long packet format*.

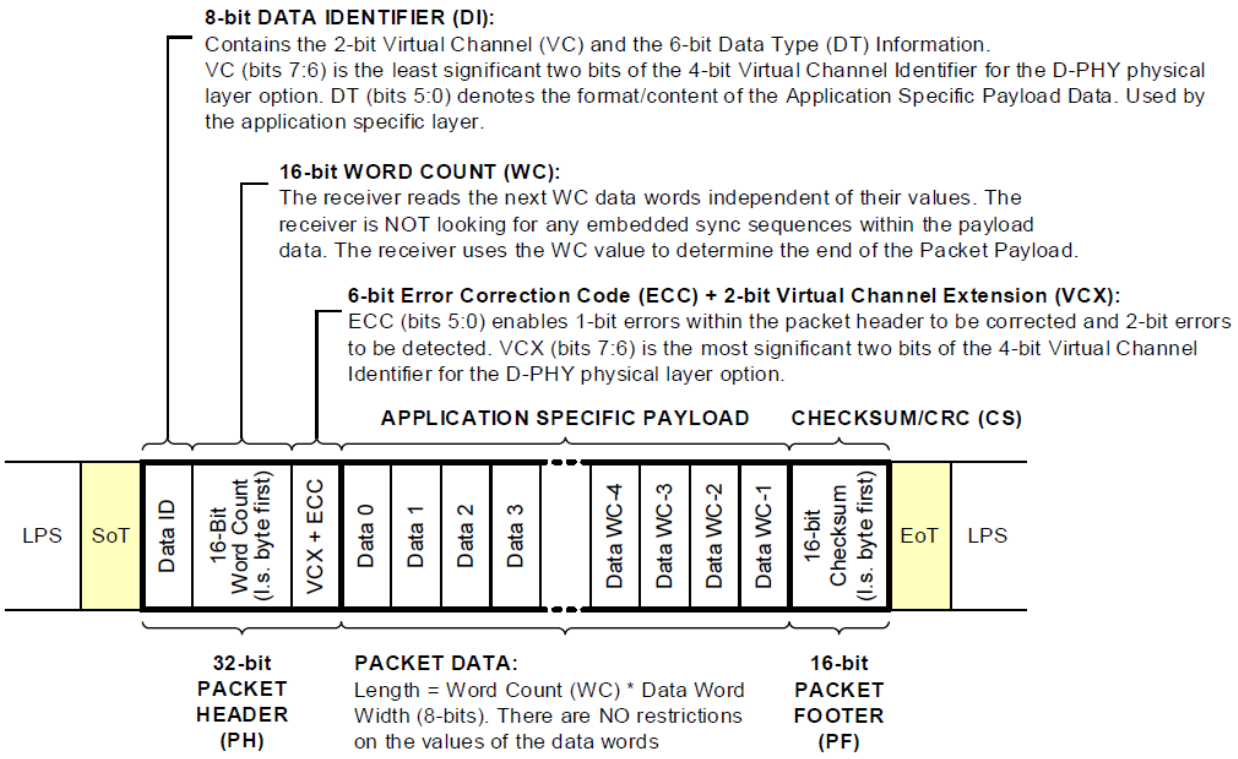


Diagram 19.21: CSI-2 long packet format

The function of the short packet is to transmit information synchronously, including three parts: Data ID, Data Field and ECC. Its format is shown in the diagram *CSI-2 short packet format*.

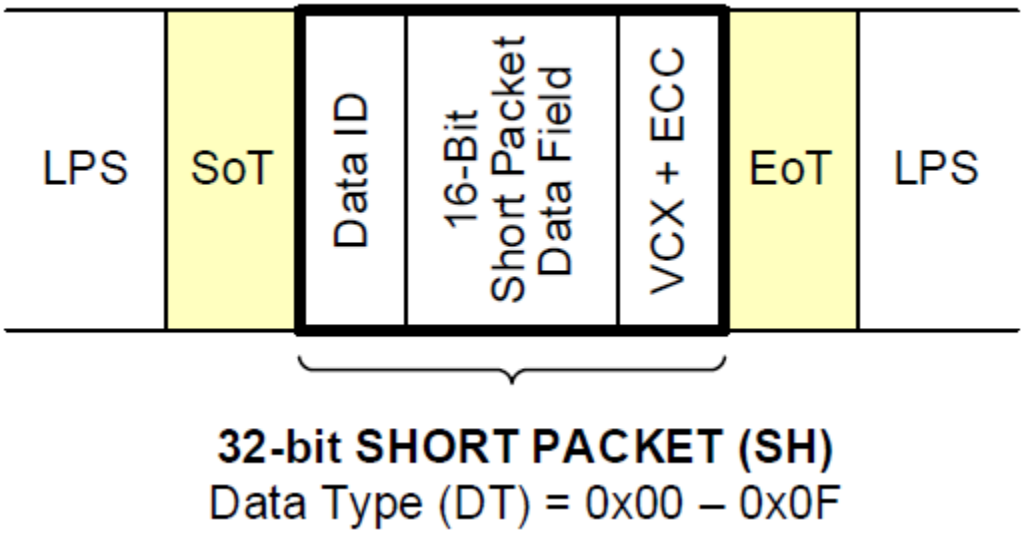


Diagram 19.22: CSI-2 short packet format

MIPI Rx supports the transmission of six video data formats, including YUV422-8bit, YUV422-10bit, RAW8, RAW10, RAW12 and RAW16. The transmission methods of different data formats are described below.

The transmission mode of YUV422-8bit is in the form of UYVY, as shown in the diagram *YUV422 8-bit data transfer sequence*.

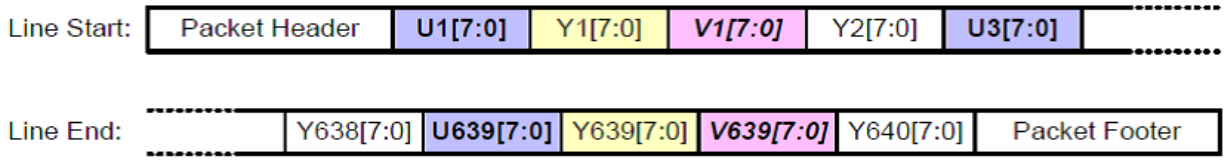


Diagram 19.23: YUV422 8-bit data transfer sequence

The correspondence between the packet and the video signal is shown in the diagram *YUV422-8bit data packet transmission mapping*.

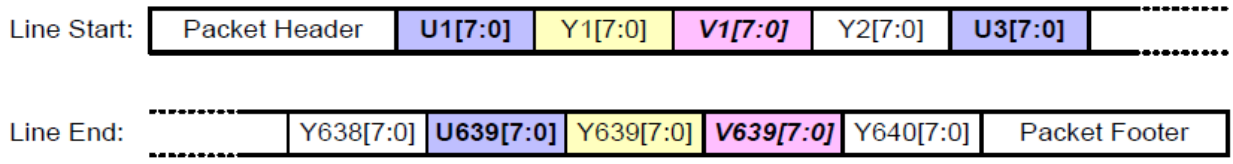


Figure 109 YUV422 8-bit Transmission

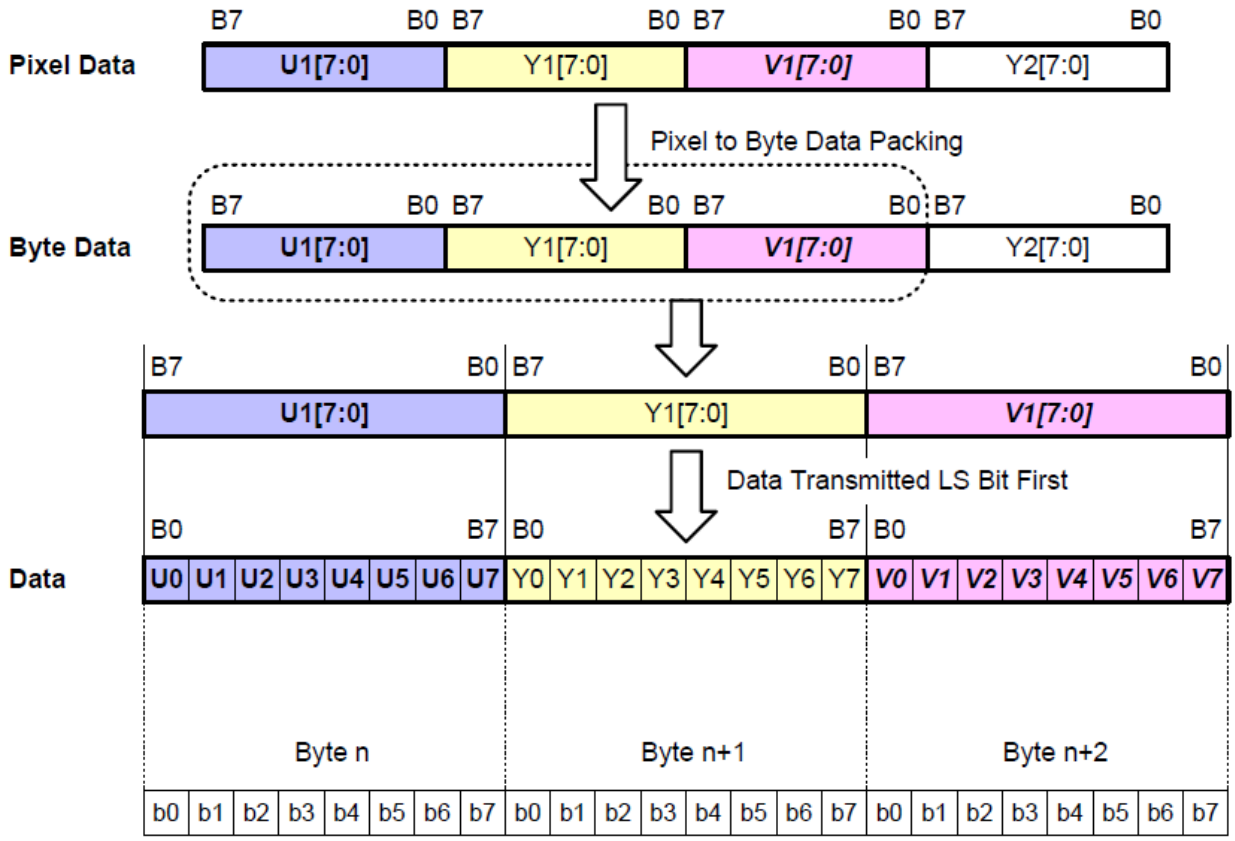


Diagram 19.24: YUV422-8bit data packet transmission mapping

The transmission format of the entire Frame will be as shown in the diagram *YUV422 8-bit frame format*.

Diagram 19.25: YUV422 8-bit frame format

The transmission mode of YUV422-10bit is also UYVY, and the transmission sequence is shown in the chart *YUV422-10bit data transfer sequence*.

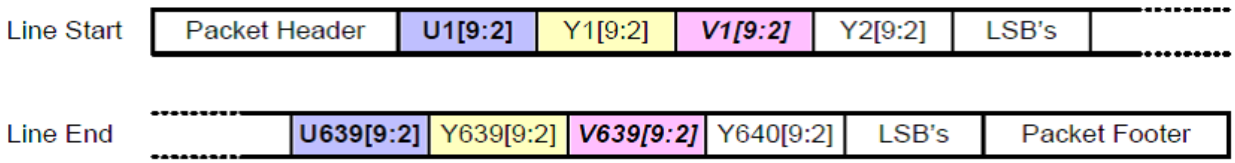


Diagram 19.26: YUV422-10bit data transfer sequence

The correspondence between the packet and the video signal is shown in the diagram *YUV422-10bit data packet transmission mapping*.

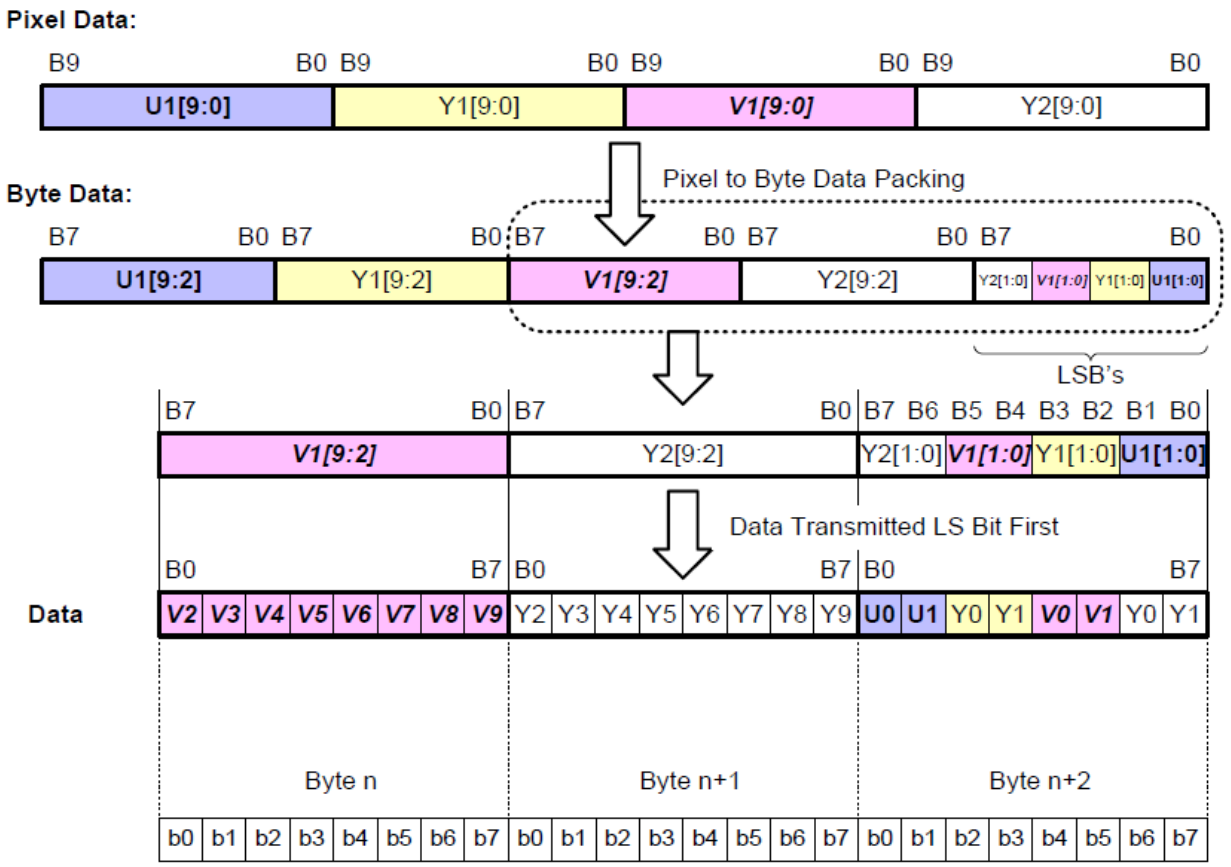


Diagram 19.27: YUV422-10bit data packet transmission mapping

The transmission format of the entire Frame will be as shown in the diagram *YUV422-10bit frame format*.

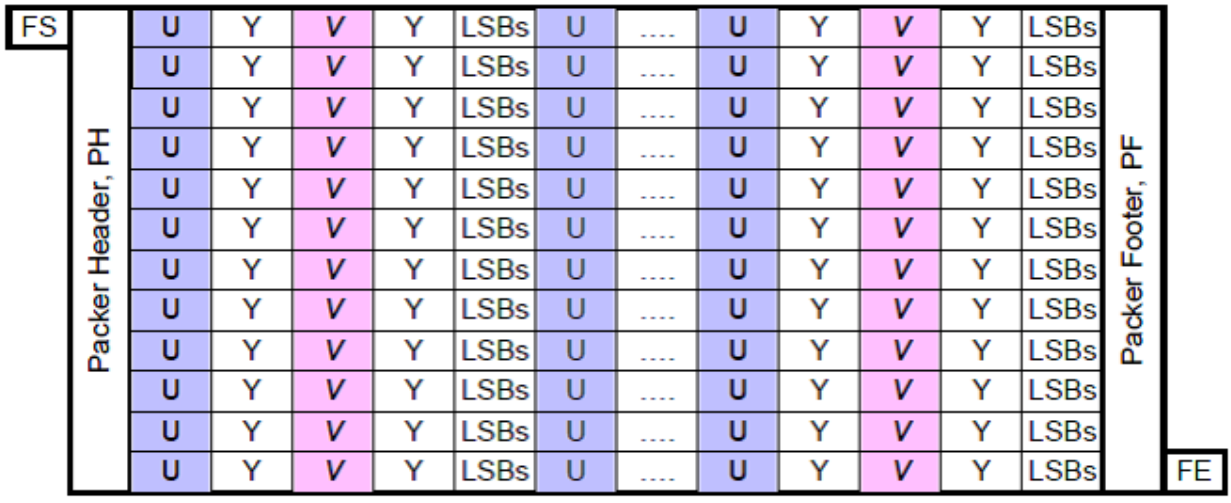


Diagram 19.28: YUV422-10bit frame format

The transfer sequence of RAW8 is shown in the diagram *RAW8 data transfer sequence*.

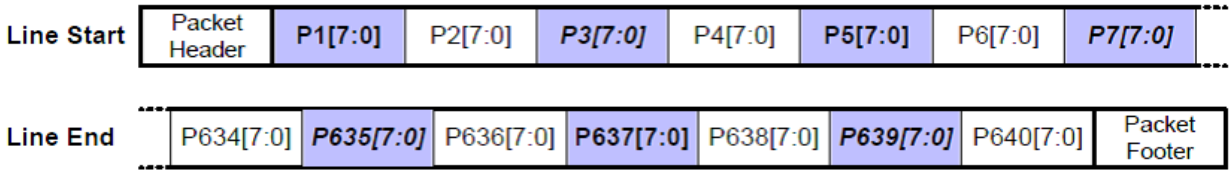


Diagram 19.29: RAW8 data transfer sequence

The transmission format of the entire Frame will be as shown in the diagram *RAW8 frame format*.

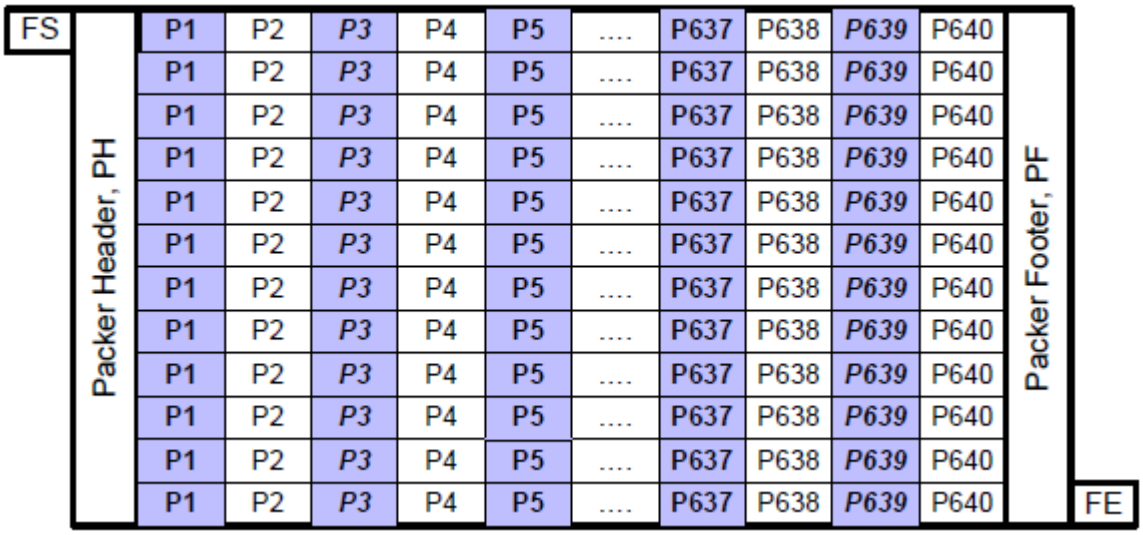


Diagram 19.30: RAW8 frame format

The transfer sequence of RAW10 is shown in the diagram *RAW10 data transfer sequence*.

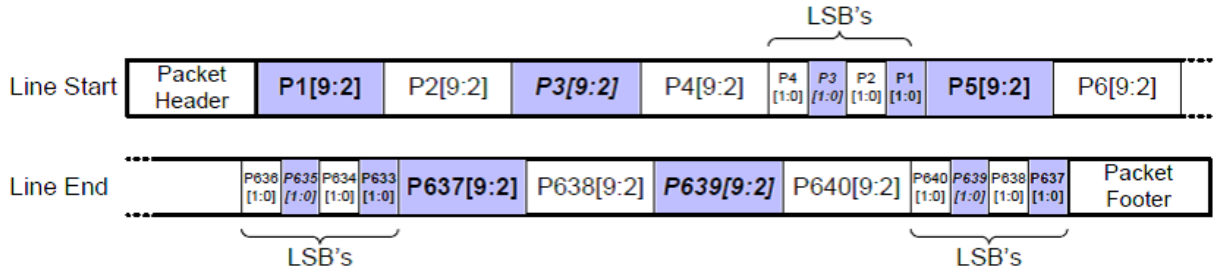


Diagram 19.31: RAW10 data transfer sequence

The transmission format of the entire Frame will be as shown in the diagram *RAW10 frame format*.

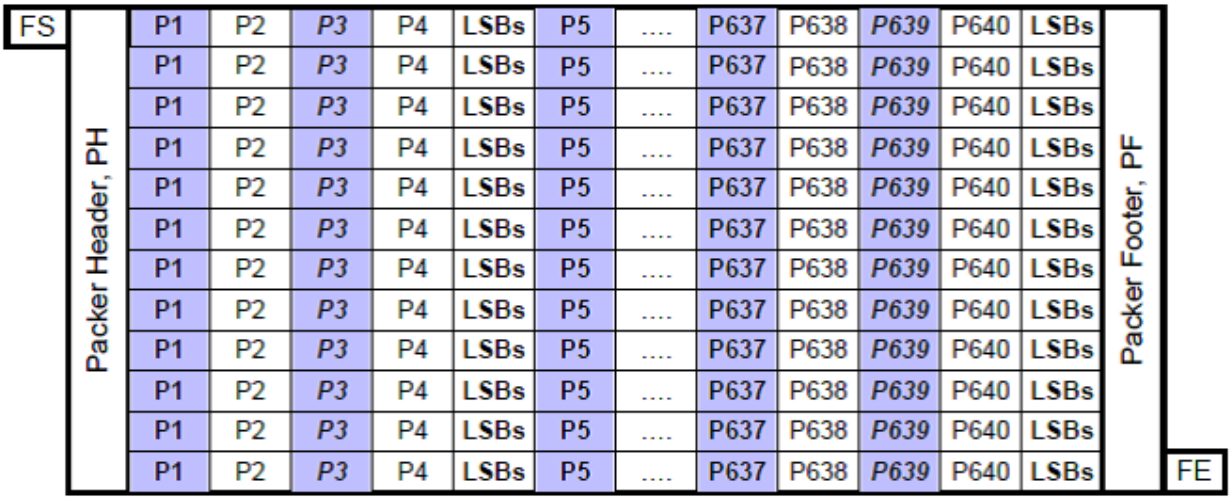


Diagram 19.32: RAW10 frame format

The transfer sequence of RAW12 is shown in the diagram *RAW12 data transfer sequence*.

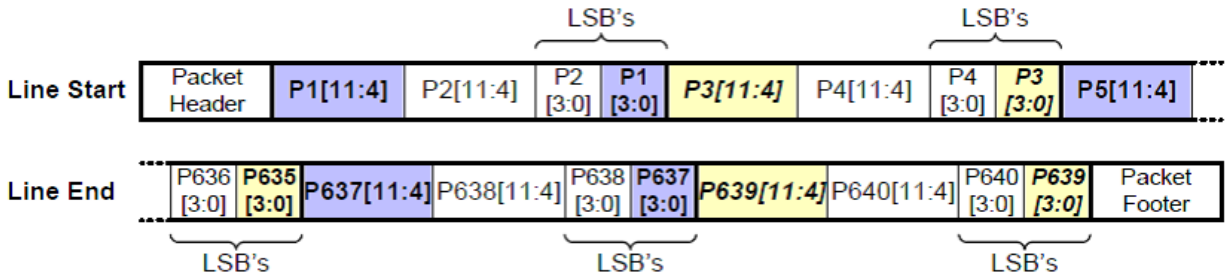


Diagram 19.33: RAW12 data transfer sequence

The transmission format of the entire Frame will be as shown in the diagram *RAW12 frame format*.

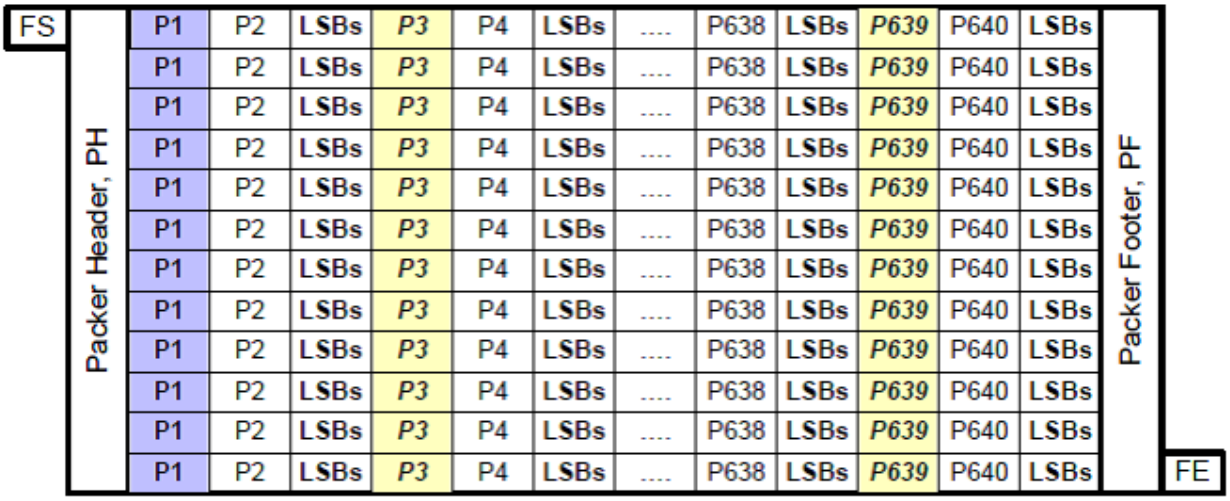


Diagram 19.34: RAW12 frame format

The transfer sequence of RAW16 is shown in the diagram *RAW16 data transfer sequence*.

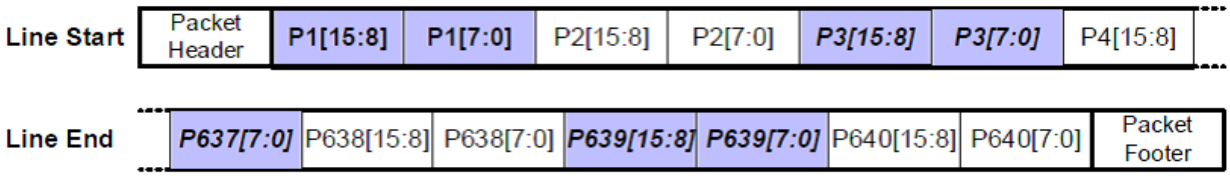


Diagram 19.35: RAW16 data transfer sequence

The transmission format of the entire Frame will be as shown in the diagram *RAW16 frame format*.

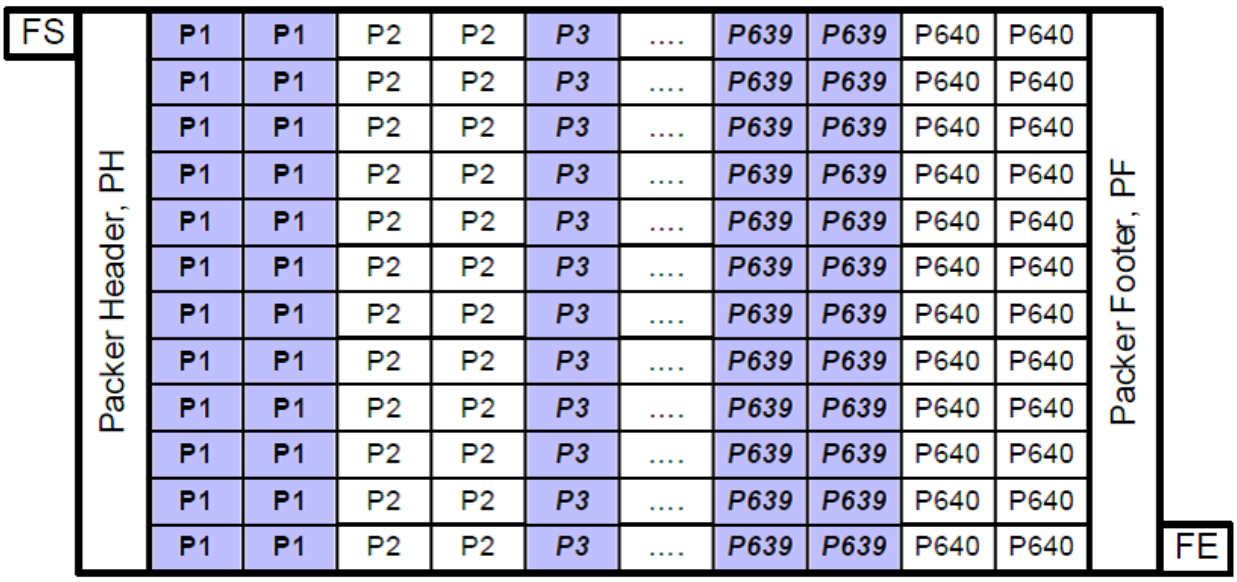


Diagram 19.36: RAW16 frame format

19.3.3.3 MIPI Interface Linear Mode

The linear mode transmission format of the MIPI interface is shown in the diagram *MIPI interface image format*. The transmission of each picture starts with the short packet Frame Start (FS) and ends with the short packet Frame End (FE). The video content in the middle is in line units, and each long packet transmits a complete video line. The long packet format is as specified by the MIPI standard. Each line has a 32-bit data packet header (PH, Pecket Header), which contains information such as the Virtual Channel and Data Type of the current line.

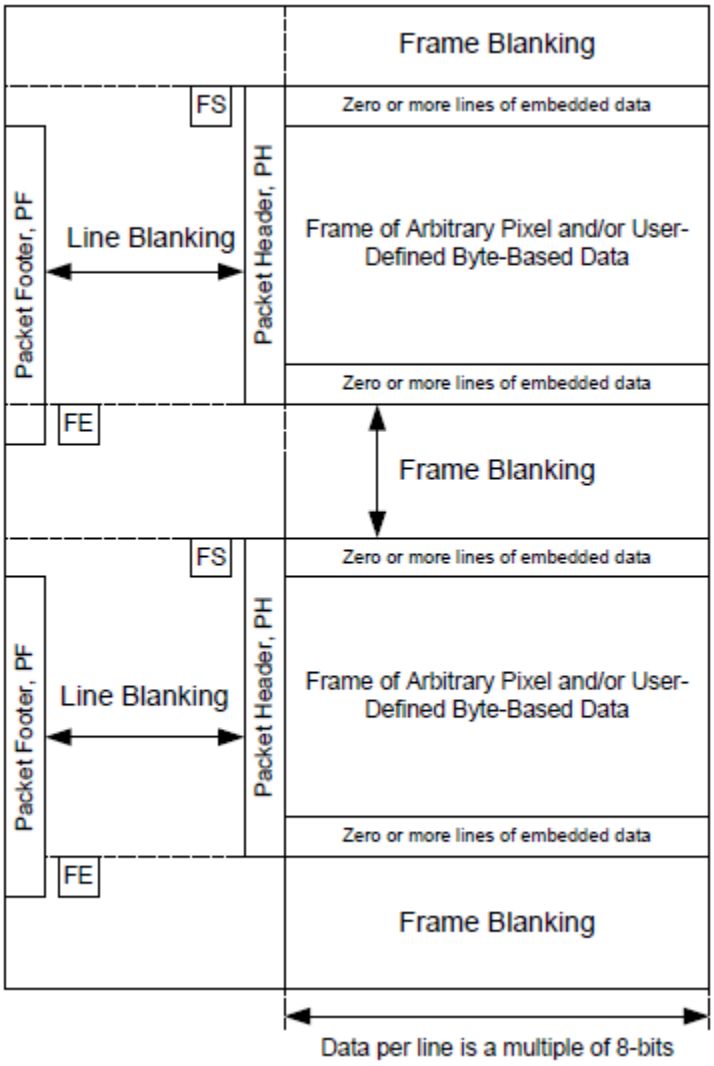


Diagram 19.37: MIPI interface image format

19.3.3.4 MIPI interface wide dynamic mode

MIPI Rx supports four wide dynamic range (WDR) modes of the MIPI interface, namely:

1. Use DT (Data Type) to distinguish long and short exposure data
2. Use the identification code (Identification Code) to distinguish long and short exposure data
3. Use registers to set long and short exposure data delay intervals

The WDR transmission method using DT is shown in the chart *MIPI interface wide dynamic data transmission (using DT)*. Different exposure lengths share a set of FS/FE short packets, and the header of the long packet contains DT information. Different DTs can be used to distinguish long and short exposure data. , the real data format DT and the two sets of DT representing long and short exposure data can be set using registers, and MIPI Rx can parse out the correct wide dynamic timing and send it to the rear video processing module.

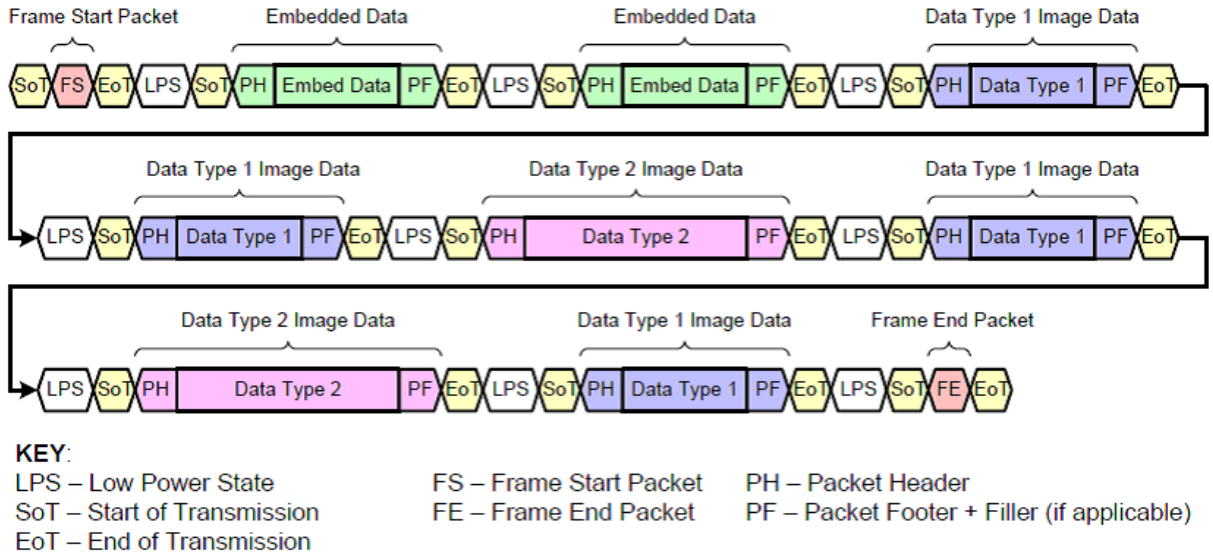


Diagram 19.38: MIPI interface wide dynamic data transmission (using DT)

The WDR transmission method using ID is shown in the chart *MIPI interface wide dynamic data transmission (using ID)*. Different exposure lengths share a set of FS/FE short packets, and the first four pixels of each long packet in the transmission data are used to transmit the data representing different exposure lengths. ID (Identification Code), the ID representing long and short exposures can be set using registers. MIPI Rx will use the ID to distinguish different exposure video signals, and remove the first four pixels before sending them to the video processing module.

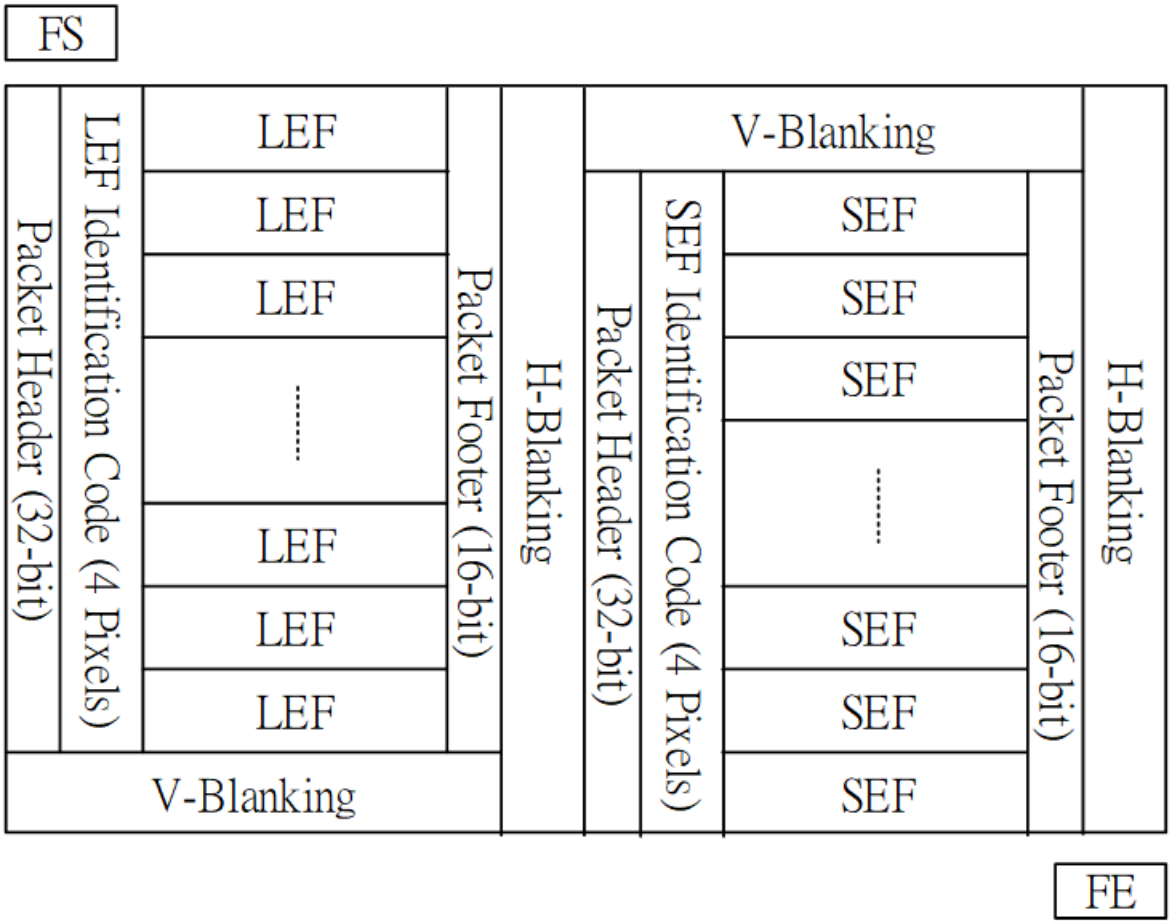


Diagram 19.39: MIPI interface wide dynamic data transmission (using ID)

The last supported WDR transmission method does not have any DT or ID to indicate whether the transmitted long packet contains long exposure or short exposure content. The user must set the register to indicate the number of exposure lines between long exposure and short exposure. Difference, MIPI Rx will parse the corresponding timing to the video processing module. The actual transmission timing is shown in the chart *MIPI interface wide dynamic data transmission (register setting)*.

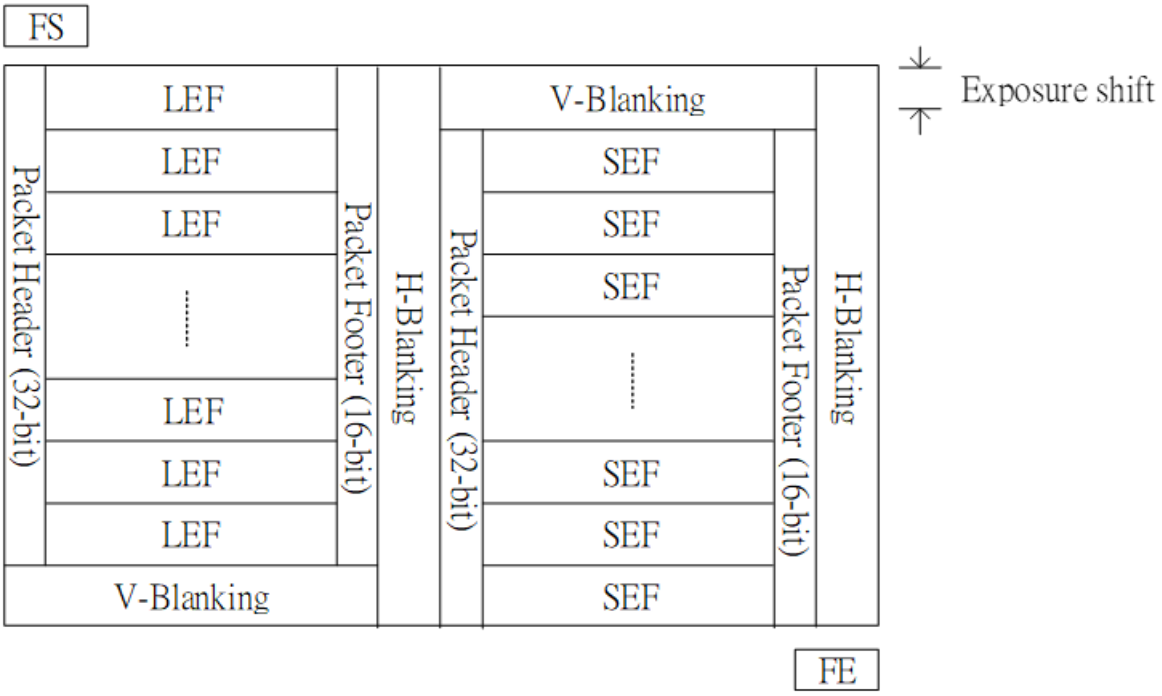


Diagram 19.40: MIPI interface wide dynamic data transmission (register setting)

19.3.3.5 Sub-LVDS interface data format

Ultra-low voltage differential signal sub-LVDS (Low-Voltage Differential Signal) is commonly used in front-end cameras. It uses synchronization codes to distinguish the range of valid video signals and the long and short exposures of wide dynamic mode.

The PHY of MIPI Rx converts differential serial data into parallel data, and then the controller of MIPI Rx decodes the parallel data into pixel data according to different modes and synchronization codes.

MIPI Rx supports three bit width Sub-LVDS transmission modes of 8bit, 10bit and 12bit. The interface data format is shown in the chart *Sub-LVDS interface data format*. All valid video signals will be in the middle of SAV and EAV synchronization codes, where synchronization Codes are composed of four fields, and the bit width of each field is the same as the following pixel bit width. The first three fields are fixed reference code words, and the fourth field can be used to distinguish the start or end of the valid interval. The Sub-LVDS synchronization code format is shown in the chart *Sample of Sub-LVDS Sync Code*. The synchronization code will use different values according to different manufacturers. The chart *Sample of Sub-LVDS Sync Code* is just one of the implementation methods. Different values can be in the register set up.

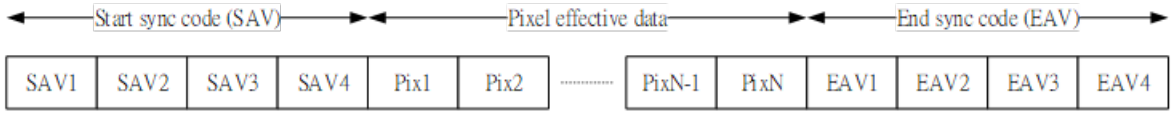


Diagram 19.41: Sub-LVDS interface data format

Table 19.215: Sample of Sub-LVDS Sync Code

12-bit					
		1st word	2nd word	3rd word	4th word
Blanking line	Start sync code (SAV)	FFFh	000h	000h	AB0h
	End sync code (EAV)				B60h
Effective line	Start sync code (SAV)				800h
	End sync code (EAV)				9D0h
10-bit					
		1st word	2nd word	3rd word	4th word
Blanking line	Start sync code (SAV)	3FFh	000h	000h	2ACh
	End sync code (EAV)				2D8h
Effective line	Start sync code (SAV)				200h
	End sync code (EAV)				274h
8-bit					
		1st word	2nd word	3rd word	4th word
Blanking line	Start sync code (SAV)	FFh	00h	00h	ABh
	End sync code (EAV)				B6h
Effective line	Start sync code (SAV)				80h
	End sync code (EAV)				9Dh

The transmission mode of Sub-LVDS synchronization code and pixel information in different Lanes is shown in the chart *Sub-LVDS Multi Lane Transmission mode*. Each Lane will transmit the same synchronization code, followed by pixel data. The pixel data will be transferred according to the used The number of channels is arranged sequentially in units of pixel width.

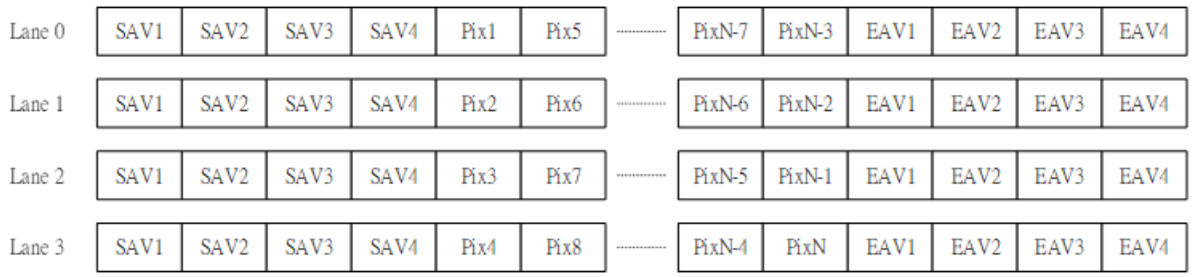


Diagram 19.42: Sub-LVDS Multi Lane Transmission mode

The synchronization code and pixel data in Sub-LVDS are serial, while MIPI Rx supports the big and small ends of the data and can be set using registers. Taking the big endian mode as an example, the timing of outputting a single pixel is shown in the chart *Sub-LVDS single pixel timing diagram*.

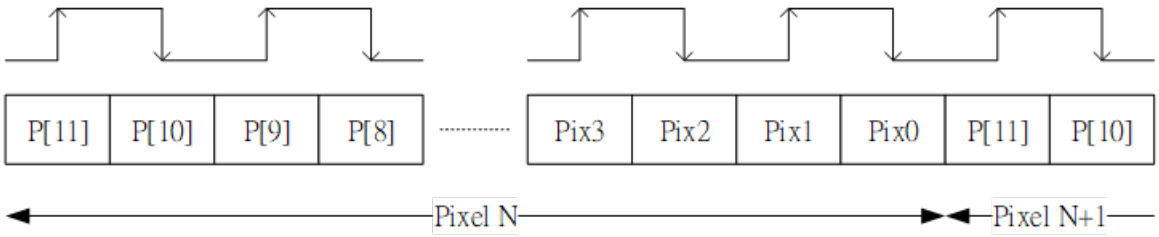


Diagram 19.43: Sub-LVDS single pixel timing diagram

19.3.3.6 Sub-LVDS interface Linear Mode

In linear mode, Sub-LVDS uses synchronization codes to mark the start and end of each line in an image data, and anything other than the synchronization codes SAV and EAV is not valid video data, as shown in the chart *Sub-LVDS Linear Mode Timing Diagram*.

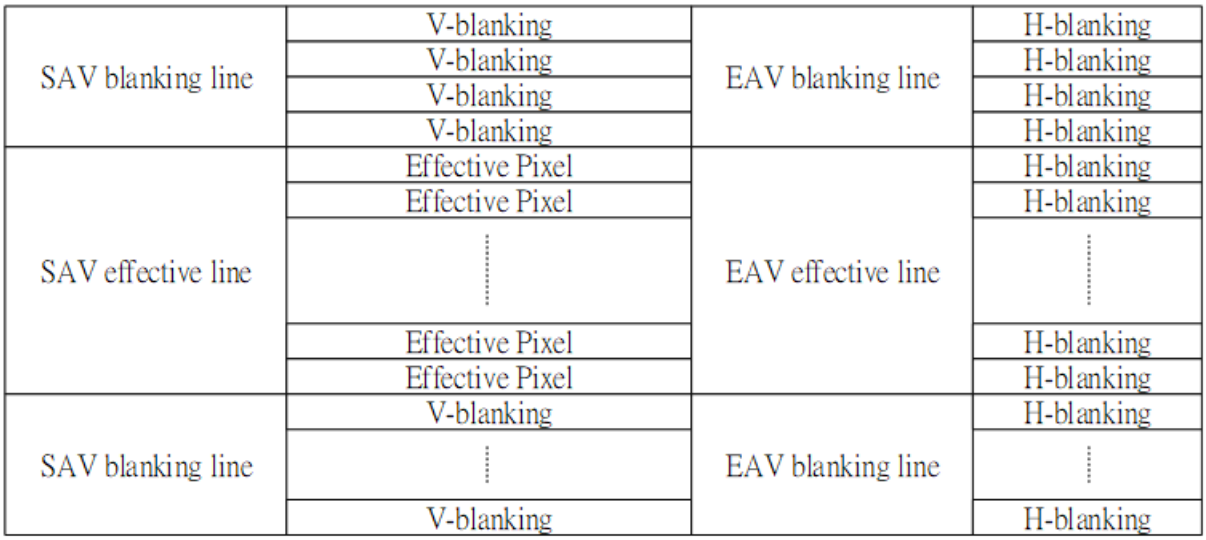


Diagram 19.44: Sub-LVDS Linear Mode Timing Diagram

19.3.3.7 Sub-LVDS Interface Wide Dynamic Mode

MIPI Rx can support two Sub-LVDS interface wide dynamic modes. In the first mode, as shown in the diagram *Sub-LVDS Wide dynamic mode I*, the long and short exposure video signals are wrapped in SAV and EAV synchronization codes respectively. MIPI Rx can use different The synchronization code analyzes whether the video signal is a long exposure or a short exposure. The second mode is as shown in the diagram *Sub-LVDS Wide dynamic mode II*. The long exposure and short exposure are wrapped in the same set of SAV and EAV. The width and blanking length of each line must be set in the register. MIPI Rx must use these registers. The settings and synchronization codes are used to analyze the timing of long exposure and short exposure, and then sent to the video processing module.

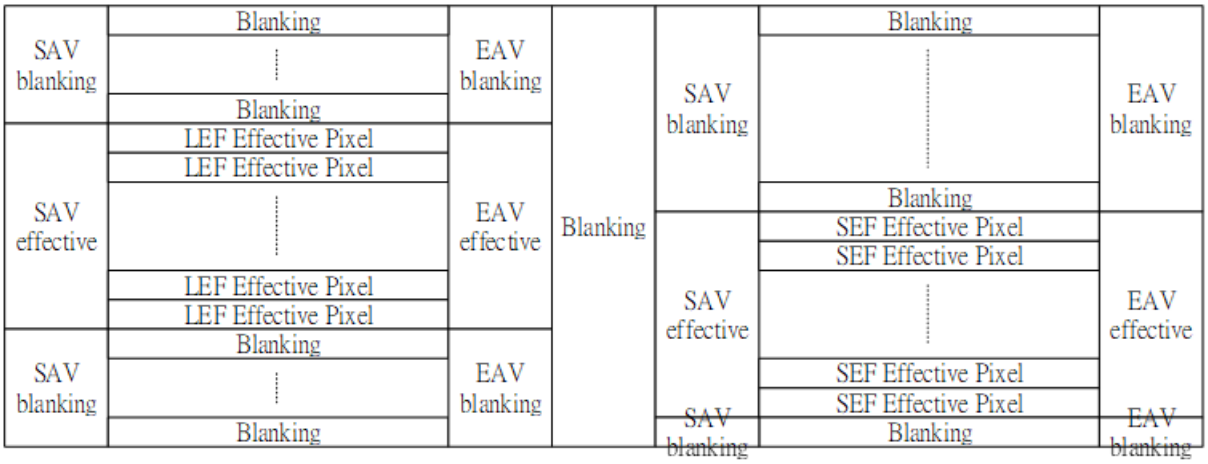


Diagram 19.45: Sub-LVDS Wide dynamic mode I

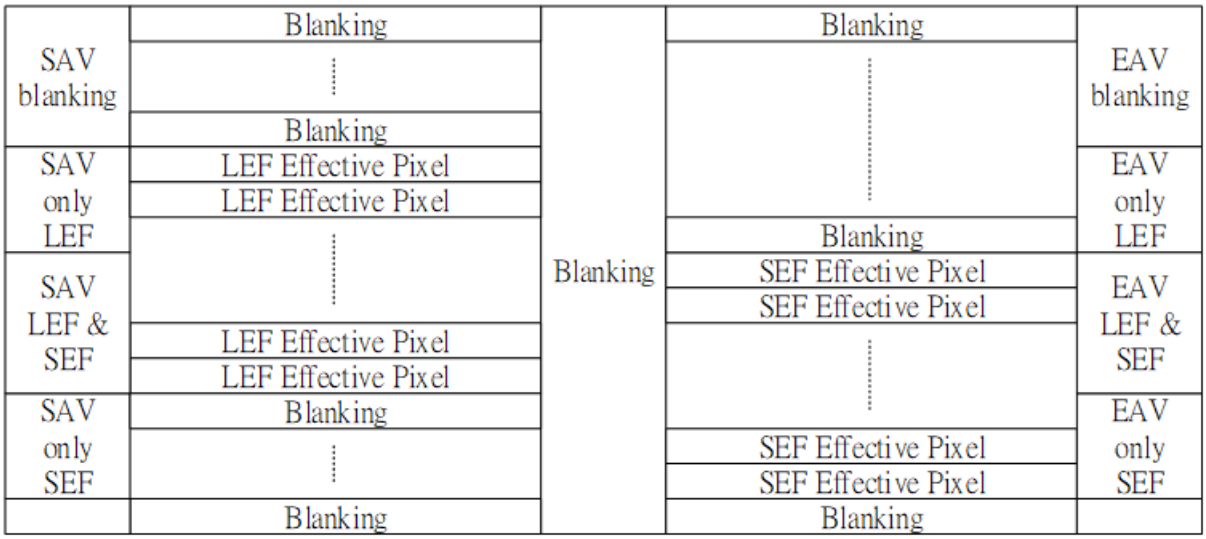


Diagram 19.46: Sub-LVDS Wide dynamic mode II

19.3.3.8 HiSPi Interface Data Format

The High-Speed Serial Pixel (HiSPi) interface is also used in some cameras. Similar to Sub-LVDS, it uses synchronization codes to distinguish valid video information and distinguish long and short exposures in wide dynamic mode. The HiSPi specification defines four different packaging modes, namely Packetized-SP, Streaming-SP, Streaming-S and ActiveStart-SP8.

MIPI Rx supports two of the more common transmission methods, Packetized-SP and Streaming-SP.

19.3.3.9 HiSPi Interface Linear Mode

MIPI Rx supports two different HiSPi modes. In Packetized-SP mode, as shown in the diagram *HiSPi Packetized-SP mode*, the image sensor uses SOF to represent the first line of the valid video signal, and uses EOF to represent the end of the last line of the valid video signal. Other valid video signals use SOL and EOL as the start and end of a line.

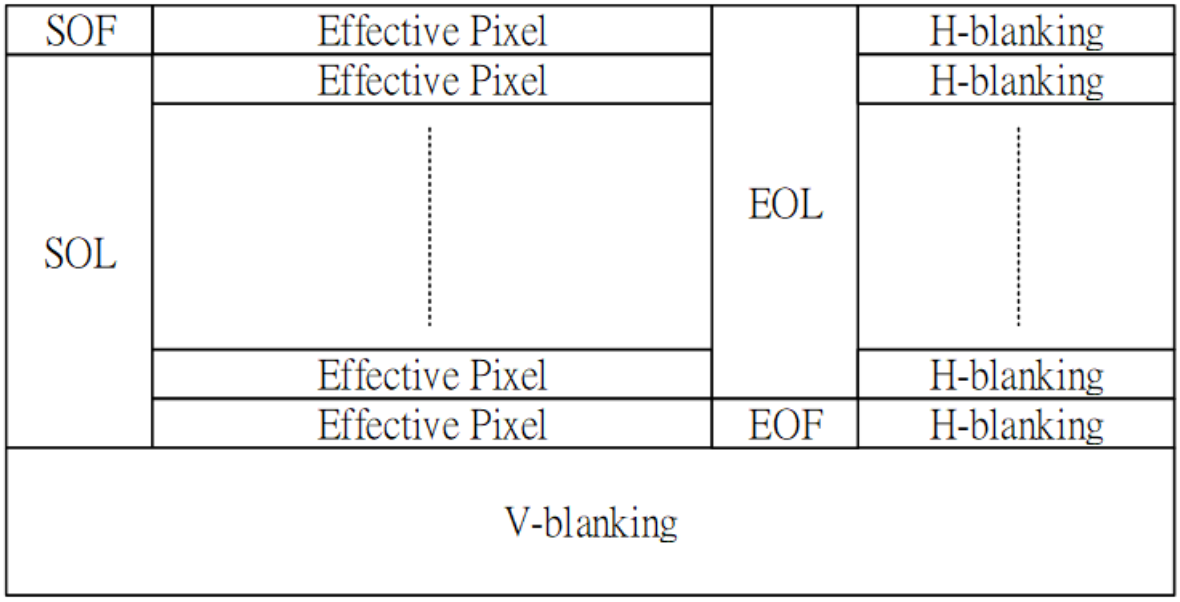


Diagram 19.47: HiSPi Packetized-SP mode

In Streaming-SP mode, as shown in the diagram *HiSPi Streaming-SP mode*, the image sensor does not transmit the EOL or EOF that represents the end, so the MIPI Rx controller must use the register settings to know the number of valid video signals before it can be parsed. The correct video signal is sent to the video processing module (ISP). In addition, the Streaming-SP mode also supports the SAV signal indicating the number of blank lines. The synchronization codes supported by the two different transmission methods are organized as shown in the chart *HiSPi Sync Code Support Mode*.

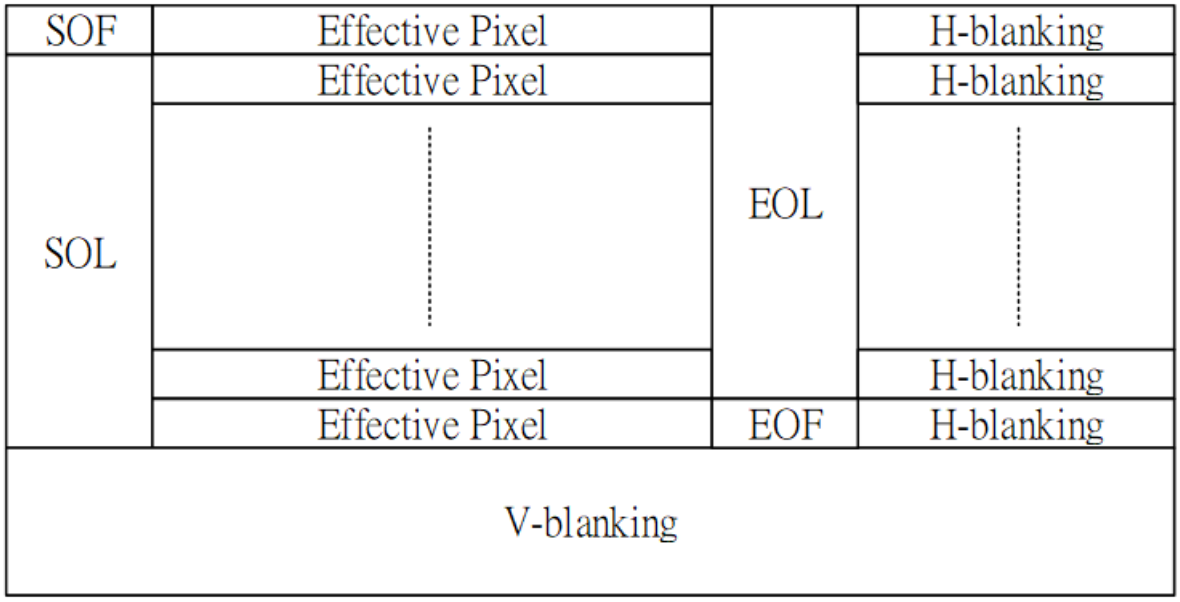


Diagram 19.48: HiSPi Streaming-SP mode

Table 19.216: HiSPi Sync Code Support Mode

Sync Code	Packetized-SP	Streaming-SP
SOF	Required	Required
SOL	Required	Required
EOF	Required	Unsupported
EOL	Required	Unsupported
SAV	Unsupported	Required

19.3.3.10 HiSPi Interface Wide Dynamic Mode

HiSPi interface wide dynamic mode is also divided into two different modes. The first Packetized-SP is shown in the chart *HiSPi Packetized-SP wide dynamic transmission*. Long exposure and short exposure will be distinguished by different synchronization codes, among which SOF_L and EOF_L The ones in are valid long exposure video signals, and the ones in SOF_S and EOF_S are the valid short exposure video signals. The last few lines of long exposure and the first few lines of short exposure are not valid pixel areas, but are filled with fixed values.

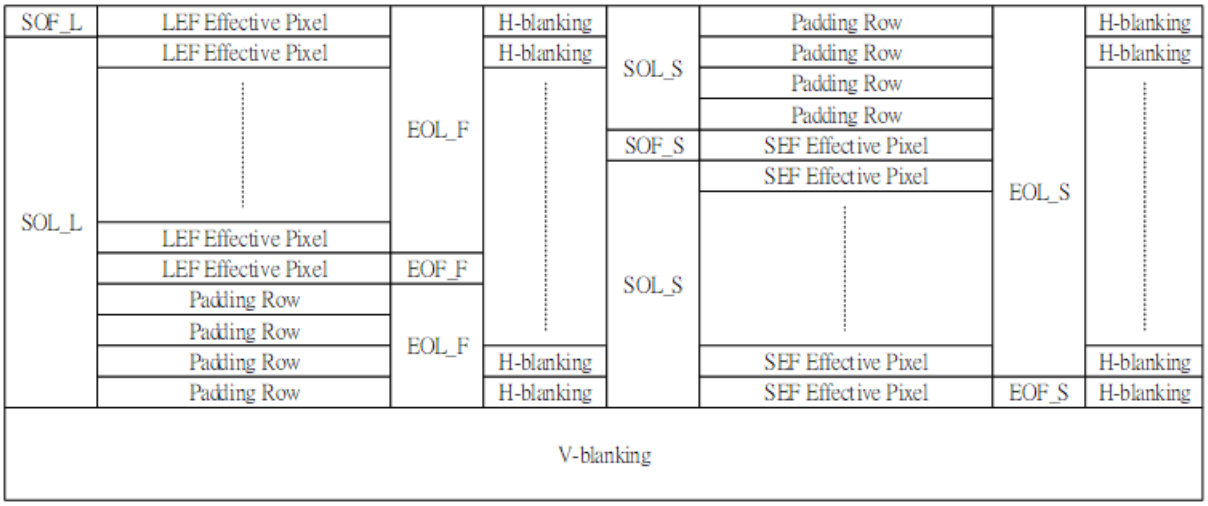


Diagram 19.49: HiSPi Packetized-SP wide dynamic transmission

The second type of wide dynamic transfer of Streaming-SP is shown in the chart *HiSPi Streaming-SP wide dynamic transmission*. The synchronization code and linear mode of long exposure and short exposure are the same, so a register is needed to set the exposure line between long exposure and short exposure. Only with this digital gap can MIPI Rx parse out the correct wide dynamic video signal.

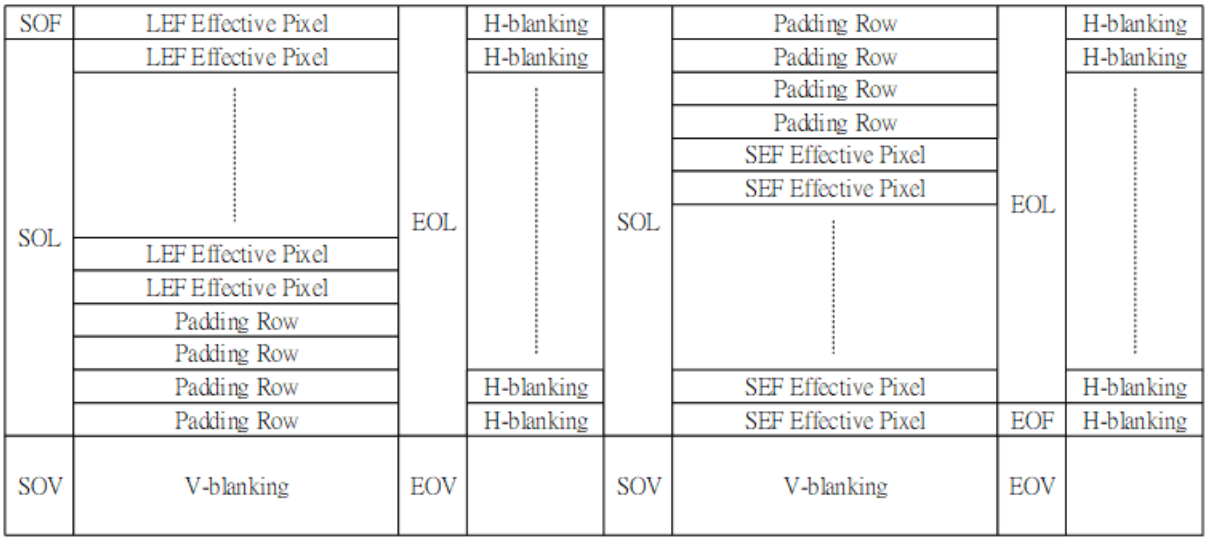


Diagram 19.50: HiSPi Streaming-SP wide dynamic transmission

19.3.4 MIPI Rx Register Overview

Up to two sets of MIPI Rx modules can be used in the chip at the same time, which are mainly divided into three sets of registers. The first part is the register that controls the PHY module, with a base address of 0x0A0D0000. The second part is a register that controls the CSI module, with a base address of 0x0A0C2400 and 0x0A0C4400. The third part is the register that controls the Sub-LVDS and HiSPi modules. The base addresses are 0x0A0C2200 and 0x0A0C4200.

Table 19.217: MIPI Rx PHY Registers Overview, Base Address: 0x0A0D0000

Name	Address Offset	Description
REG_00	0x000	PD_CTRL
REG_04	0x004	ANALOG_CTRL
REG_30	0x030	SENSOR_MODE_CTRL
REG_34	0x034	ANALOG_CAL_0
REG_38	0x038	ANALOG_CAL_1
REG_3C	0x03c	ANALOG_CAL_2
REG_40	0x040	ANALOG_CAL_3
REG_44	0x044	ANALOG_CAL_4
REG_48	0x048	ANALOG_CAL_5
REG_80	0x080	CLOCK_INVERCE_CTL
REG_A0	0x0a0	
REG_A4	0x0a4	
REG_A8	0x0a8	
REG_AC	0x0ac	

Table 19.218: MIPI Rx PHY Registers Overview, Base Address: 0x0A0D0300

Name	Address Offset	Description
REG_00	0x000	SENSOR_MODE_CTRL

continues on next page

Table 19.218 – continued from previous page

Name	Address Offset	Description
REG_04	0x004	LANE_SWAP_0
REG_08	0x008	LANE_SWAP_1
REG_0C	0x00c	CSI_GLB_CTL_0
REG_20	0x020	SLVDS_CTRL_0
REG_24	0x024	SLVDS_CTRL_1
REG_D0_0	0x100	D0_REG_CTRL_CALIB_0
REG_D0_1	0x104	D0_REG_CTRL_CALIB_1
REG_D0_3	0x10c	D0_CALIB_RESULT_0
REG_D0_4	0x110	D0_CALIB_RESULT_1
REG_D0_5	0x114	D0_CALIB_RESULT_2
REG_D0_6	0x118	D0_CALIB_RESULT_3
REG_D0_7	0x11c	D0_CALIB_RESULT_4
REG_D0_8	0x120	D0_CALIB_RESULT_5
REG_D0_9	0x124	D0_CALIB_RESULT_6
REG_D0_A	0x128	D0_CALIB_RESULT_7
REG_D1_0	0x140	D1_REG_CTRL_CALIB_0
REG_D1_1	0x144	D1_REG_CTRL_CALIB_1
REG_D1_3	0x14c	D1_CALIB_RESULT_0
REG_D1_4	0x150	D1_CALIB_RESULT_1
REG_D1_5	0x154	D1_CALIB_RESULT_2
REG_D1_6	0x158	D1_CALIB_RESULT_3
REG_D1_7	0x15c	D1_CALIB_RESULT_4
REG_D1_8	0x160	D1_CALIB_RESULT_5
REG_D1_9	0x164	D1_CALIB_RESULT_6
REG_D1_A	0x168	D1_CALIB_RESULT_7
REG_D2_0	0x180	D2_REG_CTRL_CALIB_0
REG_D2_1	0x184	D2_REG_CTRL_CALIB_1
REG_D2_3	0x18c	D2_CALIB_RESULT_0
REG_D2_4	0x190	D2_CALIB_RESULT_1
REG_D2_5	0x194	D2_CALIB_RESULT_2
REG_D2_6	0x198	D2_CALIB_RESULT_3
REG_D2_7	0x19c	D2_CALIB_RESULT_4
REG_D2_8	0x1a0	D2_CALIB_RESULT_5
REG_D2_9	0x1a4	D2_CALIB_RESULT_6
REG_D2_A	0x1a8	D2_CALIB_RESULT_7
REG_D3_0	0x1c0	D3_REG_CTRL_CALIB_0
REG_D3_1	0x1c4	D3_REG_CTRL_CALIB_1
REG_D3_3	0x1cc	D3_CALIB_RESULT_0
REG_D3_4	0x1d0	D3_CALIB_RESULT_1
REG_D3_5	0x1d4	D3_CALIB_RESULT_2
REG_D3_6	0x1d8	D3_CALIB_RESULT_3
REG_D3_7	0x1dc	D3_CALIB_RESULT_4
REG_D3_8	0x1e0	D3_CALIB_RESULT_5
REG_D3_9	0x1e4	D3_CALIB_RESULT_6
REG_D3_A	0x1e8	D3_CALIB_RESULT_7

Table 19.219: MIPI Rx PHY Registers Overview, Base Address: 0x0A0D0600

Name	Address Offset	Description
REG_00	0x000	SENSOR_MODE_CTRL
REG_04	0x004	LANE_SWAP_0
REG_08	0x008	LANE_SWAP_1
REG_0C	0x00c	CSI_GLB_CTL_0
REG_20	0x020	SLVDS_CTRL_0
REG_D0_0	0x100	D0_REG_CTRL_CALIB_0
REG_D0_1	0x104	D0_REG_CTRL_CALIB_1
REG_D0_3	0x10c	D0_CALIB_RESULT_0
REG_D0_4	0x110	D0_CALIB_RESULT_1
REG_D0_5	0x114	D0_CALIB_RESULT_2
REG_D0_6	0x118	D0_CALIB_RESULT_3
REG_D0_7	0x11c	D0_CALIB_RESULT_4
REG_D0_8	0x120	D0_CALIB_RESULT_5
REG_D0_9	0x124	D0_CALIB_RESULT_6
REG_D0_A	0x128	D0_CALIB_RESULT_7
REG_D1_0	0x140	D1_REG_CTRL_CALIB_0
REG_D1_1	0x144	D1_REG_CTRL_CALIB_1
REG_D1_3	0x14c	D1_CALIB_RESULT_0
REG_D1_4	0x150	D1_CALIB_RESULT_1
REG_D1_5	0x154	D1_CALIB_RESULT_2
REG_D1_6	0x158	D1_CALIB_RESULT_3
REG_D1_7	0x15c	D1_CALIB_RESULT_4
REG_D1_8	0x160	D1_CALIB_RESULT_5
REG_D1_9	0x164	D1_CALIB_RESULT_6
REG_D1_A	0x168	D1_CALIB_RESULT_7

Table 19.220: MIPI Rx CSI Control Registers Overview

Name	Address Offset	Description
REG_00	0x000	MODE_CTRL
REG_04	0x004	INTR_CTRL
REG_08	0x008	HDR_CTRL_0
REG_0C	0x00c	HDR_CTRL_1
REG_10	0x010	HDR_CTRL_2
REG_14	0x014	BLC_CTRL
REG_18	0x018	HDR_CTRL_3
REG_1C	0x01c	HDR_CTRL_4
REG_20	0x020	HDR_CTRL_5
REG_24	0x024	HDR_CTRL_6
REG_40	0x040	CSI_STATUS
REG_60	0x060	
REG_70	0x070	CSI_VS_GEN
REG_74	0x074	HDR_DT_CTRL

Table 19.221: MIPI Rx Sub-LVDS Control Registers Overview

Name	Address Offset	Description
REG_00	0x000	MODE_CTRL
REG_04	0x004	SYNC_CODE_0
REG_08	0x008	SYNC_CODE_1
REG_0C	0x00c	SYNC_CODE_2
REG_10	0x010	SYNC_CODE_3
REG_14	0x014	SYNC_CODE_4
REG_18	0x018	SYNC_CODE_5
REG_1C	0x01c	SYNC_CODE_6
REG_20	0x020	SYNC_CODE_7
REG_24	0x024	SYNC_CODE_8
REG_28	0x028	SYNC_CODE_9
REG_2C	0x02c	VS_GEN
REG_30	0x030	LANE_MODE
REG_50	0x050	SYNC_CODE_A
REG_54	0x054	SYNC_CODE_B
REG_58	0x058	HDR_PATTEN_2
REG_60	0x060	HISPI_MODE_CTRL_0
REG_64	0x064	HISPI_MODE_CTRL_1
REG_68	0x068	HISPI_MODE_CTRL_2
REG_6C	0x06c	HISPI_MODE_CTRL_3
REG_70	0x070	HISPI_MODE_CTRL_4
REG_74	0x074	HISPI_MODE_CTRL_5
REG_80	0x080	DBG_SEL

19.3.5 MIPI Rx Register Description

19.3.5.1 MIPI Rx PHY Register Description

Base address: 0x0A0D0000

REG_00

Table 19.222: REG_00, Offset Address: 0x000

Bits	Name	Access	Description	Reset
13:0	Reserved			
14	reg_mipirx_pd_ibias	R/W	Power down analogibias	0x1
15	Reserved			
21:16	reg_mipirx_pd_rxlp	R/W	Power down analog RXLP	0x3f
31:22	Reserved			

REG_04

Table 19.223: REG_04, Offset Address: 0x004

Bits	Name	Access	Description	Reset
15:0	Reserved			
21:16	reg_mipirx_sel_clk_channel	R/W	Analog macro clock lane select	0x0
30:22	Reserved			
31	reg_mipimpll_clk_csi_en	R/W	Gating test clock from mipimpll	0x0

REG_30

Table 19.224: REG_30, Offset Address: 0x030

Bits	Name	Access	Description	Reset
2:0	reg_sensor_phy_mode	R/W	Sensor PHY mode enable select 0: 1C4D 1: 1C2D + 1C2D else: reserved	0x0
31:3	Reserved			

REG_34

Table 19.225: REG_34, Offset Address: 0x034

Bits	Name	Access	Description	Reset
31:0	reg_mipirx_ro_cal0	RO	Analog lane 0 calibration result	

REG_38

Table 19.226: REG_38, Offset Address: 0x038

Bits	Name	Access	Description	Reset
31:0	reg_mipirx_ro_cal1	RO	Analog lane 1 calibration result	

REG_3C

Table 19.227: REG_3C, Offset Address: 0x03c

Bits	Name	Access	Description	Reset
31:0	reg_mipirx_ro_cal2	RO	Analog lane 2 calibration result	

REG_40

Table 19.228: REG_40, Offset Address: 0x040

Bits	Name	Access	Description	Reset
31:0	reg_mipirx_ro_cal3	RO	Analog lane 3 calibration result	

REG_44

Table 19.229: REG_44, Offset Address: 0x044

Bits	Name	Access	Description	Reset
31:0	reg_mipirx_ro_cal4	RO	Analog lane 4 calibration result	

REG_48

Table 19.230: REG_48, Offset Address: 0x048

Bits	Name	Access	Description	Reset
31:0	reg_mipirx_ro_cal5	RO	Analog lane 5 calibration result	

REG_80

Table 19.231: REG_80, Offset Address: 0x080

Bits	Name	Access	Description	Reset
0	reg_ad_d0_clk_inv	R/W	AD clock lane0 inverse	0x0
1	reg_ad_d1_clk_inv	R/W	AD clock lane1 inverse	0x0
2	reg_ad_d2_clk_inv	R/W	AD clock lane2 inverse	0x0
3	reg_ad_d3_clk_inv	R/W	AD clock lane3 inverse	0x0
4	reg_ad_d4_clk_inv	R/W	AD clock lane4 inverse	0x0
5	reg_ad_d5_clk_inv	R/W	AD clock lane5 inverse	0x0
31:6	Reserved			

REG_A0

Table 19.232: REG_A0, Offset Address: 0x0a0

Bits	Name	Access	Description	Reset
13:0	reg_cam0_vtt	R/W		0x0
15:14	Reserved			
29:16	reg_cam0_vs_str	R/W		0x0
31:30	Reserved			

REG_A4

Table 19.233: REG_A4, Offset Address: 0x0a4

Bits	Name	Access	Description	Reset
13:0	reg_cam0_vs_stp	R/W		0x0
15:14	Reserved			
29:16	reg_cam0_htt	R/W		0x0
31:30	Reserved			

REG_A8

Table 19.234: REG_A8, Offset Address: 0x0a8

Bits	Name	Access	Description	Reset
13:0	reg_cam0_hs_str	R/W		0x0
15:14	Reserved			
29:16	reg_cam0_hs_stp	R/W		0x0
31:30	Reserved			

REG_AC

Table 19.235: REG_AC, Offset Address: 0x0ac

Bits	Name	Access	Description	Reset
0	reg_cam0_vs_pol	R/W		0x0
1	reg_cam0_hs_pol	R/W		0x0
2	reg_cam0_tgen_en	R/W		0x0
31:3	Reserved			

Base address: 0x0A0D0300

REG_00

Table 19.236: REG_00, Offset Address: 0x000

Bits	Name	Access	Description	Reset
1:0	reg_sensor_mode	R/W	Sensor mode select 2'b00: CSI 2'b01: Sub-LVDS & HiSPi 2'b10: SLVSEC	0x0
31:2	Reserved			

REG_04

Table 19.237: REG_04, Offset Address: 0x004

Bits	Name	Access	Description	Reset
2:0	reg_csi_lane_d0_sel	R/W	Data lane 0 select	0x1
3	Reserved			
6:4	reg_csi_lane_d1_sel	R/W	Data lane 1 select	0x2
7	Reserved			
10:8	reg_csi_lane_d2_sel	R/W	Data lane 2 select	0x3
11	Reserved			
14:12	reg_csi_lane_d3_sel	R/W	Data lane 3 select	0x4
31:15	Reserved			

REG_08

Table 19.238: REG_08, Offset Address: 0x008

Bits	Name	Access	Description	Reset
2:0	reg_csi_lane_ck_sel	R/W	Clock lane select	0x0
3	Reserved			
4	reg_csi_lane_ck_pnswap	R/W	Clock lane pn swap	0x0
7:5	Reserved			
8	reg_csi_lane_d0_pnswap	R/W	Data lane 0 pn swap	0x0
9	reg_csi_lane_d1_pnswap	R/W	Data lane 1 pn swap	0x0
10	reg_csi_lane_d2_pnswap	R/W	Data lane 2 pn swap	0x0
11	reg_csi_lane_d3_pnswap	R/W	Data lane 3 pn swap	0x0
15:12	Reserved			
23:16	reg_csi_ck_phase	R/W	Clock lane phase	0x0
31:24	Reserved			

REG_0C

Table 19.239: REG_0C, Offset Address: 0x00c

Bits	Name	Access	Description	Reset
3:0	reg_deskew_lane_en	R/W	Deskew lane enable 4'h0: No lane 4'h1: 1-lane 4'h3: 2-lane 8'hf: 4-lane	0x0
31:4	Reserved			

REG_20

Table 19.240: REG_20, Offset Address: 0x020

Bits	Name	Access	Description	Reset
0	reg_slvds_inv_en	R/W	Sub-LVDS bit reverse 1'b0: LSB first 1'b1: MSB first	0x1
1	Reserved			
3:2	reg_slvds_bit_mode	R/W	Sub-LVDS bit mode 2'b00: 8-bit 2'b01: 10-bit 2'b10: 12-bit	0x2
7:4	reg_slvds_lane_en	R/W	Sub-LVDS lane enable Set this register to start finding sync code	0x0
15:8	Reserved			
27:16	reg_slvds_sav_1st	R/W	Sub-LVDS sync code 1st symbol	0xfff
31:28	Reserved			

REG_24

Table 19.241: REG_24, Offset Address: 0x024

Bits	Name	Access	Description	Reset
11:0	reg_slvds_sav_2nd	R/W	Sub-LVDS sync code 2nd symbol	0x0
15:12	Reserved			
27:16	reg_slvds_sav_3rd	R/W	Sub-LVDS sync code 3rd	0x0
31:28	Reserved			

REG_D0_0

Table 19.242: REG_D0_0, Offset Address: 0x100

Bits	Name	Access	Description	Reset
0	reg_d0_prbs9_en	R/W	Manual PRBS9 enable	0x0
1	reg_d0_prbs9_clr_err	R/W	PRBS9 clear error	0x0
2	reg_d0_prbs9_source	R/W	PRBS9 source select 1'b0: after sync code shift 1'b1: direct from input	0x0
3	reg_d0_prbs9_stop_when_done	R/W	PRBS9 error count accumalation 1'b0: still count after test time done 1'b1: do not count after test time done	0x0
7:4	Reserved			
15:8	reg_d0_calib_max	R/W	Calibration max step	0x1f
23:16	reg_d0_calib_step	R/W	Calibration one step value	0x1
31:24	reg_d0_calib_pattern	R/W	Calibration golden pattern	0xaa

REG_D0_1

Table 19.243: REG_D0_1, Offset Address: 0x104

Bits	Name	Access	Description	Reset
0	reg_d0_calib_en	R/W	Calibration software enable	0x0
1	reg_d0_calib_source	R/W	Calibration source 1'b0: normal position 1'b1: direct from analog	0x0
2	reg_d0_calib_mode	R/W	Calibration software mode 1'b0: use identical calibration pattern 1'b1: use PRBS9 pattern	0x0
3	reg_d0_calib_ignore	R/W	Ignore calibration command	0x0
31:4	Reserved			

REG_D0_3

Table 19.244: REG_D0_3, Offset Address: 0x10c

Bits	Name	Access	Description	Reset
31:0	reg_d0_skew_calib_result_0	RO	Calibration result phase 0~31	

REG_D0_4

Table 19.245: REG_D0_4, Offset Address: 0x110

Bits	Name	Access	Description	Reset
31:0	reg_d0_skew_calib_result_1	RO	Calibration result phase 32~63	

REG_D0_5

Table 19.246: REG_D0_5, Offset Address: 0x114

Bits	Name	Access	Description	Reset
31:0	reg_d0_skew_calib_result_2	RO	Calibration result phase 64~95	

REG_D0_6

Table 19.247: REG_D0_6, Offset Address: 0x118

Bits	Name	Access	Description	Reset
31:0	reg_d0_skew_calib_result_3	RO	Calibration result phase 96~127	

REG_D0_7

Table 19.248: REG_D0_7, Offset Address: 0x11c

Bits	Name	Access	Description	Reset
31:0	reg_d0_skew_calib_result_4	RO	Calibration result phase 128~159	

REG_D0_8

Table 19.249: REG_D0_8, Offset Address: 0x120

Bits	Name	Access	Description	Reset
31:0	reg_d0_skew_calib_result_5	RO	Calibration result phase 160~191	

REG_D0_9

Table 19.250: REG_D0_9, Offset Address: 0x124

Bits	Name	Access	Description	Reset
31:0	reg_d0_skew_calib_result_6	RO	Calibration result phase 192~223	

REG_D0_A

Table 19.251: REG_D0_A, Offset Address: 0x128

Bits	Name	Access	Description	Reset
31:0	reg_d0_skew_calib_result_7	RO	Calibration result phase 224~255	

REG_D1_0

Table 19.252: REG_D1_0, Offset Address: 0x140

Bits	Name	Access	Description	Reset
0	reg_d1_prbs9_en	R/W	Manual PRBS9 enable	0x0
1	reg_d1_prbs9_clr_err	R/W	PRBS9 clear error	0x0
2	reg_d1_prbs9_source	R/W	PRBS9 source select 1'b0: after sync code shift 1'b1: direct from input	0x0
3	reg_d1_prbs9_stop_when_done	R/W	PRBS9 error count accumalation 1'b0: still count after test time done 1'b1: do not count after test time done	0x0
7:4	Reserved			
15:8	reg_d1_calib_max	R/W	Calibration max step	0x1f
23:16	reg_d1_calib_step	R/W	Calibration one step value	0x1
31:24	reg_d1_calib_pattern	R/W	Calibration golden pattern	0xaa

REG_D1_1

Table 19.253: REG_D1_1, Offset Address: 0x144

Bits	Name	Access	Description	Reset
0	reg_d1_calib_en	R/W	Calibration software enable	0x0
1	reg_d1_calib_source	R/W	Calibration source 1'b0: normal position 1'b1: direct from analog	0x0
2	reg_d1_calib_mode	R/W	Calibration software mode 1'b0: use identical calibration pattern 1'b1: use PRBS9 pattern	0x0
3	reg_d1_calib_ignore	R/W	Ignore calibration command	0x0
31:4	Reserved			

REG_D1_3

Table 19.254: REG_D1_3, Offset Address: 0x14c

Bits	Name	Access	Description	Reset
31:0	reg_d1_skew_calib_result_0	RO	Calibration result phase 0~31	

REG_D1_4

Table 19.255: REG_D1_4, Offset Address: 0x150

Bits	Name	Access	Description	Reset
31:0	reg_d1_skew_calib_result_1	RO	Calibration result phase 32~63	

REG_D1_5

Table 19.256: REG_D1_5, Offset Address: 0x154

Bits	Name	Access	Description	Reset
31:0	reg_d1_skew_calib_result_2	RO	Calibration result phase 64~95	

REG_D1_6

Table 19.257: REG_D1_6, Offset Address: 0x158

Bits	Name	Access	Description	Reset
31:0	reg_d1_skew_calib_result_3	RO	Calibration result phase 96~127	

REG_D1_7

Table 19.258: REG_D1_7, Offset Address: 0x15c

Bits	Name	Access	Description	Reset
31:0	reg_d1_skew_calib_result_4	RO	Calibration result phase 128~159	

REG_D1_8

Table 19.259: REG_D1_8, Offset Address: 0x160

Bits	Name	Access	Description	Reset
31:0	reg_d1_skew_calib_result_5	RO	Calibration result phase 160~191	

REG_D1_9

Table 19.260: REG_D1_9, Offset Address: 0x164

Bits	Name	Access	Description	Reset
31:0	reg_d1_skew_calib_result_6	RO	Calibration result phase 192~223	

REG_D1_A

Table 19.261: REG_D1_A, Offset Address: 0x168

Bits	Name	Access	Description	Reset
31:0	reg_d1_skew_calib_result_7	RO	Calibration result phase 224~255	

REG_D2_0

Table 19.262: REG_D2_0, Offset Address: 0x180

Bits	Name	Access	Description	Reset
0	reg_d2_prbs9_en	R/W	Manual PRBS9 enable	0x0
1	reg_d2_prbs9_clr_err	R/W	PRBS9 clear error	0x0
2	reg_d2_prbs9_source	R/W	PRBS9 source select 1'b0: after sync code shift 1'b1: direct from input	0x0
3	reg_d2_prbs9_stop_when_done	R/W	PRBS9 error count accumalation 1'b0: still count after test time done 1'b1: do not count after test time done	0x0
7:4	Reserved			
15:8	reg_d2_calib_max	R/W	Calibration max step	0x1f
23:16	reg_d2_calib_step	R/W	Calibration one step value	0x1
31:24	reg_d2_calib_pattern	R/W	Calibration golden pattern	0xaa

REG_D2_1

Table 19.263: REG_D2_1, Offset Address: 0x184

Bits	Name	Access	Description	Reset
0	reg_d2_calib_en	R/W	Calibration software enable	0x0
1	reg_d2_calib_source	R/W	Calibration source 1'b0: normal position 1'b1: direct from analog	0x0
2	reg_d2_calib_mode	R/W	Calibration software mode 1'b0: use identical calibration pattern 1'b1: use PRBS9 pattern	0x0
3	reg_d2_calib_ignore	R/W	Ignore calibration command	0x0
31:4	Reserved			

REG_D2_3

Table 19.264: REG_D2_3, Offset Address: 0x18c

Bits	Name	Access	Description	Reset
31:0	reg_d2_skew_calib_result_0	RO	Calibration result phase 0~31	

REG_D2_4

Table 19.265: REG_D2_4, Offset Address: 0x190

Bits	Name	Access	Description	Reset
31:0	reg_d2_skew_calib_result_1	RO	Calibration result phase 32~63	

REG_D2_5

Table 19.266: REG_D2_5, Offset Address: 0x194

Bits	Name	Access	Description	Reset
31:0	reg_d2_skew_calib_result_2	RO	Calibration result phase 64~95	

REG_D2_6

Table 19.267: REG_D2_6, Offset Address: 0x198

Bits	Name	Access	Description	Reset
31:0	reg_d2_skew_calib_result_3	RO	Calibration result phase 96~127	

REG_D2_7

Table 19.268: REG_D2_7, Offset Address: 0x19c

Bits	Name	Access	Description	Reset
31:0	reg_d2_skew_calib_result_4	RO	Calibration result phase 128~159	

REG_D2_8

Table 19.269: REG_D2_8, Offset Address: 0x1a0

Bits	Name	Access	Description	Reset
31:0	reg_d2_skew_calib_result_5	RO	Calibration result phase 160~191	

REG_D2_9

Table 19.270: REG_D2_9, Offset Address: 0x1a4

Bits	Name	Access	Description	Reset
31:0	reg_d2_skew_calib_result_6	RO	Calibration result phase 192~223	

REG_D2_A

Table 19.271: REG_D2_A, Offset Address: 0x1a8

Bits	Name	Access	Description	Reset
31:0	reg_d2_skew_calib_result_7	RO	Calibration result phase 224~255	

REG_D3_0

Table 19.272: REG_D3_0, Offset Address: 0x1c0

Bits	Name	Access	Description	Reset
0	reg_d3_prbs9_en	R/W	Manual PRBS9 enable	0x0
1	reg_d3_prbs9_clr_err	R/W	PRBS9 clear error	0x0
2	reg_d3_prbs9_source	R/W	PRBS9 source select 1'b0: after sync code shift 1'b1: direct from input	0x0
3	reg_d3_prbs9_stop_when_done	R/W	PRBS9 error count accumalation 1'b0: still count after test time done 1'b1: do not count after test time done	0x0
7:4	Reserved			
15:8	reg_d3_calib_max	R/W	Calibration max step	0x1f
23:16	reg_d3_calib_step	R/W	Calibration one step value	0x1
31:24	reg_d3_calib_pattern	R/W	Calibration golden pattern	0xaa

REG_D3_1

Table 19.273: REG_D3_1, Offset Address: 0x1c4

Bits	Name	Access	Description	Reset
0	reg_d3_calib_en	R/W	Calibration software enable	0x0
1	reg_d3_calib_source	R/W	Calibration source 1'b0: normal position 1'b1: direct from analog	0x0
2	reg_d3_calib_mode	R/W	Calibration software mode 1'b0: use identical calibration pattern 1'b1: use PRBS9 pattern	0x0
3	reg_d3_calib_ignore	R/W	Ignore calibration command	0x0
31:4	Reserved			

REG_D3_3

Table 19.274: REG_D3_3, Offset Address: 0x1cc

Bits	Name	Access	Description	Reset
31:0	reg_d3_skew_calib_result_0	RO	Calibration result phase 0~31	

REG_D3_4

Table 19.275: REG_D3_4, Offset Address: 0x1d0

Bits	Name	Access	Description	Reset
31:0	reg_d3_skew_calib_result_1	RO	Calibration result phase 32~63	

REG_D3_5

Table 19.276: REG_D3_5, Offset Address: 0x1d4

Bits	Name	Access	Description	Reset
31:0	reg_d3_skew_calib_result_2	RO	Calibration result phase 64~95	

REG_D3_6

Table 19.277: REG_D3_6, Offset Address: 0x1d8

Bits	Name	Access	Description	Reset
31:0	reg_d3_skew_calib_result_3	RO	Calibration result phase 96~127	

REG_D3_7

Table 19.278: REG_D3_7, Offset Address: 0x1dc

Bits	Name	Access	Description	Reset
31:0	reg_d3_skew_calib_result_4	RO	Calibration result phase 128~159	

REG_D3_8

Table 19.279: REG_D3_8, Offset Address: 0x1e0

Bits	Name	Access	Description	Reset
31:0	reg_d3_skew_calib_result_5	RO	Calibration result phase 160~191	

REG_D3_9

Table 19.280: REG_D3_9, Offset Address: 0x1e4

Bits	Name	Access	Description	Reset
31:0	reg_d3_skew_calib_result_6	RO	Calibration result phase 192~223	

REG_D3_A

Table 19.281: REG_D3_A, Offset Address: 0x1e8

Bits	Name	Access	Description	Reset
31:0	reg_d3_skew_calib_result_7	RO	Calibration result phase 224~255	

Base address: 0x0A0D0600

REG_00

Table 19.282: REG_00, Offset Address: 0x000

Bits	Name	Access	Description	Reset
1:0	reg_sensor_mode	R/W	Sensor mode select 2'b00: CSI 2'b01: Sub-LVDS & HiSPi 2'b10: SLVSEC	0x0
31:2	Reserved			

REG_04

Table 19.283: REG_04, Offset Address: 0x004

Bits	Name	Access	Description	Reset
1:0	reg_csi_lane_d0_sel	R/W	Data lane 0 select	0x1
3:2	Reserved			
5:4	reg_csi_lane_d1_sel	R/W	Data lane 1 select	0x2
31:6	Reserved			

REG_08

Table 19.284: REG_08, Offset Address: 0x008

Bits	Name	Access	Description	Reset
1:0	reg_csi_lane_ck_sel	R/W	Clock lane select	0x0
3:2	Reserved			
4	reg_csi_lane_ck_pnswap	R/W	Clock lane pn swap	0x0
7:5	Reserved			
8	reg_csi_lane_d0_pnswap	R/W	Data lane 0 pn swap	0x0
9	reg_csi_lane_d1_pnswap	R/W	Data lane 1 pn swap	0x0
15:10	Reserved			
23:16	reg_csi_ck_phase	R/W	Clock lane phase	0x0
31:24	Reserved			

REG_0C

Table 19.285: REG_0C, Offset Address: 0x00c

Bits	Name	Access	Description	Reset
1:0	reg_deskew_lane_en	R/W	Deskew lane enable 2'h0: No lane 2'h1: 1-lane 2'h3: 2-lane	0x0
31:2	Reserved			

REG_20

Table 19.286: REG_20, Offset Address: 0x020

Bits	Name	Access	Description	Reset
0	reg_slvds_inv_en	R/W	Sub-LVDS bit reverse 1'b0: LSB first 1'b1: MSB first	0x1
1	Reserved			
3:2	reg_slvds_bit_mode	R/W	Sub-LVDS bit mode 2'b00: 8-bit 2'b01: 10-bit 2'b10: 12-bit	0x2
5:4	reg_slvds_lane_en	R/W	Sub-LVDS lane enable Set this register to start finding sync code	0x0
31:6	Reserved			

REG_D0_0

Table 19.287: REG_D0_0, Offset Address: 0x100

Bits	Name	Access	Description	Reset
0	reg_d0_prbs9_en	R/W	Manual PRBS9 enable	0x0
1	reg_d0_prbs9_clr_err	R/W	PRBS9 clear error	0x0
2	reg_d0_prbs9_source	R/W	PRBS9 source select 1'b0: after sync code shift 1'b1: direct from input	0x0
3	reg_d0_prbs9_stop_when_done	R/W	PRBS9 error count accumalation 1'b0: still count after test time done 1'b1: do not count after test time done	0x0
7:4	Reserved			
15:8	reg_d0_calib_max	R/W	Calibration max step	0x1f
23:16	reg_d0_calib_step	R/W	Calibration one step value	0x1
31:24	reg_d0_calib_pattern	R/W	Calibration golden pattern	0xaa

REG_D0_1

Table 19.288: REG_D0_1, Offset Address: 0x104

Bits	Name	Access	Description	Reset
0	reg_d0_calib_en	R/W	Calibration software enable	0x0
1	reg_d0_calib_source	R/W	Calibration source 1'b0: normal position 1'b1: direct from analog	0x0
2	reg_d0_calib_mode	R/W	Calibration software mode 1'b0: use identical calibration pattern 1'b1: use PRBS9 pattern	0x0
3	reg_d0_calib_ignore	R/W	Ignore calibration command	0x0
31:4	Reserved			

REG_D0_3

Table 19.289: REG_D0_3, Offset Address: 0x10c

Bits	Name	Access	Description	Reset
31:0	reg_d0_skew_calib_result_0	RO	Calibration result phase 0~31	

REG_D0_4

Table 19.290: REG_D0_4, Offset Address: 0x110

Bits	Name	Access	Description	Reset
31:0	reg_d0_skew_calib_result_1	RO	Calibration result phase 32~63	

REG_D0_5

Table 19.291: REG_D0_5, Offset Address: 0x114

Bits	Name	Access	Description	Reset
31:0	reg_d0_skew_calib_result_2	RO	Calibration result phase 64~95	

REG_D0_6

Table 19.292: REG_D0_6, Offset Address: 0x118

Bits	Name	Access	Description	Reset
31:0	reg_d0_skew_calib_result_3	RO	Calibration result phase 96~127	

REG_D0_7

Table 19.293: REG_D0_7, Offset Address: 0x11c

Bits	Name	Access	Description	Reset
31:0	reg_d0 _skew_calib_result_4	RO	Calibration result phase 128~159	

REG_D0_8

Table 19.294: REG_D0_8, Offset Address: 0x120

Bits	Name	Access	Description	Reset
31:0	reg_d0_skew_calib_result_5	RO	Calibration result phase 160~191	

REG_D0_9

Table 19.295: REG_D0_9, Offset Address: 0x124

Bits	Name	Access	Description	Reset
31:0	reg_d0_skew_calib_result_6	RO	Calibration result phase 192~223	

REG_D0_A

Table 19.296: REG_D0_A, Offset Address: 0x128

Bits	Name	Access	Description	Reset
31:0	reg_d0_skew_calib_result_7	RO	Calibration result phase 224~255	

REG_D1_0

Table 19.297: REG_D1_0, Offset Address: 0x140

Bits	Name	Access	Description	Reset
0	reg_d1_prbs9_en	R/W	Manual PRBS9 enable	0x0
1	reg_d1_prbs9_clr_err	R/W	PRBS9 clear error	0x0
2	reg_d1_prbs9_source	R/W	PRBS9 source select 1'b0: after sync code shift 1'b1: direct from input	0x0
3	reg_d1_prbs9_stop_when_done	R/W	PRBS9 error count accumalation 1'b0: still count after test time done 1'b1: do not count after test time done	0x0
7:4	Reserved			
15:8	reg_d1_calib_max	R/W	Calibration max step	0x1f
23:16	reg_d1_calib_step	R/W	Calibration one step value	0x1
31:24	reg_d1_calib_pattern	R/W	Calibration golden pattern	0xaa

REG_D1_1

Table 19.298: REG_D1_1, Offset Address: 0x144

Bits	Name	Access	Description	Reset
0	reg_d1_calib_en	R/W	Calibration software enable	0x0
1	reg_d1_calib_source	R/W	Calibration source 1'b0: normal position 1'b1: direct from analog	0x0
2	reg_d1_calib_mode	R/W	Calibration software mode 1'b0: use identical calibration pattern 1'b1: use PRBS9 pattern	0x0
3	reg_d1_calib_ignore	R/W	Ignore calibration command	0x0
31:4	Reserved			

REG_D1_3

Table 19.299: REG_D1_3, Offset Address: 0x14c

Bits	Name	Access	Description	Reset
31:0	reg_d1_skew_calib_result_0	RO	Calibration result phase 0~31	

REG_D1_4

Table 19.300: REG_D1_4, Offset Address: 0x150

Bits	Name	Access	Description	Reset
31:0	reg_d1_skew_calib_result_1	RO	Calibration result phase 32~63	

REG_D1_5

Table 19.301: REG_D1_5, Offset Address: 0x154

Bits	Name	Access	Description	Reset
31:0	reg_d1_skew_calib_result_2	RO	Calibration result phase 64~95	

REG_D1_6

Table 19.302: REG_D1_6, Offset Address: 0x158

Bits	Name	Access	Description	Reset
31:0	reg_d1_skew_calib_result_3	RO	Calibration result phase 96~127	

REG_D1_7

Table 19.303: REG_D1_7, Offset Address: 0x15c

Bits	Name	Access	Description	Reset
31:0	reg_d1_skew_calib_result_4	RO	Calibration result phase 128~159	

REG_D1_8

Table 19.304: REG_D1_8, Offset Address: 0x160

Bits	Name	Access	Description	Reset
31:0	reg_d1_skew_calib_result_5	RO	Calibration result phase 160~191	

REG_D1_9

Table 19.305: REG_D1_9, Offset Address: 0x164

Bits	Name	Access	Description	Reset
31:0	reg_d1_skew_calib_result_6	RO	Calibration result phase 192~223	

REG_D1_A

Table 19.306: REG_D1_A, Offset Address: 0x168

Bits	Name	Access	Description	Reset
31:0	reg_d1_skew_calib_result_7	RO	Calibration result phase 224~255	

19.3.5.2 MIPI Rx CSI Control Registers

REG_00

Table 19.307: REG_00, Offset Address: 0x000

Bits	Name	Access	Description	Reset
2:0	reg_csi_lane_mode	R/W	Lane mode 3'b000: 1-lane 3'b001: 2-lane 3'b011: 4-lane 3'b111: 8-lane	0x0
3	reg_csi_ignore_ecc	R/W	Ignore ecc result 1'b0: normal 1'b1: still processing even ecc error	0x0
4	reg_csi_vc_check	R/W	VC check enable 1'b0: do not check VC 1'b1: only process packets that meet vc_set[3:0]	0x0
7:5	Reserved			
11:8	reg_csi_vc_set	R/W	VC set only use when reg_csi_vc_check assert	0x0
12	reg_csi_line_start_sent	R/W	LS and LE packet sent 1'b0: create hsync signal by controller 1'b1: use LS and LE to create hsync signal	0x0
31:13	Reserved			

REG_04

Table 19.308: REG_04, Offset Address: 0x004

Bits	Name	Access	Description	Reset
7:0	reg_csi_intr_mask	R/W	Interrupt mask control	0x0
15:8	reg_csi_intr_clr	W1T	Interrupt clear	
16	reg_csi_hdr_en	R/W	HDR mode enable	0x0
17	reg_csi_hdr_mode	R/W	HDR mode selection 1'b0: HDR VC mode 1'b1: HDR ID mode	0x0
18	reg_csi_id_rm_else	R/W	Remove non reconized ID line 1'b0: dont remove 1'b1: remove	0x1
19	reg_csi_id_rm_ob	R/W	Remove ob line 1'b0: don't remove 1'b1: remove	0x1
31:20	Reserved			

REG_08

Table 19.309: REG_08, Offset Address: 0x008

Bits	Name	Access	Description	Reset
15:0	reg_csi_n0_ob_lef	R/W	ID for LEF ob n0	0x221
31:16	reg_csi_n0_ob_sef	R/W	ID for SEF ob n0	0x222

REG_0C

Table 19.310: REG_0C, Offset Address: 0x00c

Bits	Name	Access	Description	Reset
15:0	reg_csi_n0_lef	R/W	ID for LEF active n0	0x241
31:16	reg_csi_n1_ob_lef	R/W	ID for LEF ob n1	0x231

REG_10

Table 19.311: REG_10, Offset Address: 0x010

Bits	Name	Access	Description	Reset
15:0	reg_csi_n1_ob_sef	R/W	ID for SEF ob n1	0x232
31:16	reg_csi_n1_lef	R/W	ID for LEF active n1	0x251

REG_14

Table 19.312: REG_14, Offset Address: 0x014

Bits	Name	Access	Description	Reset
5:0	reg_csi_blc_dt	R/W	Data type for optical black line	0x37
7:6	Reserved			
8	reg_csi_blc_en	R/W	Optical black line mode enable	0x0
11:9	Reserved			
14:12	reg_csi_blc_format_set	R/W	Optical black line data format set 3'd0: YUV422 8bit 3'd1: YUV422 10bit 3'd2: RAW8 3'd3: RAW10 3'd4: RAW12 3'd5: RAW16 else: reserved	0x2
31:15	Reserved			

REG_18

Table 19.313: REG_18, Offset Address: 0x018

Bits	Name	Access	Description	Reset
3:0	reg_csi_vc_map_ch00	R/W	VC mapping to ISP channel 00	0x0
7:4	reg_csi_vc_map_ch01	R/W	VC mapping to ISP channel 01	0x1
11:8	reg_csi_vc_map_ch10	R/W	VC mapping to ISP channel 10	0x2
15:12	reg_csi_vc_map_ch11	R/W	VC mapping to ISP channel 11	0x3
31:16	Reserved			

REG_1C

Table 19.314: REG_1C, Offset Address: 0x01c

Bits	Name	Access	Description	Reset
15:0	reg_csi_n0_sef	R/W	ID for SEF active n0	0x242
31:16	reg_csi_n1_sef	R/W	ID for SEF active n1	0x252

REG_20

Table 19.315: REG_20, Offset Address: 0x020

Bits	Name	Access	Description	Reset
15:0	reg_csi_n0_sef2	R/W	ID for SEF2 active n0	0x244
31:16	reg_csi_n1_sef2	R/W	ID for SEF2 active n1	0x254

REG_24

Table 19.316: REG_24, Offset Address: 0x024

Bits	Name	Access	Description	Reset
15:0	reg_csi_n0_ob_sef2	R/W	ID for SEF2 ob n0	0x224
31:16	reg_csi_n1_ob_sef2	R/W	ID for SEF2 ob n1	0x234

REG_40

Table 19.317: REG_40, Offset Address: 0x040

Bits	Name	Access	Description	Reset
0	reg_csi_ecc_no_error	RO	ECC no error	
1	reg_csi_ecc_corrected _error	RO	ECC corrected error	
2	reg_csi_ecc_error	RO	ECC error	
3	Reserved			
4	reg_csi_crc_error	RO	CRC error	
5	reg_csi_wc_error	RO	WC error	
7:6	Reserved			
8	reg_csi_fifo_full	RO	CSI FIFO full	
15:9	Reserved			
21:16	reg_csi_decode_format	RO	CSI decode format from header bit[0]: YUV422 8bit bit[1]: YUV422 10bit bit[2]: RAW8 bit[3]: RAW10 bit[4]: RAW12 bit[5]: RAW16	
31:22	Reserved			

REG_60

Table 19.318: REG_60, Offset Address: 0x060

Bits	Name	Access	Description	Reset
7:0	reg_csi_intr_status	RO	Interrupt status bit[0]: ecc error bit[1]: crc error bit[2]: hdr id error bit[3]: word count error bit[4]: fifo full	
31:8	Reserved			

REG_70

Table 19.319: REG_70, Offset Address: 0x070

Bits	Name	Access	Description	Reset
1:0	reg_csi_vs_gen_mode	R/W	2'b00: vs gen by FS 2'b01: vs gen by FE else: vs gen by FS & FE	0x2
3:2	Reserved			
4	reg_csi_vs_gen_by_vcset	R/W	Vsync generation setting 1'b0: generated by all vc short packet 1'b1: only generated by indicated vc short packet	0x0
31:5	Reserved			

REG_74

Table 19.320: REG_74, Offset Address: 0x074

Bits	Name	Access	Description	Reset
0	reg_csi_hdr_dt_mode	R/W	CSI HDR DT mode enable	0x0
3:1	Reserved			
9:4	reg_csi_hdr_dt_format	R/W	CSI HDR DT mode video format data type	0x0
11:10	Reserved			
17:12	reg_csi_hdr_dt_lef	R/W	CSI HDR DT mode LEF data type	0x0
19:18	Reserved			
25:20	reg_csi_hdr_dt_sef	R/W	CSI HDR DT mode SEF data type	0x0
31:26	Reserved			

19.3.5.3 MIPI Rx Sub-LVDS Control Registers

REG_00

Table 19.321: REG_00, Offset Address: 0x000

Bits	Name	Access	Description	Reset
7:0	reg_slvds_enable	R/W	Sub-LVDS lane enable for each lane	0x0
9:8	reg_slvds_bit_mode	R/W	Sub-LVDS bit mode 2'b00: 8-bit 2'b01: 10-bit 2'b10: 12-bit	0x2
10	reg_slvds_data_reverse	R/W	Sub-LVDS data packet bit inverse	0x0
11	Reserved			
12	reg_slvds_hdr_mode	R/W	Sub-LVDS HDR mode enable	0x0
13	reg_slvds_hdr_pattern	R/W	Sub-LVDS HDR pattern mode 1'b0: pattern 1 1'b1: pattern 2	0x0
31:14	Reserved			

REG_04

Table 19.322: REG_04, Offset Address: 0x004

Bits	Name	Access	Description	Reset
11:0	reg_slvds_sync_1st	R/W	Sub-LVDS SYNC code 1st word	0xFFF
15:12	Reserved			
27:16	reg_slvds_sync_2nd	R/W	Sub-LVDS SYNC code 2nd word	0x000
31:28	Reserved			

REG_08

Table 19.323: REG_08, Offset Address: 0x008

Bits	Name	Access	Description	Reset
11:0	reg_slvds_sync_3rd	R/W	Sub-LVDS SYNC code 3rd word	0x000
15:12	Reserved			
27:16	reg_slvds_norm_bk_sav	R/W	Normal mode blanking SAV	0xAB0
31:28	Reserved			

REG_0C

Table 19.324: REG_0C, Offset Address: 0x00c

Bits	Name	Access	Description	Reset
11:0	reg_slvds_norm_bk_eav	R/W	Normal mode blanking EAV	0xB60
15:12	Reserved			
27:16	reg_slvds_norm_sav	R/W	Normal mode active SAV	0x800
31:28	Reserved			

REG_10

Table 19.325: REG_10, Offset Address: 0x010

Bits	Name	Access	Description	Reset
11:0	reg_slvds_norm_eav	R/W	Normal mode active EAV	0x9D0
15:12	Reserved			
27:16	reg_slvds_n0_bk_sav	R/W	HDR mode n0 blanking SAV	0x2B0
31:28	Reserved			

REG_14

Table 19.326: REG_14, Offset Address: 0x014

Bits	Name	Access	Description	Reset
11:0	reg_slvds_n0_bk_eav	R/W	HDR mode n0 blanking EAV	0x360
15:12	Reserved			
27:16	reg_slvds_n1_bk_sav	R/W	HDR mode n1 blanking SAV	0x6B0
31:28	Reserved			

REG_18

Table 19.327: REG_18, Offset Address: 0x018

Bits	Name	Access	Description	Reset
11:0	reg_slvds_n1_bk_eav	R/W	HDR mode n1 blanking EAV	0x760
15:12	Reserved			
27:16	reg_slvds_n0_lef_sav	R/W	Sub-LVDS mode: n0 long exposure sav Sub-LVDS 12-bit LEF SAV n0 (801) Sub-LVDS 10-bit LEF SAV n0 (004) HiSPi P-SP mode: SOL T1 (800)	0x801
31:28	Reserved			

REG_1C

Table 19.328: REG_1C, Offset Address: 0x01c

Bits	Name	Access	Description	Reset
11:0	reg_slvds_n0_lef_eav	R/W	Sub-LVDS mode: n0 long exposure eav Sub-LVDS 12-bit LEF EAV n0 (9D1) Sub-LVDS 10-bit LEF EAV n0 (1D4) HiSPi P-SP mode: EOL T1 (A00)	0x9D1
15:12	Reserved			
27:16	reg_slvds_n0_sef_sav	R/W	Sub-LVDS mode: n0 short exposure sav Sub-LVDS 12-bit SEF SAV n0 (802) Sub-LVDS 10-bit SEF SAV n0 (008) HiSPi P-SP mode: SOL T2 (820)	0x802
31:28	Reserved			

REG_20

Table 19.329: REG_20, Offset Address: 0x020

Bits	Name	Access	Description	Reset
11:0	reg_slvds_n0_sef_eav	R/W	Sub-LVDS mode: n0 short exposure eav Sub-LVDS 12-bit SEF EAV n0 (9D2) Sub-LVDS 10-bit SEF EAV n0 (1d8) HiSPi P-SP mode: EOL T2 (A20)	0x9D2
15:12	Reserved			
27:16	reg_slvds_n1_lef_sav	R/W	Sub-LVDS mode: n1 long exposure sav Sub-LVDS 12-bit LEF SAV n1 (C01) Sub-LVDS 10-bit LEF SAV n1 (404) HiSPi P-SP mode: SOF T1 (C00)	0xC01
31:28	Reserved			

REG_24

Table 19.330: REG_24, Offset Address: 0x024

Bits	Name	Access	Description	Reset
11:0	reg_slvds_n1_lef_eav	R/W	Sub-LVDS mode: n1 long exposure eav Sub-LVDS 12-bit LEF EAV n1 (DD1) Sub-LVDS 10-bit LEF EAV n1 (5D4) HiSPi P-SP mode: EOF T1 (E00)	0xDD1
15:12	Reserved			
27:16	reg_slvds_n1_sef_sav	R/W	Sub-LVDS mode: n1 short exposure sav Sub-LVDS 12-bit SEF SAV n1 (C02) Sub-LVDS 10-bit SEF SAV n1 (408) HiSPi P-SP mode: SOF T2 (C20)	0xC02
31:28	Reserved			

REG_28

Table 19.331: REG_28, Offset Address: 0x028

Bits	Name	Access	Description	Reset
11:0	reg_slvds_n1_sef_eav	R/W	Sub-LVDS mode: n1 short exposure eav Sub-LVDS 12-bit SEF EAV n1 (DD2) Sub-LVDS 10-bit SEF EAV n1 (5D8) HiSPi P-SP mode: EOF T2 (E20)	0xDD2
31:12	Reserved			

REG_2C

Table 19.332: REG_2C, Offset Address: 0x02c

Bits	Name	Access	Description	Reset
11:0	reg_vs_gen_sync_code	R/W	vs generate sync code value using scenario: HiSPi P-SP HDR	0xC00
12	reg_vs_gen_by_sync_code	R/W	vs generate by identical sync code using scenario: HiSPi P-SP HDR	0x0
31:13	Reserved			

REG_30

Table 19.333: REG_30, Offset Address: 0x030

Bits	Name	Access	Description	Reset
2:0	reg_slvds_lane_mode	R/W	Sub-LVDS lane mode 2'b0: 1-lane 2'b1: 2-lane 2'b3: 4-lane 2'b7: 8-lane	0x3
3	Reserved			
11:4	reg_slvds_sync_source	R/W	Sub-LVDS output sync source select	0x1
31:12	Reserved			

REG_50

Table 19.334: REG_50, Offset Address: 0x050

Bits	Name	Access	Description	Reset
11:0	reg_slvds_n0_lsef_sav	R/W	SAV for n0 long & short exposure both exist line only used for pattern 2	0x803
15:12	Reserved			
27:16	reg_slvds_n0_lsef_eav	R/W	EAV for n0 long & short exposure both exist line only used for pattern 2	0x9D3
31:28	Reserved			

REG_54

Table 19.335: REG_54, Offset Address: 0x054

Bits	Name	Access	Description	Reset
11:0	reg_slvds_n1_lsef_sav	R/W	SAV for n1 long & short exposure both exist line only used for pattern 2	0xC03
15:12	Reserved			
27:16	reg_slvds_n1_lsef_eav	R/W	EAV for n1 long & short exposure both exist line only used for pattern 2	0xDD3
31:28	Reserved			

REG_58

Table 19.336: REG_58, Offset Address: 0x058

Bits	Name	Access	Description	Reset
13:0	reg_slvds_hdr_p2_hsize	R/W	Hsize for pattern 2	0xF0
15:14	Reserved			
29:16	reg_slvds_hdr_p2_hblank	R/W	Hblank size for pattern 2	0x14
31:30	Reserved			

REG_60

Table 19.337: REG_60, Offset Address: 0x060

Bits	Name	Access	Description	Reset
0	reg_hispi_mode	R/W	HiSPi mode enable 1'b0: Sub-LVDS 1'b1: HiSPi	0x0
1	reg_hispi_use_hsize	R/W	HiSPi DE de-assert by register count	0x0
3:2	Reserved			
4	reg_hispi_hdr_psp_mode	R/W	HiSPi P-SP HDR mode enable	0x0
31:5	Reserved			

REG_64

Table 19.338: REG_64, Offset Address: 0x064

Bits	Name	Access	Description	Reset
11:0	reg_hispi_norm_sof	R/W	HiSPi SOF sync code	0xC00
15:12	Reserved			
27:16	reg_hispi_norm_eof	R/W	HiSPi EOF sync code	0xE00
31:28	Reserved			

REG_68

Table 19.339: REG_68, Offset Address: 0x068

Bits	Name	Access	Description	Reset
11:0	reg_hispi_hdr_t1_sof	R/W	HiSPi HDR T1 SOF	0xC00
15:12	Reserved			
27:16	reg_hispi_hdr_t1_eof	R/W	HiSPi HDR T1 EOF	0xE00
31:28	Reserved			

REG_6C

Table 19.340: REG_6C, Offset Address: 0x06c

Bits	Name	Access	Description	Reset
11:0	reg_hispi_hdr_t1_sol	R/W	HiSPi HDR T1 SOL	0x800
15:12	Reserved			
27:16	reg_hispi_hdr_t1_eol	R/W	HiSPi HDR T1 EOL	0xA00
31:28	Reserved			

REG_70

Table 19.341: REG_70, Offset Address: 0x070

Bits	Name	Access	Description	Reset
11:0	reg_hispi_hdr_t2_sof	R/W	HiSPi HDR T2 SOF	0xC20
15:12	Reserved			
27:16	reg_hispi_hdr_t2_eof	R/W	HiSPi HDR T2 EOF	0xE20
31:28	Reserved			

REG_74

Table 19.342: REG_74, Offset Address: 0x074

Bits	Name	Access	Description	Reset
11:0	reg_hispi_hdr_t2_sol	R/W	HiSPi HDR T2 SOL	0x820
15:12	Reserved			
27:16	reg_hispi_hdr_t2_eol	R/W	HiSPi HDR T2 EOL	0xA20
31:28	Reserved			

REG_80

Table 19.343: REG_80, Offset Address: 0x080

Bits	Name	Access	Description	Reset
7:0	reg_dbg_sel	R/W	Debug signal select	0x0
31:8	Reserved			

19.4 MIPI Tx

19.4.1 Overview

The Display Serial Interface (DSI) interface is a high-speed serial interface defined by the Mobile Industry Processor Interface alliance (MIPI Alliance) and is mainly used for the connection between the processor and the display module. The MIPI Tx interface implements the DSI interface and supports MIPI D-PHY V1.0 serial signal output.

MIPI Tx consists of two parts: analog PHY and digital Controller. The system architecture and functional block diagram are shown in the figure.

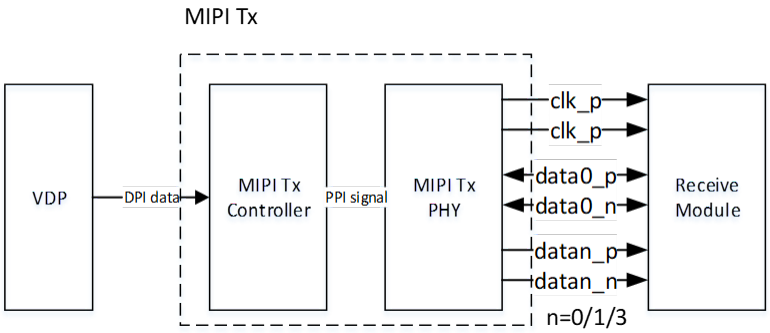


Diagram 19.51: MIPI TX functional block diagram

19.4.2 Features

MIPI Tx has the following features:

- Supports 1/2/4 Data Lane MIPI D-PHY interface, sequential, PN extremely configurable.
- High-speed mode supports up to 2500Mbps/Lane.
- Only Data Lane0 supports low-speed sending and receiving, BTA (Bus Turn-Around) function, and the speed in low-speed mode is up to 10Mbps.
- Support DSI RGB16/18/24/30 bit data type output.
- Supports DSI video mode and command mode, video mode supports Burst mode, Non-burst mode with Sync Events and Non-burst mode with Sync pulses.

19.4.3 Function Description

MIPI Tx includes Tx D-PHY and Tx Controller

- Tx D-PHY supports MIPI D-PHY ver2.1 protocol, which mainly implements the transmission specification of the physical layer.
- The Tx Controller encapsulates the data format according to the MIPI DSI protocol.

19.4.3.1 Tx D-PHY

Tx D-PHY has two working modes, High Speed (HS) and Low Power (LP):

- Video mode data is transmitted through high-speed mode.
- Command mode data is transmitted through low-speed mode.

The data rate range of each lane (Lane) in high-speed mode is 80~2500Mbps, and the maximum rate in low-speed mode is 10Mbps.

High-speed mode supports up to 4 data lanes. The actual number of data lanes used can be 1/2/4, and the order and polarity are configurable. Low-speed mode transmission, reception and Bus Turn-Around (BTA) are only supported by configured data lane0.

19.4.3.2 Tx Controller

19.4.3.3 Sending of Data Packets

When there are multiple high-speed packets to be transmitted, the Tx D-PHY will automatically switch between HS and LP modes according to the high-speed data transmission requirements sent by the Tx Controller. Tx Controller supports whether to send EoT packet (End of Transmission, EoT) at the end of HS transmission.

19.4.3.4 Type of Data

The controller supports the transmission of DSI RGB16/18/24/30 bit. The composition format of various data types is as shown in the figure.

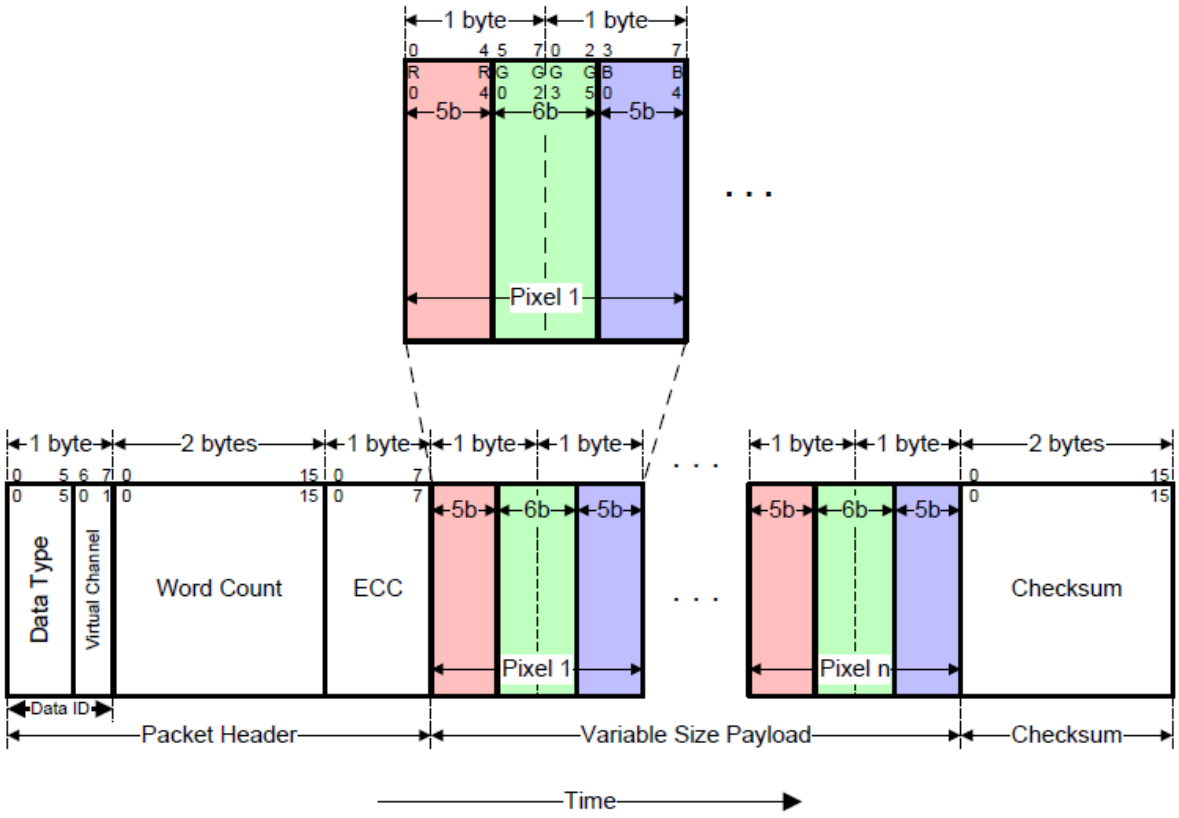


Diagram 19.52: RGB 16-bit format (Data type = 0x0E)

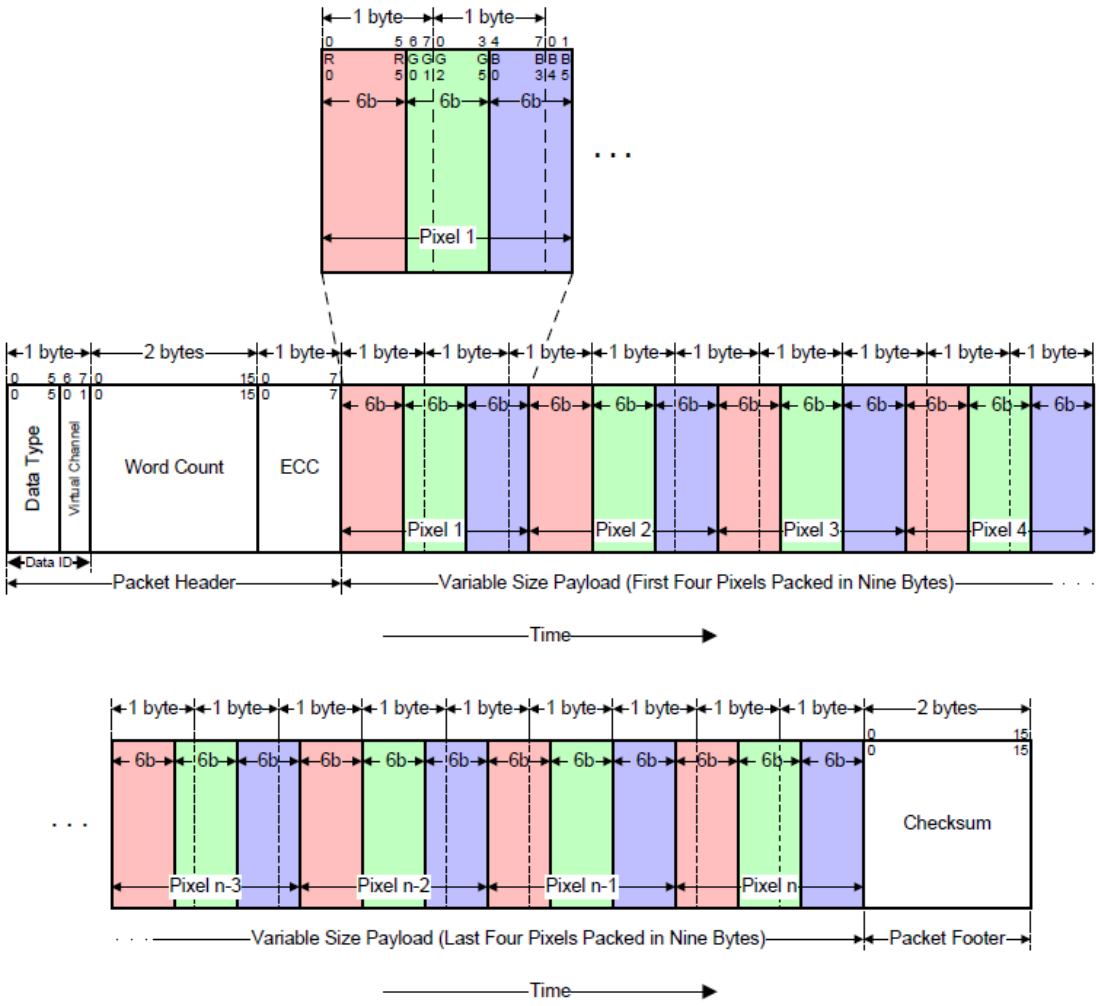


Diagram 19.53: RGB 18-bit format (Data type = 0x1E)

Note: RGB 18-bit only supports data type = 0x0E, does not support loosely Packet mode (data type = 0x2E)

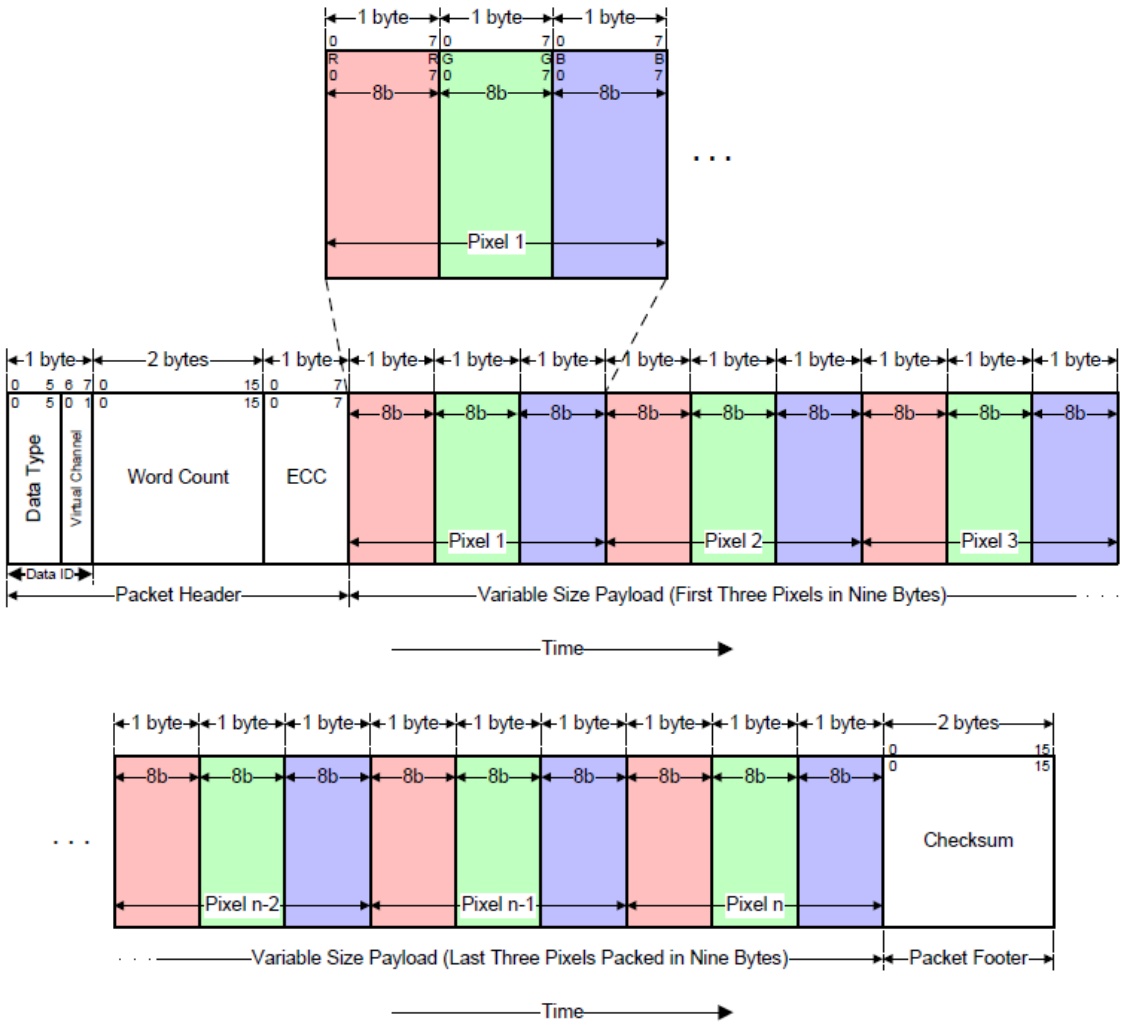


Diagram 19.54: RGB 24-bit format (Data type = 0x3E)

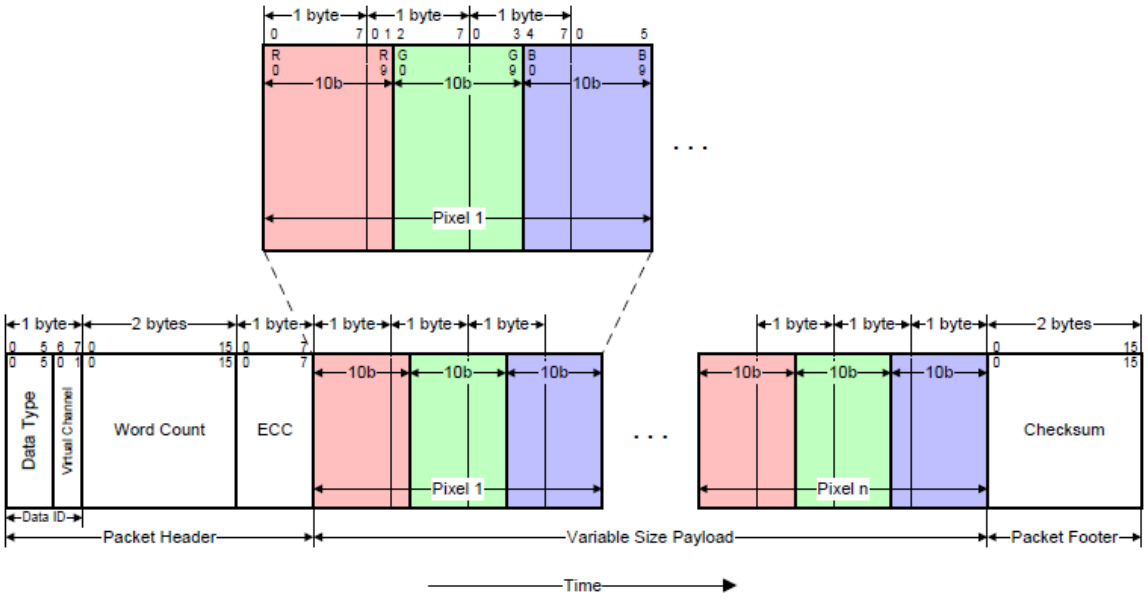


Diagram 19.55: RGB 30-bit format (Data type = 0x0D)

19.4.3.5 Interface Timing

The timing marks are as follows

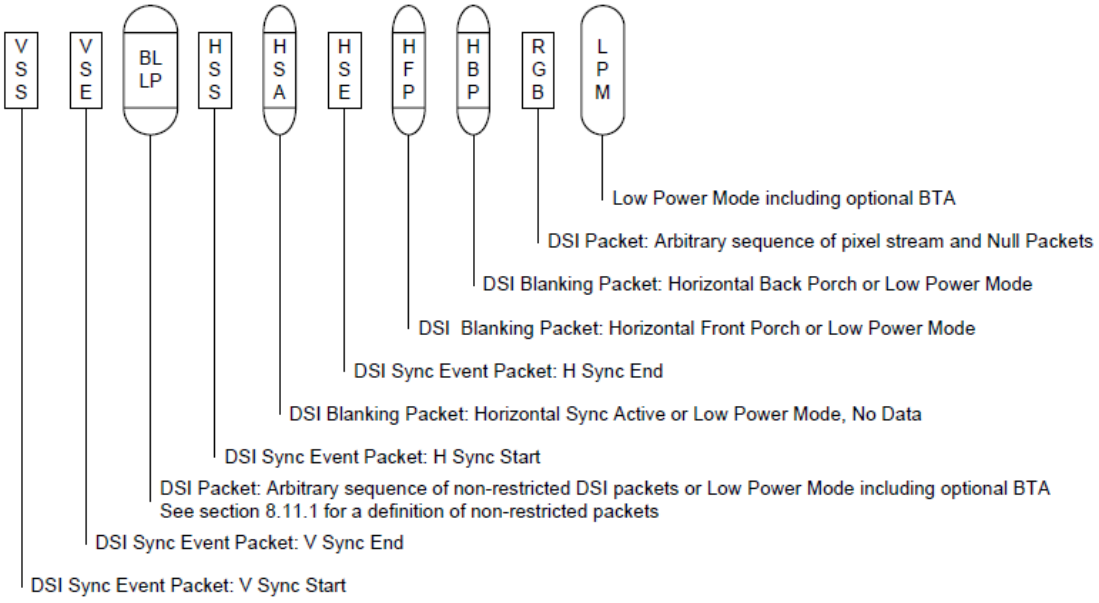


Diagram 19.56: MIPI TX Timing marks

- VSA: Number of frame synchronization steps
- VBP: Number of blanking lines after frame
- VACT: frame valid data line

- VFP: blanking line before frame
- VSS: frame synchronization signal
- HSS: Horizontal Synchronization Signal
- HBP: Post-line blanking area
- RGB: row of valid data
- HFP: Pre-travel blanking zone
- BLLP: Blanking or Low-Power Interval

Video vertical timing is:

Frame synchronization signal (VSS), blanking area after frame (VBP), valid line area (VACT), blanking area before frame (VFP), frame synchronization signal.

The horizontal timing is:

Horizontal sync signal (HSS), line blanking area after (HBP), active pixel (HACT), line blanking area before (HFP), horizontal synchronization signal.

The effective synchronization signal is VSS, HSS, and the effective pixel is the intersection of the effective row area (VACT) and the effective pixel (HACT).

Video mode Burst Transmission timing is as follows:

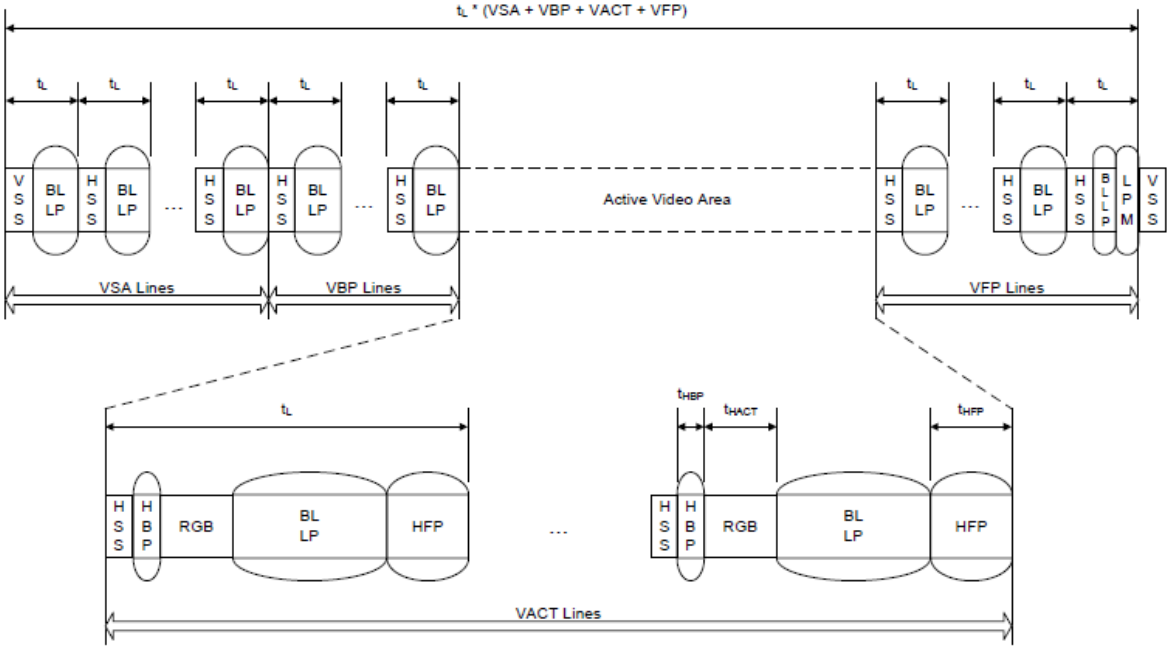


Diagram 19.57: MIPI TX Video mode Burst Transmission timing

MIPI Tx only transmits valid synchronization signals and data, and enters the BLLP area the rest of the time to reduce power consumption.

19.4.4 MIPI Tx Register Overview

MIPI Tx Control register location is 0x0A08A000.

Table 19.344: MIPI Tx Control Registers Overview

Name	Address Offset	Description
DSI_MAC_REG_00	0x000	DSI_MAC_EN
DSI_MAC_REG_01	0x004	HS_REG_00
DSI_MAC_REG_02	0x008	HS_REG_01
DSI_MAC_REG_03	0x00c	ESC_REG
DSI_MAC_REG_04	0x010	ESC_REG_TX0
DSI_MAC_REG_05	0x014	ESC_REG_TX1
DSI_MAC_REG_06	0x018	ESC_REG_TX2
DSI_MAC_REG_07	0x01c	ESC_REG_TX3
DSI_MAC_REG_08	0x020	ESC_REG_RX0
DSI_MAC_REG_09	0x024	ESC_REG_RX1

19.4.5 MIPI Tx Register Description

19.4.5.1 DSI_MAC_REG_00

Table 19.345: DSI_MAC_REG_00, Offset Address: 0x000

Bits	Name	Access	Description	Reset
0	reg_sw_spkt_en	RWS	trigger soft short packet transmission (only work while reg_video_mode=0)	
1	reg_esc_en	RWS	Escape mode enable	
2	reg_video_mode	RWS	HS video mode enable 0: HS command mode or Escape mode 1: HS Video mode	
3	Reserved			
4	reg_sw_spkt_done	R	soft Short packet transmission finish , and clear after reg_sw_spkt_en deassert	
5	reg_esc_done_d0	R	Escape mode finish	
31:6	Reserved			

19.4.5.2 DSI_MAC_REG_01

Table 19.346: DSI_MAC_REG_01, Offset Address: 0x004

Bits	Name	Access	Description	Reset
23:0	reg_sw_spkt	R/W	Soft Short packet for HS mode [7:0] Data ID [15:8]: Data 0 [23:16] Data 1 (* ECC is generated by HW)	0x0
25:24	reg_lane_mode	R/W	lane active setting while HS mode 0: D0 Lane active 1: D0/D1 Lane Active 2: D0/D1/D2/D3 Lane Active 3: reserved	0x1
26	reg_eotp_en	R/W	enable insert eopt short packet followed each transmission	0x1
27	reg_skew_cal	R/W	skew calibration	0x0
28	reg_skew_cal_ini	R/W	skew calibration initialize	0x0
29	reg_hs_c_conti	R/W	c lane continous mode	0x1
31:30	reg_rgb_clr_fmt	R/W	RGB Packed format 0: RGB888 1: RGB666 2: RGB565 3: RGB10-10-10	0x0

19.4.5.3 DSI_MAC_REG_02

Table 19.347: DSI_MAC_REG_02, Offset Address: 0x008

Bits	Name	Access	Description	Reset
15:0	reg_lpkt_wc	R/W	Video line data paket byte counts (packed data must be byte alignment)	0x0
26:16	reg_event_delay	R/W	time shift for each video packet event (unit: pixel time)	0x80
31:27	Reserved			

19.4.5.4 DSI_MAC_REG_03

Table 19.348: DSI_MAC_REG_03, Offset Address: 0x00c

Bits	Name	Access	Description	Reset
2:0	reg_esc_mode	R/W	Escape mode 0: TX trigger 1: TX LPDT 2: ULPS 3: ULPS EXIT 4: RX 5: reserved 6: C ULPS 7: C ULPS EXIT	0x1
3	Reserved			
7:4	reg_esc_trig	R/W	Trigger command for Escape TX trigger mode	0x0
11:8	reg_tx_bc	R/W	TX LPDT byte counts 0: 1 byte 1: 2 bytes ..... 15: 16 bytes	0x0
15:12	reg_bta_rx_bc	R	Byte counts for RX received 0: no received 1: 1 byte ..... 8: 8 bytes	
31:16	Reserved			

19.4.5.5 DSI_MAC_REG_04

Table 19.349: DSI_MAC_REG_04, Offset Address: 0x010

Bits	Name	Access	Description	Reset
7:0	reg_tx_byte0	R/W	TX byte 0	0x0
15:8	reg_tx_byte1	R/W	TX byte 1	0x0
23:16	reg_tx_byte2	R/W	TX byte 2	0x0
31:24	reg_tx_byte3	R/W	TX byte 3	0x0

19.4.5.6 DSI_MAC_REG_05

Table 19.350: DSI_MAC_REG_05, Offset Address: 0x014

Bits	Name	Access	Description	Reset
7:0	reg_tx_byte4	R/W	TX byte 4	0x0
15:8	reg_tx_byte5	R/W	TX byte 5	0x0
23:16	reg_tx_byte6	R/W	TX byte 6	0x0
31:24	reg_tx_byte7	R/W	TX byte 7	0x0

19.4.5.7 DSI_MAC_REG_06

Table 19.351: DSI_MAC_REG_06, Offset Address: 0x018

Bits	Name	Access	Description	Reset
7:0	reg_tx_byte8	R/W	TX byte 8	0x0
15:8	reg_tx_byte9	R/W	TX byte 9	0x0
23:16	reg_tx_bytea	R/W	TX byte 10	0x0
31:24	reg_tx_byteb	R/W	TX byte 11	0x0

19.4.5.8 DSI_MAC_REG_07

Table 19.352: DSI_MAC_REG_07, Offset Address: 0x01c

Bits	Name	Access	Description	Reset
7:0	reg_tx_bytec	R/W	TX byte 12	0x0
15:8	reg_tx_byted	R/W	TX byte 13	0x0
23:16	reg_tx_bytee	R/W	TX byte 14	0x0
31:24	reg_tx_bytef	R/W	TX byte 15	0x0

19.4.5.9 DSI_MAC_REG_08

Table 19.353: DSI_MAC_REG_08, Offset Address: 0x020

Bits	Name	Access	Description	Reset
7:0	reg_bta_rx_byte0	R	RX byte 0	
15:8	reg_bta_rx_byte1	R	RX byte 1	
23:16	reg_bta_rx_byte2	R	RX byte 2	
31:24	reg_bta_rx_byte3	R	RX byte 3	

19.4.5.10 DSI_MAC_REG_09

Table 19.354: DSI_MAC_REG_09, Offset Address: 0x024

Bits	Name	Access	Description	Reset
7:0	reg_bta_rx_byte4	R	RX byte 4	
15:8	reg_bta_rx_byte5	R	RX byte 5	
23:16	reg_bta_rx_byte6	R	RX byte 6	
31:24	reg_bta_rx_byte7	R	RX byte 7	

19.4.6 MIPI Tx PHY Register Overview

MIPI Tx PHY register location is 0x0A0D_1000.

Table 19.355: MIPI Tx PHY Registers Overview

Name	Address Offset	Description
REG_00	0x000	DSI PHY REG_00
REG_01	0x004	DSI PHY REG_01
REG_02	0x008	DSI PHY REG_02
REG_03	0x00c	DSI PHY REG_03
REG_04	0x010	DSI PHY REG_04
REG_05	0x014	DSI PHY REG_05
REG_23	0x08c	DSI PHY REG_23
REG_24	0x090	DSI PHY REG_24
REG_25	0x094	DSI PHY REG_25
REG_26	0x098	DSI PHY REG_26
REG_27	0x09c	DSI PHY REG_27
REG_28	0x0a0	DSI PHY REG_28
REG_2D	0x0b4	DSI PHY REG_2D

19.4.7 MIPI Tx PHY Register Description

19.4.7.1 REG_00

Table 19.356: REG_00, Offset Address: 0x000

Bits	Name	Access	Description	Reset
0	reg_clk_lane_en	R/W	MIPI Tx clock Lane Enable	0x0
1	reg_dat_0_lane_en	R/W	MIPI Tx data Lane 0 enable	0x0
2	reg_dat_1_lane_en	R/W	MIPI Tx data Lane 1 enable	0x0
3	reg_dat_2_lane_en	R/W	MIPI Tx data Lane 2 enable	0x0
4	reg_dat_3_lane_en	R/W	MIPI Tx data Lane 3 enable	0x0
31:5	Reserved			

19.4.7.2 REG_01

Table 19.357: REG_01, Offset Address: 0x004

Bits	Name	Access	Description	Reset
7:0	reg_t_clk_prepare	R/W	CLK-PREPARE timing parameter, unit : 8UI	0x5
15:8	reg_t_clk_zero	R/W	CLK-ZERO timing parameter, unit : 8UI	0x24
23:16	reg_t_clk_pre	R/W	CLK-PRE timing parameter, unit : 8UI	0x8
31:24	reg_t_clk_post	R/W	CLK-POST timing parameter, unit : 8UI	0x20

19.4.7.3 REG_02

Table 19.358: REG_02, Offset Address: 0x008

Bits	Name	Access	Description	Reset
7:0	reg_t_clk_trail	R/W	CLK-TRAIL timing parameter, unit : 8UI	0x1
31:8	Reserved			

19.4.7.4 REG_03

Table 19.359: REG_03, Offset Address: 0x00c

Bits	Name	Access	Description	Reset
19:0	reg_t_esc_init	R/W	ESC-INIT timing parameter, unit : T_LPX	0x1000
31:20	Reserved			

19.4.7.5 REG_04

Table 19.360: REG_04, Offset Address: 0x010

Bits	Name	Access	Description	Reset
19:0	reg_t_esc_wakeup	R/W	ESC-WAKEUP timing parameter, unit : T_LPX	0x4e20
31:20	Reserved			

19.4.7.6 REG_05

Table 19.361: REG_05, Offset Address: 0x014

Bits	Name	Access	Description	Reset
7:0	reg_t_hs_pre_on	R/W	HS-PRE-ON timing parameter, unit : 8UI	0x1
15:8	reg_t_hs_prepare	R/W	HS-PREPARE timing parameter, unit : 8UI	0x6
23:16	reg_t_hs_zero	R/W	HS-ZERO timing parameter, unit : 8UI	0x20
31:24	reg_t_hs_trail	R/W	HS-TRAIL timing parameter, unit : 8UI	0x1

19.4.7.7 REG_23

Table 19.362: REG_23, Offset Address: 0x08c

Bits	Name	Access	Description	Reset
0	Reserved			
1	reg_en_ssc	R/W	MIPI Tx Spred Spectrum Enable	0x0
3:2	reg_ssc_mode	R/W	Spread Spectrum Mode	0x0
31:4	Reserved			

19.4.7.8 REG_24

Table 19.363: REG_24, Offset Address: 0x090

Bits	Name	Access	Description	Reset
31:0	reg_set	R/W	Frequency Synthesizer divider, format 6.26	0x11F514F9

19.4.7.9 REG_25

Table 19.364: REG_25, Offset Address: 0x094

Bits	Name	Access	Description	Reset
15:0	reg_span	R/W	SSC Span parameter	0x0
31:16	Reserved			

19.4.7.10 REG_26

Table 19.365: REG_26, Offset Address: 0x098

Bits	Name	Access	Description	Reset
23:0	reg_step	R/W	SSC Step parameter	0x0
31:24	Reserved			

19.4.7.11 REG_27

Table 19.366: DMAC_IREG_27DREG, Offset Address: 0x09c

Bits	Name	Access	Description	Reset
2:0	reg_dsi_lane_0_sel	R/W	MIPI Tx PAD0 Config 0 : clock lane 1 : Data Lane 0 2 : Data Lane 1 3 : Data Lane 2 4 : Data Lane 3	0x1
3	Reserved			
6:4	reg_dsi_lane_1_sel	R/W	MIPI Tx PAD1 Config 0 : clock lane 1 : Data Lane 0 2 : Data Lane 1 3 : Data Lane 2 4 : Data Lane 3	0x2
7	Reserved			
10:8	reg_dsi_lane_2_sel	R/W	MIPI Tx PAD2 Config 0 : clock lane 1 : Data Lane 0 2 : Data Lane 1 3 : Data Lane 2 4 : Data Lane 3	0x0
11	Reserved			
14:12	reg_dsi_lane_3_sel	R/W	MIPI Tx PAD3 Config 0 : clock lane 1 : Data Lane 0 2 : Data Lane 1 3 : Data Lane 2 4 : Data Lane 3	0x3
15	Reserved			
18:16	reg_dsi_lane_4_sel	R/W	MIPI Tx PAD4 Config 0 : clock lane 1 : Data Lane 0 2 : Data Lane 1 3 : Data Lane 2 4 : Data Lane 3	0x4
23:19	Reserved			
28:24	reg_sel_clk_lane	R/W	MIPI Tx Clock lane Config 5'h01 : PAD0 as MIPI Tx Clock Lane 5'h02 : PAD1 as MIPI Tx Clock Lane 5'h04 : PAD2 as MIPI Tx Clock Lane 5'h08 : PAD3 as MIPI Tx Clock Lane 5'h10 : PAD4 as MIPI Tx Clock Lane	0x4
31:29	Reserved			

19.4.7.12 REG_28

Table 19.367: REG_28, Offset Address: 0x0a0

Bits	Name	Access	Description	Reset
0	reg_dsi_lane_0_pn_swap	R/W	MIPI Tx PAD0 PN swap enable	0x0
1	reg_dsi_lane_1_pn_swap	R/W	MIPI Tx PAD1 PN swap enable	0x0
2	reg_dsi_lane_2_pn_swap	R/W	MIPI Tx PAD2 PN swap enable	0x0
3	reg_dsi_lane_3_pn_swap	R/W	MIPI Tx PAD3 PN swap enable	0x0
31:4	Reserved			

19.4.7.13 REG_2D

Table 19.368: REG_2D, Offset Address: 0x0b4

Bits	Name	Access	Description	Reset
0	reg_lvds_en	R/W	MIPI Tx modue in LVDS Tx mode	0x0
31:1	Reserved			

CHAPTER

TWENTY

AUDIO INTERFACE

20.1 AIAO

20.1.1 Overview

The audio input/output interface (Audio Input/Audio Output) is used to interface with the chip's built-in Audio Codec or the chip's external Audio Codec and digital microphone to complete the sending and receiving of audio data and realize functions such as recording, playback, and intercom. The chip integrates AIAO related modules into a subsystem, and the built-in Audio Codec ADC/DAC supports stereo input and output. It supports two sets of I2S interfaces externally, and integrates 4 sets of I2S TX/RX modules internally, which can receive and send audio data at the same time, and can support the simultaneous sending and receiving of multi-channel data. The basic module block diagram is shown below:

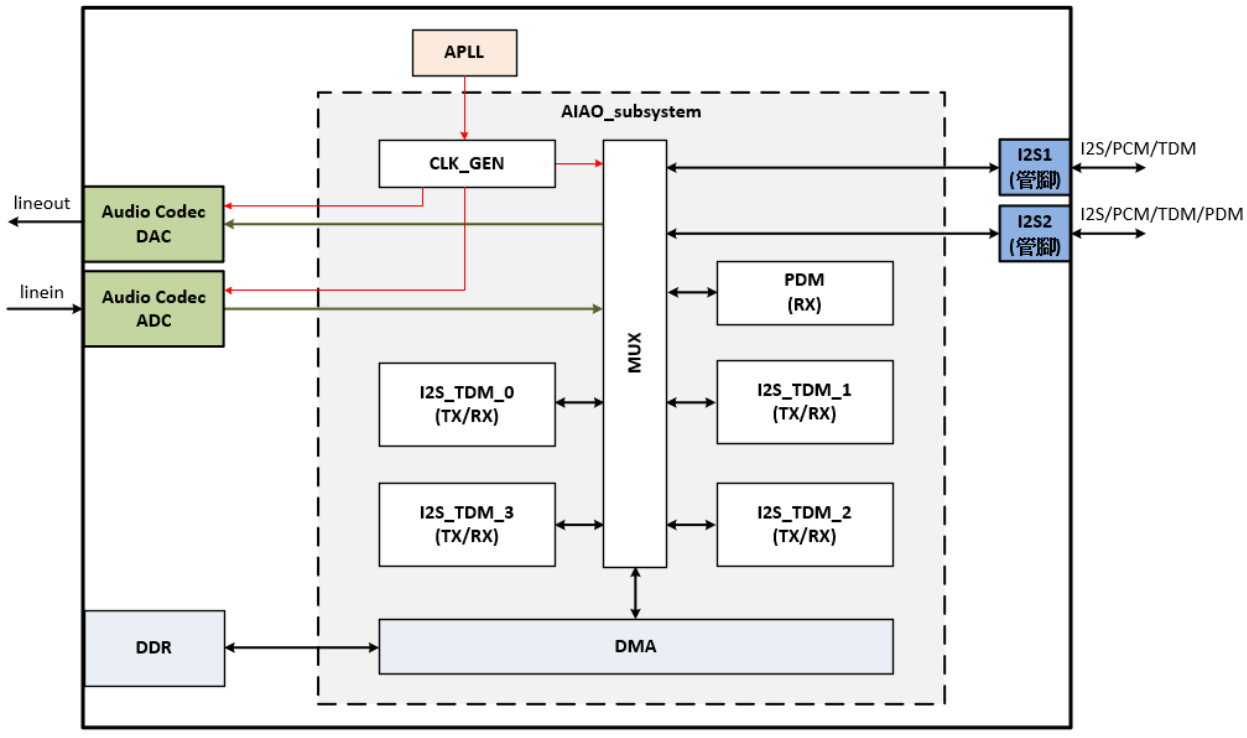


Diagram 20.1: AIAO block diagram

20.1.2 Features

The AIAO interface supports Master-mode, Slave-mode I2S and PCM modes, and supports multi-channel TDM mode. Receive and send audio data to access the DDR space through DMA. The specific features are as follows:

- Highly flexible and configurable timing parameters, frame period, frame synchronization signal duration and polarity are all configurable
- Configurable clock edges for signal generation and sampling
- Supports master mode and slave mode stereo I2S mode audio data sending and receiving
- Supports transmission and reception of audio data in master mode and slave mode mono and stereo PCM mode
- Supports sending and receiving of multi-channel TDM mode audio data in master mode and slave mode
- Receive and send can be enabled individually or simultaneously
- Data adopts DMA operation and can be accessed circularly through the buffer developed by the software.

20.1.2.1 PCM Interface

- Supports 16-bit linear PCM encoding for sending and receiving
- The PCM interface frame synchronization signal supports short pulses (1 clock cycle) and long pulses (the number of clock cycles is configurable)
- Interface timing supports standard mode and left-aligned mode

20.1.2.2 I2S Interface

- Supports 16/24-bit data sending and receiving
- Supports 8kHz ~ 192kHz sampling rate
- The I2S interface frame synchronization signal can support low-level left channel or high-level left channel
- Interface timing supports standard mode and left-aligned mode

20.1.3 Function Description

The AIAO subsystem connects the built-in Audio codec, I2S pins and TX/RX modules through the internal PIN-MUX, and configures the registers appropriately according to the application requirements through software to achieve different connection methods.

20.1.3.1 Typical Application

Typical applications are as follows:

- Supports I2S slave mode to connect to the built-in Audio Codec ADC, or I2S/PCM/TDM master/slave mode to connect to an external ADC for audio collection.
- Supports I2S master mode to connect to the built-in Audio Codec DAC, or I2S/PCM/TDM master/slave mode to connect to an external DAC for audio playback.

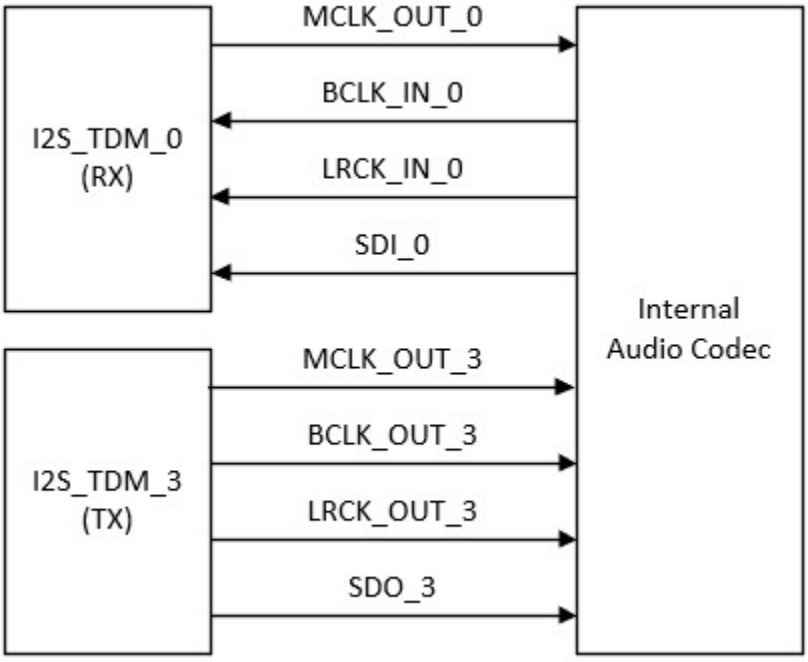


Diagram 20.2: Schematic diagram of connection with built-in Audio Codec through I2S interface

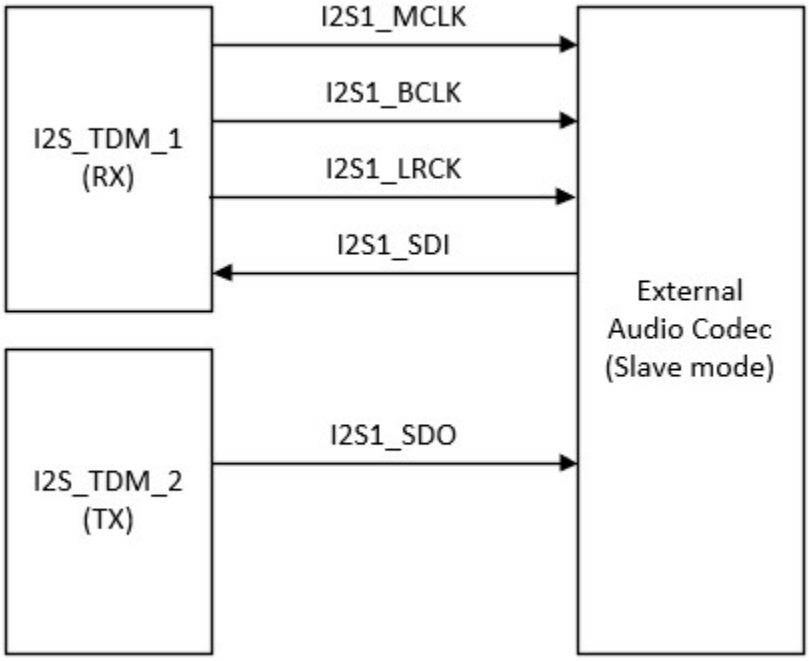


Diagram 20.3: Schematic diagram of connecting AIAO to an external Audio Codec through the I2S interface in master mode

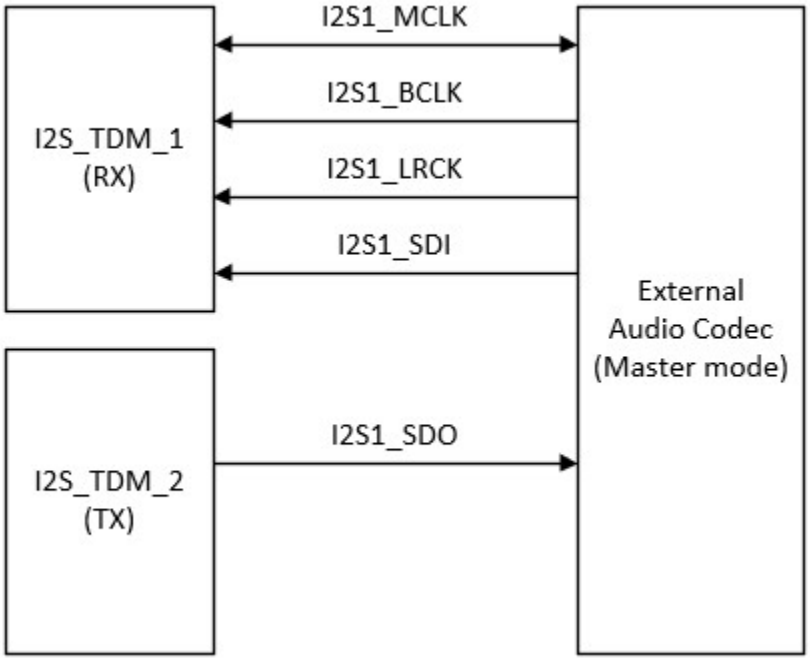


Diagram 20.4: Schematic diagram of connecting AIAO to an external Audio Codec through the I2S interface in slave mode

20.1.3.2 Functional Principle

The audio source is Analog-to-Digital converted into audio data through the built-in or external Audio Codec ADC, which is received by the connected RX module through the I2S or PCM interface, and stored in the circular buffer via DMA, and then taken out by the CPU for storage, thus Complete the recording function. The TX module reads audio data from the circular buffer through DMA, and transmits the audio data to the connected built-in or external Audio through the I2S or PCM interface. Codec DAC, performs Digital-to-Analog conversion for audio source playback.

When connecting to an external I2S interface, the supported I2S timing is as shown in the diagram *I2S interface timing*.

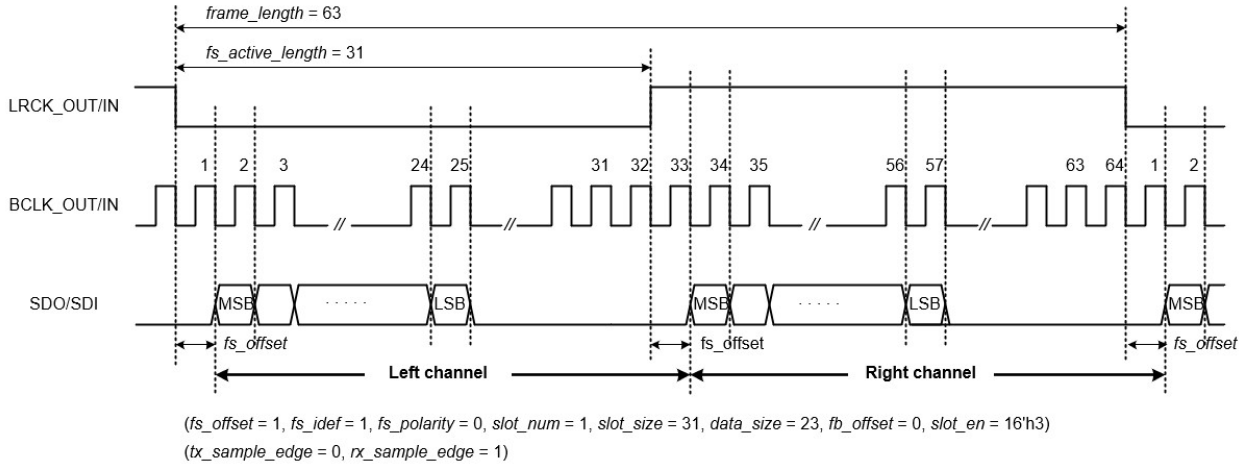


Diagram 20.5: I2S interface timing

Chart *I2S interface timing* takes the audio data width 24-bit as an example. The data is transmitted in MSB First mode. The MSB is delayed by one BCLK cycle relative to the LRCK signal. The data and LRCK signal are sent out using the falling edge of BCLK and sampled on the rising edge of BCLK. (tx_sample_edge = 0, rx_sample_edge = 1).

When connecting to the external PCM interface, it supports PCM standard timing and data left-aligned timing. The standard mode timing is as shown in the diagram *PCM interface standard mode timing*, and the left-aligned mode timing is as shown in the diagram *PCM Interface Left Justified Mode Timing*.

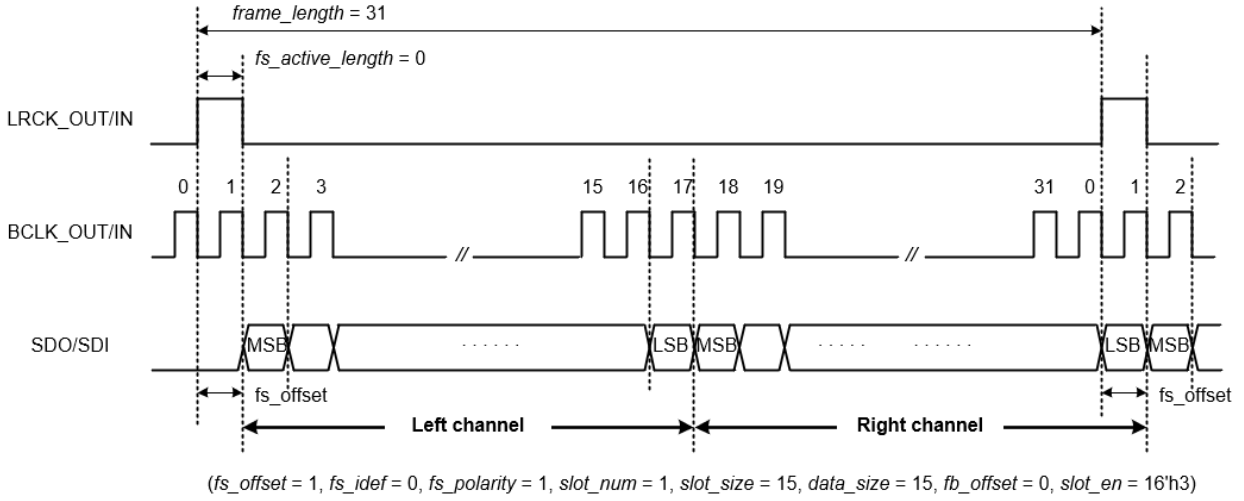


Diagram 20.6: PCM interface standard mode timing

Chart *PCM interface standard mode timing* takes the stereo audio data width of 16-bit as an example. The data is transmitted in MSB First mode. The MSB is delayed by one BCLK cycle relative to the LRCK signal. The data and LRCK signal are sent out using the falling edge of BCLK. On the rising edge of BCLK sample(tx_sample_edge = 0, rx_sample_edge = 1).

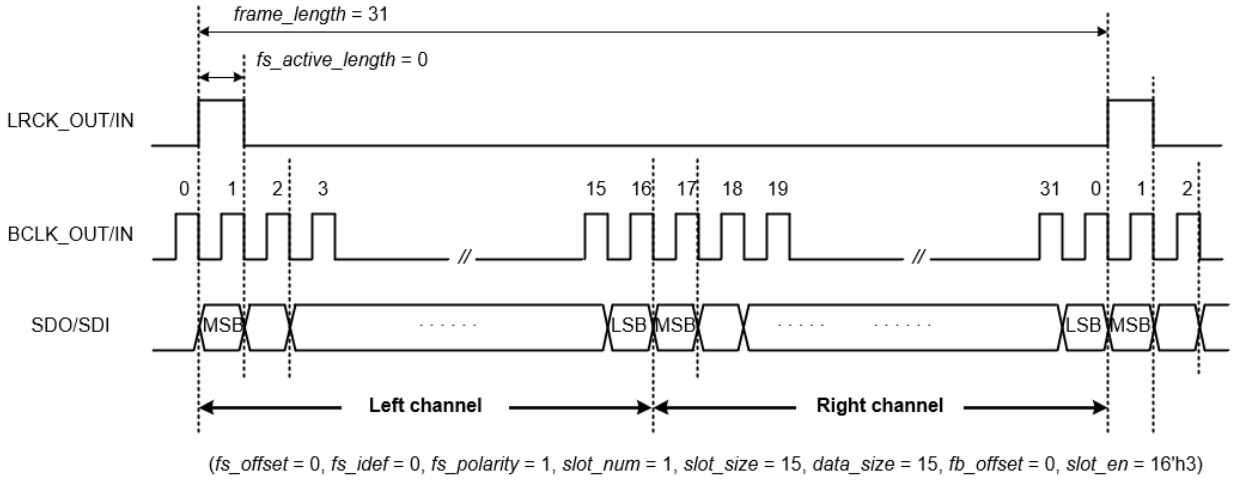


Diagram 20.7: PCM Interface Left Justified Mode Timing

In left-justified mode, the data and LRCK signals are sent out at the beginning of the same beat.

20.1.4 Way of Working

The Audio Codec integrated within AIAO and the external Audio Codec connected through the I2S interface can work simultaneously through software configuration. The connection method is as shown in the figure: *Schematic diagram of connection with built-in Audio Codec through I2S interface*, *Schematic diagram of connecting AIAO to an external Audio Codec through the I2S interface in master mode* and *Schematic diagram of connecting AIAO to an external Audio Codec through the I2S interface in slave mode*. Before enabling data transmission according to application needs, the software must first configure the AIAO subsystem registers i2s_tdm_sclk_in_sel, i2s_tdm_fs_in_sel, i2s_tdm_sdi_in_sel, i2s_tdm_sdo_out_sel to connect each interface to the corresponding TX/RX module (I2S_TDM_0~I2S_TDM_3).

20.1.4.1 Clock Gating and Clock Configuration

If AIAO operates in master mode, you must first set the TX/RX module register master_mode as the MCLK/BCLK clock source to 1, configure the frequency division register I2S_CLK_CTRL1 (mclk_div, bclk_div) depending on the sampling rate, and then set the register I2S_CLK_CTRL0 (aud_en) is 1, turns on clock gating.

20.1.4.2 Soft Reset

The four TX/RX modules integrated by AIAO all have independent soft resets. Before enabling the TX/RX modules for data transmission, the registers FIFO_RESET and I2S_RESET must be configured for soft reset.

20.1.5 AIAO Register Overview

An overview of AIAO subsystem registers is shown in table *AIAO Registers Overview (Base Address: 0x0410_8000)*.

Table 20.1: AIAO Registers Overview (Base Address: 0x0410_8000)

Name	Address Offset	Description
i2s_tdm_sclk_in_sel	0x000	Select the Source of the Mode TX/RX Module SCLK Source
i2s_tdm_fs_in_sel	0x004	Select the Synchronous signal Source of the Mode TX/RX Module
i2s_tdm_sdi_in_sel	0x008	Select the RX module SDI signal source
i2s_tdm_sdo_out_sel	0x00c	Select the subsystem SDO output source
i2s_bclk_oen_sel	0x030	BCLK IO output control
audio_pdm_ctrl	0x040	Enable PDM mode
i2s_sys_int_en	0x060	Enable the I2S subsystem interrupt signal
i2s_sys_ints	0x064	I2S subsystem interrupt signal status

The I2S_TDM_0 ~ I2S_TDM_3 module register overview is shown in the table *I2S_TDM_0/1/2/3 Registers Overview (Base Address: 0x0410_0000 + n*0x10000)*.

Table 20.2: I2S_TDM_0/1/2/3 Registers Overview (Base Address: 0x0410_0000 + n*0x10000)

Name	Address Offset	Description
BLK_MODE_SETTING	0x000	TX/RX module operation control
FRAME_SETTING	0x004	Audio frame timing control
SLOT_SETTING1	0x008	Classic and Data Control
SLOT_SETTING2	0x00c	Enable channels
DATA_FORMAT	0x010	Set the storage data format
BLK_CFG	0x014	TX/RX module function configuration
I2S_ENABLE	0x018	Enable TX/RX module
I2S_RESET	0x01c	TX/RX module reset
I2S_INT_EN	0x020	Enable interruption
I2S_INT	0x024	Interrupt status
FIFO_THRESHOLD	0x028	FIFO threshold
I2S_LRCK_MASTER	0x02c	Make the main mode of LRCK continuous output mode
FIFO_RESET	0x030	FIFO reset
RX_STATUS	0x040	RX module internal state register
TX_STATUS	0x048	TX module internal state register
I2S_CLK_CTRL0	0x060	Clock Control
I2S_CLK_CTRL1	0x064	The main mode of the clock frequency
I2S_PCM_SYNTH	0x068	PCM frame synchronization signal synthesis mode control
RX_RD_PORT	0x080	RX FIFO read port
TX_WR_PORT	0x0c0	TX FIFO write port

20.1.6 AIAO Register Description

20.1.6.1 AIAO Subsystem Register Description

i2s_tdm_sclk_in_sel

Select sclk source

Table 20.3: i2s_tdm_sclk_in_sel, Offset Address: 0x000

Bits	Name	Access	Description	Reset
2:0	i2s_tdm_0_sclk_in_sel	R/W	Selects SCLK input source when i2s_tdm_0 operates in slave mode 3'b000 = Reserved 3'b001 = From i2s_tdm_1 BCLK_out 3'b010 = From i2s_tdm_2 BCLK_out 3'b011 = From i2s_tdm_3 BCLK_out 3'b100 = From internal Audio Codec ADC BCLK 3'b101 = From IO I2S1_BCLK 3'b110 = From IO I2S2_BCLK 3'b111 = Reserved	0x4
3	Reserved			
6:4	i2s_tdm_1_sclk_in_sel	R/W	Selects SCLK input source when i2s_tdm_1 operates in slave mode 3'b000 = From i2s_tdm_0 BCLK_out 3'b001 = From audio_pdm module SCLK_out 3'b010 = From i2s_tdm_2 BCLK_out 3'b011 = From i2s_tdm_3 BCLK_out 3'b100 = From internal Audio Codec ADC BCLK 3'b101 = From IO I2S1_BCLK 3'b110 = From IO I2S2_BCLK 3'b111 = Reserved	0x5
7	Reserved			
10:8	i2s_tdm_2_sclk_in_sel	R/W	Selects SCLK input source when i2s_tdm_2 operates in slave mode 3'b000 = From i2s_tdm_0 BCLK_out 3'b001 = From i2s_tdm_1 BCLK_out 3'b010 = Reserved 3'b011 = From i2s_tdm_3 BCLK_out 3'b100 = From internal Audio Codec ADC BCLK 3'b101 = From IO I2S1_BCLK 3'b110 = From IO I2S2_BCLK 3'b111 = Reserved	0x6
11	Reserved			
14:12	i2s_tdm_3_sclk_in_sel	R/W	Selects SCLK input source when i2s_tdm_3 operates in slave mode 3'b000 = From i2s_tdm_0 BCLK_out 3'b001 = From i2s_tdm_1 BCLK_out 3'b010 = From i2s_tdm_2 BCLK_out 3'b011 = Reserved 3'b100 = From internal Audio Codec ADC BCLK 3'b101 = From IO I2S1_BCLK 3'b110 = From IO I2S2_BCLK 3'b111 = Reserved	0x7
31:15	Reserved			

i2s_tdm_fs_in_sel

Table 20.4: i2s_tdm_fs_in_sel, Offset Address: 0x004

Bits	Name	Access	Description	Reset
2:0	i2s_tdm_0_fs_in_sel	R/W	Select LRCK/FS input source for i2s_tdm_0 operation in slave mode 3'b000 = Reserved 3'b001 = From i2s_tdm_1 lrck_out 3'b010 = From i2s_tdm_2 lrck_out 3'b011 = From i2s_tdm_3 lrck_out 3'b100 = From internal Audio Codec ADC LRCK 3'b101 = From IO I2S1_LRCK 3'b110 = From IO I2S2_LRCK 3'b111 = Reserved	0x4
3	Reserved			
6:4	i2s_tdm_1_fs_in_sel	R/W	Select LRCK/FS input source for i2s_tdm_1 operation in slave mode 3'b000 = From i2s_tdm_0 lrck_out 3'b001 = From audio_pdm module lrck_out 3'b010 = From i2s_tdm_2 lrck_out 3'b011 = From i2s_tdm_3 lrck_out 3'b100 = From internal Audio Codec ADC LRCK 3'b101 = From IO I2S1_LRCK 3'b110 = From IO I2S2_LRCK 3'b111 = Reserved	0x5
7	Reserved			
10:8	i2s_tdm_2_fs_in_sel	R/W	Select LRCK/FS input source for i2s_tdm_2 operation in slave mode 3'b000 = From i2s_tdm_0 lrck_out 3'b001 = From i2s_tdm_1 lrck_out 3'b010 = Reserved 3'b011 = From i2s_tdm_3 lrck_out 3'b100 = From internal Audio Codec ADC LRCK 3'b101 = From IO I2S1_LRCK 3'b110 = From IO I2S2_LRCK 3'b111 = Reserved	0x6
11	Reserved			
14:12	i2s_tdm_3_fs_in_sel	R/W	Select LRCK/FS input source for i2s_tdm_3 operation in slave mode 3'b000 = From i2s_tdm_0 lrck_out 3'b001 = From i2s_tdm_1 lrck_out 3'b010 = Reserved 3'b011 = From i2s_tdm_3 lrck_out 3'b100 = From internal Audio Codec ADC LRCK 3'b101 = From IO I2S1_LRCK 3'b110 = From IO I2S2_LRCK 3'b111 = Reserved	0x7
31:15	Reserved			

i2s_tdm_sdi_in_sel

Table 20.5: i2s_tdm_sdi_in_sel, Offset Address: 0x008

Bits	Name	Access	Description	Reset
2:0	i2s_tdm_0_sdi_in_sel	R/W	Select i2s_tdm_0 RX Module SDI Signal Source 3'b000 = Reserved 3'b001 = From i2s_tdm_1 sdo 3'b010 = From i2s_tdm_2 sdo 3'b011 = From i2s_tdm_3 sdo 3'b100 = From Internal Audio Codec ADC SDO 3'b101 = From IO I2S1_SDI 3'b110 = From IO I2S2_SDI 3'b111 = Reserved	0x4
3	Reserved			
6:4	i2s_tdm_1_sdi_in_sel	R/W	Select i2s_tdm_1 RX Module SDI Signal Source 3'b000 = From i2s_tdm_0 sdo 3'b001 = From Audio PDM Module sdo 3'b010 = From i2s_tdm_2 sdo 3'b011 = From i2s_tdm_3 sdo 3'b100 = From Internal Audio Codec ADC SDO 3'b101 = From IO I2S1_SDI 3'b110 = From IO I2S2_SDI 3'b111 = Reserved	0x5
7	Reserved			
10:8	i2s_tdm_2_sdi_in_sel	R/W	Select i2s_tdm_2 RX Module SDI Signal Source 3'b000 = From i2s_tdm_0 sdo 3'b001 = From i2s_tdm_1 sdo 3'b010 = Reserved 3'b011 = From i2s_tdm_3 sdo 3'b100 = From Internal Audio Codec ADC SDO 3'b101 = From IO I2S1_SDI 3'b110 = From IO I2S2_SDI 3'b111 = Reserved	0x6
11	Reserved			
14:12	i2s_tdm_3_sdi_in_sel	R/W	Select i2s_tdm_3 RX Module SDI Signal Source 3'b000 = From i2s_tdm_0 sdo 3'b001 = From i2s_tdm_1 sdo 3'b010 = From i2s_tdm_2 sdo 3'b011 = Reserved 3'b100 = From Internal Audio Codec DAC SDO 3'b101 = From IO I2S1_SDI 3'b110 = From IO I2S2_SDI 3'b111 = Reserved	0x7
31:15	Reserved			

i2s_tdm_sdo_out_sel

Table 20.6: i2s_tdm_sdo_out_sel, Offset Address: 0x00c

Bits	Name	Access	Description	Reset
2:0	i2s_tdm_0_sdo_out_sel	R/W	Only allowed to configure as default value 0x4	0x4
3	Reserved			
6:4	i2s_tdm_1_sdo_out_sel	R/W	Select I2S1_SDO Signal Source 3'b100 = I2S1_SDO from i2s_tdm_0 TX Module sdo 3'b101 = I2S1_SDO from i2s_tdm_1 TX Module sdo 3'b110 = I2S1_SDO from i2s_tdm_2 TX Module sdo 3'b111 = I2S1_SDO from i2s_tdm_3 TX Module sdo Other values = Reserved	0x5
7	Reserved			
10:8	i2s_tdm_2_sdo_out_sel	R/W	Select I2S2_SDO Signal Source 3'b100 = I2S2_SDO from i2s_tdm_0 TX Module sdo 3'b101 = I2S2_SDO from i2s_tdm_1 TX Module sdo 3'b110 = I2S2_SDO from i2s_tdm_2 TX Module sdo 3'b111 = I2S2_SDO from i2s_tdm_3 TX Module sdo Other values = Reserved	0x6
11	Reserved			
14:12	i2s_tdm_3_sdo_out_sel	R/W	Only allowed to configure as default value 0x7 Internal Audio Codec DAC SDI fixed from i2s_tdm_3 TX Module sdo	0x7
31:15	Reserved			

i2s_bclk_oen_sel

Table 20.7: i2s_bclk_oen_sel, Offset Address: 0x030

Bits	Name	Access	Description	Reset
0	i2s0_bclk_oen_sel	R/W	Reserved	0x0
1	i2s1_bclk_oen_sel	R/W	Select I2S1_BCLK IO OEN Control Mode 0 = Controlled by i2s_tdm_1 TX Module 1 = Controlled by Register i2s1_bclk_oen_ext	0x0
2	i2s2_bclk_oen_sel	R/W	Select I2S2_BCLK IO OEN Control Mode 0 = Controlled by i2s_tdm_2 TX Module 1 = Controlled by Register i2s2_bclk_oen_ext	0x0
3	i2s3_bclk_oen_sel	R/W	Only allowed to configure as default value 0x0	0x0
7:4	Reserved			
8	i2s0_bclk_oen_ext	R/W	Reserved	0x0
9	i2s1_bclk_oen_ext	R/W	Controlled by Sub-Register to Control I2S1_BCLK IO OEN 0 = Disable IO Output 1 = Enable IO Output	0x0
10	i2s2_bclk_oen_ext	R/W	Controlled by Sub-Register to Control I2S2_BCLK IO OEN 0 = Disable IO Output 1 = Enable IO Output	0x0
11	i2s3_bclk_oen_ext	R/W	Only allowed to configure as default value 0x0	0x0
15:12	Reserved			
16	i2s_bclk_oen_no_delay	R/W	Only allowed to configure as default value 0x0	0x0
31:17	Reserved			

audio_pdm_ctrl

Table 20.8: audio_pdm_ctrl, Offset Address: 0x040

Bits	Name	Access	Description	Reset
0	Reserved			
1	audio_pdm_sel_i2s1	R/W	Enable PDM Mode 0 = Normal Operation 1 = I2S2 IO Operates in PDM Mode, using i2s_tdm_1 RX Module When this is set to 1, I2S2_BCLK IO OEN is controlled by Register i2s2_bclk_oen _ext	0x0
31:2	Reserved			

i2s_sys_int_en

Table 20.9: i2s_sys_int_en, Offset Address: 0x060

Bits	Name	Access	Description	Reset
0	i2s0_int_en	R/W	Enable I2S0 Interrupt	0x1
1	i2s1_int_en	R/W	Enable I2S1 Interrupt	0x1
2	i2s2_int_en	R/W	Enable I2S2 Interrupt	0x1
3	i2s3_int_en	R/W	Enable I2S3 Interrupt	0x1
7:4	Reserved			
8	i2s_subsys_int_en	R/W	Enable I2S_SUBSYS Interrupt	0x1
31:9	Reserved			

i2s_sys_ints

Table 20.10: i2s_sys_ints, Offset Address: 0x064

Bits	Name	Access	Description	Reset
0	i2s0_int	RO	I2S0 Interrupt Status When an I2S0 interrupt occurs, this can be further read to check I2S0 register I2S_INT value to determine the interrupt status.	
1	i2s1_int	RO	I2S1 Interrupt Status When an I2S1 interrupt occurs, this can be further read to check I2S1 register I2S_INT value to determine the interrupt status.	
2	i2s2_int	RO	I2S2 Interrupt Status When an I2S2 interrupt occurs, this can be further read to check I2S2 register I2S_INT value to determine the interrupt status.	
3	i2s3_int	RO	I2S3 Interrupt Status When an I2S3 interrupt occurs, this can be further read to check I2S3 register I2S_INT value to determine the interrupt status.	
31:4	Reserved			

20.1.6.2 I2S_TDM Module Register Description

BLK_MODE_SETTING

Table 20.11: BLK_MODE_SETTING, Offset Address: 0x000

Bits	Name	Access	Description	Reset
0	tx_mode	R/W	Select transmission mode 0 = Receive mode (RX), 1 = Transmit mode (TX)	0x0
1	master_mode	R/W	Select master or slave mode 0 = Slave mode 1 = Master mode, BCLK and LRCK are internally generated	0x0
2	rx_sample_edge	R/W	Select edge for sampling BCLK signal in receive mode 0 = Falling edge sampling 1 = Rising edge sampling	0x1
3	tx_sample_edge	R/W	Select edge for transmitting data SDO with BCLK signal 0 = Falling edge change 1 = Rising edge change	0x0
6:4	Reserved		Reserved	
7	dma_mode	R/W	Data transfer mode 0 = Software mode, 1 = DMA mode	0x1
8	Reserved	R/W	Reserved for internal debug, must be set to value 0x0	0x0
9	Reserved		Reserved	
10	slave_tx_fs_direct_in	R/W	Select source of LRCK for transmitting in slave mode 0 = LRCK signal is internally generated by TX module 1 = LRCK signal directly from master, register for debug only	0x0
11	Reserved		Reserved	
12	pcm_synth_mode	R/W	Master mode PCM frame sync signal uses frequency synthesis mode 0 = PCM frame sync signal period controlled by frame_length 1 = PCM frame sync signal period controlled by ck_coef_n/ck_coef_m	0x0
15:13	Reserved		Reserved	
31:16	reg_dummy	R/W	Dummy register	0xff00

FRAME_SETTING

Table 20.12: FRAME_SETTING, Offset Address: 0x004

Bits	Name	Access	Description	Reset
8:0	frame_length	R/W	Audio frame width (BCLK clock) 0~511 = 1~512 bits	0x1F
11:9	Reserved		Reserved	
12	fs_polarity	R/W	LRCK signal polarity 0 = Low level start (active) 1 = High level start (active) In I2S mode, 0 indicates left channel priority, LRCK outputs low level, 1 indicates left channel priority, LRCK outputs high level. In PCM mode, 1 indicates frame sync pulse is high.	0x0
13	fs_offset	R/W	Frame source data MSB delay 0 = In phase with LRCK signal 1 = Delayed by one beat compared to LRCK signal	0x1
14	fs_idef	R/W	Select LRCK signal mode 0 = As PCM/TDM frame sync 1 = As I2S left-right channel	0x1
15	Reserved		Reserved	
23:16	fs_active_length	R/W	LRCK signal active width 0~255 = 1~256 bits.	0x0F
31:24	Reserved		Reserved	

SLOT_SETTING1

Table 20.13: SLOT_SETTING1, Offset Address: 0x008

Bits	Name	Access	Description	Reset
3:0	slot_num	R/W	Number of channels per audio frame 0~15 = 1~16 channels Config 1 for stereo means two channels (left/right)	0x1
7:4	Reserved		Reserved	
13:8	slot_size	R/W	Channel width 0~63 = 1~64 bits	0x0F
15:14	Reserved		Reserved	
20:16	data_size	R/W	Valid data width within channel 0~31 = 1~32 bits In general (e.g., I2S and PCM modes), configure data width to be the same as channel width	0x0F
23:21	Reserved		Reserved	
28:24	fb_offset	R/W	Data MSB delay within channel 0 = Data and channel timing aligned, 1~31 = Delayed by 1~31 bits This value controls the timing of the first valid bit within each channel.	0x0
31:29	Reserved		Reserved	

SLOT_SETTING2

Table 20.14: SLOT_SETTING2, Offset Address: 0x00c

Bits	Name	Access	Description	Reset
15:0	slot_en	R/W	Enable active channel data Configuring slot_en[n] to 1 indicates the n-th channel data is active, configuring to 0 deactivates it. In TX mode, if a channel is set to inactive, then data sent on that channel will be all zeros. In RX mode, if a channel is set to inactive, received data for that channel will be replaced with zeros.	0x0003
31:16	Reserved		Reserved	

DATA_FORMAT

Table 20.15: DATA_FORMAT, Offset Address: 0x010

Bits	Name	Access	Description	Reset
0	data_format	R/W	Only allowed to be set to default 0x0	0x0
2:1	word_length	R/W	Select memory width of audio data 2'b00 = 8-bit, 2'b01 = 16-bit, 2'b10 = 32-bit, 2'b11 = reserved If the effective bits of data transmitted/received are less than this value, the data will be right aligned, and MSB padded with 0.	0x1
3	pad_slot_no	R/W	Pad channel notation into RX data 0 = No 1 = Fill the 4-bit channel sequence into the MSB of the audio data When receiving 24-bit data, bits 31~28 can be sequentially filled with the channel order and stored as 32-bit for software post-processing.	0x0
4	skip_rx_inactive_slot	R/W	In RX mode, ignore inactive channel data 0 = When receiving data for an inactive channel, it will be replaced with zeros and stored 1 = When receiving data for an inactive channel, it will be directly ignored and not stored in the RX FIFO.	0x0
5	skip_tx_inactive_slot	R/W	In TX mode, ignore inactive channel data 0 = When sending data for an inactive channel, data will be read from TX FIFO but replaced with zeros before sending 1 = When sending data for an inactive channel, it will not be read from TX FIFO, and zeros will be directly sent.	0x0
6	tx_source_left_align	R/W	TX mode memory data left aligned 0 = Data right aligned 1 = Data left aligned If memory width is greater than audio data width, e.g.,memory width is 32-bit and data width is 24-bit, when this value is set to 0, [23:0] will be sent, when set to 1, [31:8] will be sent.	0x0
31:7	Reserved		Reserved	

BLK_CFG

Table 20.16: BLK_CFG, Offset Address: 0x014

Bits	Name	Access	Description	Reset
0	force_complete	R/W	Force end mode 0 = After i2s_enable is set from 1 to 0, it ends after a complete frame of audio data is received or sent 1 = After i2s_enable is set from 1 to 0, forcibly ends data reception or transmission Recommended default value 0x0.	0x0
1	dma_req_force_stop	R/W	dma_req force stop mode 0 = After i2s_enable is set from 1 to 0, the sent dma_req remains high until DMA acknowledges with dma_ack 1 = After i2s_enable is set from 1 to 0, cancels the sent dma_req Recommended default value 0x1.	0x1
3:2	Reserved			
4	auto_disable_with_ch_en	R/W	I2S FIFO transfer auto stop 0 = Normal operation 1 = FIFO operation linked with system DMA When DMA channel in TX/RX mode ends transmission before i2s_enable is set to 0, TX/RX FIFO will stop accordingly. In RX mode, data continues to be received but not written to FIFO. In TX mode, zero data is sent without reading from FIFO.	0x0
5	Reserved			
6	rx_start_wait_dma_en	R/W	RX mode wait for DMA enable 0 = RX mode starts operating after i2s_enable is set from 0 to 1 1 = RX mode waits for DMA to be enabled after i2s_enable is set from 0 to 1 before starting operation	0x0
7	Reserved			
8	rx_blk_clk_force_en	R/W	RX block clock gate always on 0 = Automatic clock gate off when enabled 1 = Clock remains always on	0x0
9	rx_fifo_dma_clk_force_en	R/W	RX FIFO control module clock gate always on 0 = Automatic clock gate off when enabled 1 = Clock remains always on	0x1
15:10	Reserved			
16	tx_blk_clk_force_en	R/W	TX block clock gate always on 0 = Automatic clock gate off when enabled 1 = Clock remains always on	0x0
17	tx_fifo_dma_clk_force_en	R/W	TX FIFO control module clock gate always on 0 = Automatic clock gate off when enabled 1 = Clock remains always on	0x1
31:18	Reserved			

I2S_ENABLE

block enable

Table 20.17: I2S_ENABLE, Offset Address: 0x018

Bits	Name	Access	Description	Reset
0	i2s_enable	R/W	Enable TX/RX module 0 = Disable TX/RX 1 = Enable TX/RX module Before setting this value to 1, tx_fifo_reset/rx_fifo_reset and i2s_reset_tx/i2s_reset_rx must be configured first.	0x0
31:1	Reserved		Reserved	

I2S_RESET

sw reset

Table 20.18: I2S_RESET, Offset Address: 0x01c

Bits	Name	Access	Description	Reset
0	i2s_reset_rx	R/W	RX module soft reset 0 = No reset, 1 = Reset Write 1 then write 0 to reset the RX module. Due to cross- clock domain, users must wait a certain time to write 0 bafter writing 1 for the reset signal to take effect.	0x0
1	i2s_reset_tx	R/W	TX module soft reset 0 = No reset, 1 = Reset Write 1 then write 0 to reset the TX module. Due to cross- clock domain, must wait a certain time after writing 1 before writing 0, to let the reset signal take effect.	0x0
31:2	Reserved			

I2S_INT_EN

interrupt enable

Table 20.19: I2S_INT_EN, Offset Address: 0x020

Bits	Name	Access	Description	Reset
0	rx_fifo_avail_int_en	R/W	Enable RX FIFO data available interrupt	0x0
1	rx_fifo_overflow_int_en	R/W	Enable RX FIFO overflow interrupt	0x1
2	rx_fifo_underflow_int_en	R/W	Enable RX FIFO underflow interrupt	0x1
3	Reserved		Reserved	
4	tx_fifo_avail_int_en	R/W	Enable TX FIFO data available interrupt	0x0
5	tx_fifo_overflow_int_en	R/W	Enable TX FIFO overflow interrupt	0x1
6	tx_fifo_underflow_int_en	R/W	Enable TX FIFO underflow interrupt	0x1
7	Reserved		Reserved	
8	i2s_int_en	R/W	Enable I2S IP interrupt All I2S interrupt signals are merged into a 1-bit signal and reflected in the I2S subsystem register i2s_sys_ints	0x1
31:9	Reserved		Reserved	

I2S_INT

interrupt status

Table 20.20: I2S_INT, Offset Address: 0x024

Bits	Name	Access	Description	Reset
0	rx_fifo_avail_int	RO	RX FIFO data available interrupt status When RX FIFO depth is greater than rx_fifo_threshold, this value is 1. Write 1 to clear interrupt status. (rx_fifo_avail_int_en must be set to 1)	0x0
1	rx_fifo_overflow_int	RO	RX FIFO overflow interrupt status Write 1 to clear interrupt status. (rx_fifo_overflow_int_en must be set to 1)	
2	rx_fifo_underflow_int	RO	RX FIFO underflow interrupt status Write 1 to clear interrupt status. (rx_fifo_underflow_int_en must be set to 1)	
3	Reserved		Reserved	
4	tx_fifo_avail_int	RO	TX FIFO space available interrupt status When TX FIFO vacancy is greater than tx_fifo_threshold, this value is 1. Write 1 to clear interrupt status. (tx_fifo_avail_int_en must be set to 1)	
5	tx_fifo_overflow_int	RO	TX FIFO overflow interrupt status Write 1 to clear interrupt status. (tx_fifo_overflow_int_en must be set to 1)	
6	tx_fifo_underflow_int	RO	TX FIFO underflow interrupt status Write 1 to clear interrupt status. (tx_fifo_underflow_int_en must be set to 1)	
7	Reserved		Reserved	
8	rx_fifo_avail_int_raw	RO	RX FIFO data available raw interrupt status When RX FIFO depth is greater than rx_fifo_threshold, this value is 1. Write 1 to clear interrupt status. (rx_fifo_avail_int_en must be set to 1)	
9	rx_fifo_overflow_int_raw	RO	RX FIFO overflow raw interrupt status Write 1 to clear interrupt status.	
10	rx_fifo_underflow_int_raw	RO	RX FIFO underflow raw interrupt status Write 1 to clear interrupt status.	
11	Reserved		Reserved	
12	tx_fifo_avail_int_raw	RO	TX FIFO space available raw interrupt status When TX FIFO vacancy is greater than tx_fifo_threshold, this value is 1. Write 1 to clear interrupt status.	
13	tx_fifo_overflow_int_raw	RO	TX FIFO overflow raw interrupt status Write 1 to clear interrupt status.	
14	tx_fifo_underflow_int_raw	RO	TX FIFO underflow raw interrupt status Write 1 to clear interrupt status.	
31:15	Reserved		Reserved	

FIFO_THRESHOLD

Table 20.21: FIFO_THRESHOLD, Offset Address: 0x028

Bits	Name	Access	Description	Reset
4:0	rx_fifo_threshold	R/W	RX FIFO threshold In DMA transfer mode, when the RX FIFO's fill level is greater than or equal to this value, a read request signal is sent to DMA.	0x7
15:5	Reserved			
20:16	tx_fifo_threshold	R/W	TX FIFO threshold In DMA transfer mode, when the TX FIFO's vacancy level is greater than or equal to this value, a write request signal is sent to DMA.	0x7
23:21	Reserved			
28:24	tx_fifo_high_threshold	R/W	TX FIFO high threshold After enabling the TX module, it sends a write request to DMA until the FIFO's fill level is greater than or equal to this value, at which point the TX module starts transmitting data externally.	0x1F
31:29	Reserved			

I2S_LRCK_MASTER

block enable

Table 20.22: I2S_LRCK_MASTER, Offset Address: 0x02c

Bits	Name	Access	Description	Reset
0	i2s_lrck_master_enable	R/W	Enable i2s_tdm module as BCLK/LRCK master mode output generator 0 = When operating in master mode (master_mode set to 1), BCLK and LRCK signals are output externally only after enabling TX/RX (i2s_enable set to 1) 1 = In master mode, and with aud_en set to 1, setting this value to 1 starts outputting BCLK and LRCK immediately. At this time, the i2s_tdm module acts solely as a master mode BCLK/LRCK signal generator.	0x0
31:1	Reserved			

FIFO_RESET

Table 20.23: FIFO_RESET, Offset Address: 0x030

Bits	Name	Access	Description	Reset
0	rx_fifo_reset	R/W	RX channel FIFO soft reset Write 1 then 0 to soft reset the RX FIFO	0x0
15:1	Reserved		Reserved	
16	tx_fifo_reset	R/W	TX channel FIFO soft reset Write 1 then 0 to soft reset the TX FIFO	0x0
31:17	Reserved		Reserved	

RX_STATUS

Table 20.24: RX_STATUS, Offset Address: 0x040

Bits	Name	Access	Description	Reset
8:0	rx_frame_size_cnt	RO	RX frame width counter value	
9	rx_i2s_disable_req	RO	Internal signal	
10	i2s_rx_start_wait	RO	Internal signal	
16:11	rx_data_size_cnt	RO	RX module effective data width counter value	
22:17	rx_slot_size_cnt	RO	RX slot width counter value	
23	i2s_reset_rx_sclk	RO	RX module clock domain's i2s_reset_rx signal state	
28:24	rx_slot_num_cnt	RO	RX slot number counter value	
29	receive_start_sclk	RO	RX module clock domain's i2s_enable signal state	
30	rx_blk_active	RO	RX module is active	
31	rx_dma_req	RO	RX module issues dma_req	

TX_STATUS

Table 20.25: TX_STATUS, Offset Address: 0x048

Bits	Name	Access	Description	Reset
8:0	tx_frame_size_cnt	RO	Value of the internal frame width counter in the TX module	
9	tx_i2s_disable_req	RO	Internal signal	
10	i2s_tx_start_wait	RO	Internal signal	
16:11	tx_data_size_cnt	RO	Value of the internal valid data width counter in the TX module	
22:17	tx_slot_size_cnt	RO	Value of the internal channel width counter in the TX module	
23	i2s_reset_tx_sclk	RO	Status of the i2s_reset_tx signal in the TX module clock domain	
28:24	tx_slot_num_cnt	RO	Value of the internal channel count counter in the TX module	
29	transmit_start_sclk	RO	Status of the i2s_enable signal in the TX module clock domain	
30	tx_blk_active	RO	TX module is active	
31	tx_dma_req	RO	TX module issues dma_req	

I2S_CLK_CTRL0

Table 20.26: I2S_CLK_CTRL0, Offset Address: 0x060

Bits	Name	Access	Description	Reset
0	aud_clk_sel	R/W	Select audio clock source (aud_clk) 0 = From internal audio PLL 1 = From external mclk_in	0x0
1	Reserved			
2	mclk_out_inv	R/W	Invert mclk_out signal for external interface in master mode 0 = mclk_out not inverted 1 = mclk_out inverted	0x0
3	bclk_out_inv	R/W	Invert bclk_out signal for external interface in master mode 0 = bclk_out not inverted 1 = bclk_out inverted	0x0
4	bclk_in_inv	R/W	Invert bclk_in signal in slave mode 0 = bclk_in not inverted 1 = bclk_in inverted	0x0
5	Reserved			
6	bclk_out_clk_force_en	R/W	Force bclk_out clock always on in master mode 1 = Always on 0 = Output only when TX/RX mode is enabled and data transmission begins	0x1
7	mclk_out_en	R/W	Enable mclk_out IO output 0 = disable 1 = enable	0x0
8	aud_en	R/W	Enable clock generator and master mode clock output 0 = Clock generator off 1 = Enable clock generator and output clock signals Configure to 0 in slave mode. Before setting to 1 in master mode, configure the dividers mclk_div/bclk_div according to the data sampling rate.	0x0
31:9	Reserved			

I2S_CLK_CTRL1

Table 20.27: I2S_CLK_CTRL1, Offset Address: 0x064

Bits	Name	Access	Description	Reset
15:0	mclk_div	R/W	mclk clock division from source clock 1 = divide by 1, 2 = divide by 2, 3 = divide by 3, …	0x1
31:16	bclk_div	R/W	bclk clock division from mclk clock (only valid in master mode) 1 = divide by 1, 2 = divide by 2, 3 = divide by 3, …	0x2

I2S_PCM_SYNTH

Table 20.28: I2S_PCM_SYNTH, Offset Address: 0x068

Bits	Name	Access	Description	Reset
11:0	ck_coef_n	R/W	PCM frame sync signal period division coefficient N When pcm_synth_mode is set to 1, frame sync signal period = BCLK * (N/M), N value must be less than M value.	0x1
15:12	Reserved		Reserved	
31:16	ck_coef_m	R/W	PCM frame sync signal period division coefficient M When pcm_synth_mode is set to 1, frame sync signal period = BCLK * (N/M), N value must be less than M value.	0x40

RX_RD_PORT

Table 20.29: RX_RD_PORT, Offset Address: 0x080

Bits	Name	Access	Description	Reset
31:0	rx_rd_port	RO	RX FIFO read address DMA should configure this address as the read address for receiving data.	

TX_WR_PORT

Table 20.30: TX_WR_PORT, Offset Address: 0x0c0

Bits	Name	Access	Description	Reset
31:0	tx_wr_port	WO	TX FIFO write address DMA should configure this address as the write address for sending data.	0x0

20.2 Audio Codec

20.2.1 Overview

The chip integrates high-performance Audio Codec, including stereo playback DAC (90dB DR A-Weighted), which supports two single-ended lineout outputs; stereo recording ADC (90dB DR A-Weighted), which supports stereo single-ended input.

20.2.2 Features

- 90dB DR A-Weighted Stereo DAC
- Stereo single-ended Lineout output
- DAC digital volume control range: -24dB ~ 0dB
- DAC supports 8kHz ~ 48kHz sampling rate
- 90dB DR A-Weighted Stereo ADC
- ADC PGA gain range: 0db ~ 48dB
- ADC supports 8kHz ~ 48kHz sampling rate
- ADC supports Mic in stereo single-ended input or Line in stereo single-ended input

20.2.3 Audio Codec Registers

20.2.3.1 Audio DAC/ADC Register Overview

Table 20.31: Audio DAC/ADC Registers Overview

Name	Address Offset	Description
txdac_ctrl0	0x000	
txdac_ctrl1	0x004	
txdac_status	0x008	
txdac_afe0	0x00c	
txdac_afe1	0x010	
txdac_ana0	0x020	
txdac_ana1	0x024	
rxadc_ctrl0	0x100	
rxadcc_ctrl1	0x104	
rxadc_status	0x108	
rxadc_ana0	0x110	
rxadc_ana1	0x114	
rxadc_ana2	0x118	
rxadc_ana3	0x11c	
rxadc_ana4	0x120	

20.2.3.2 Audio ADC Register Description

rxadc_ctrl0

Table 20.32: rxadc_ctrl0, Offset Address: 0x100

Bits	Name	Access	Description	Reset
0	reg_rxadc_en	R/W		0x0
1	reg_i2s_tx_en	R/W		0x0
31:2	Reserved			

rxadcc_ctrl1

Table 20.33: rxadcc_ctrl1, Offset Address: 0x104

Bits	Name	Access	Description	Reset
1:0	reg_rxadc_cic_opt	R/W	CIC decimation filter option 0: downsample ratio 64 1: downsample ratio 128 2: downsample ratio 256 3: downsample ratio 512	0x0
2	reg_rxadc_chn_swap	R/W	L/R input data channel swap	0x0
3	reg_rxadc_single	R/W	Only single channel supported, used when ANALOG in differential input mode	0x0
6:4	reg_rxadc_dcb_opt	R/W	DC blocking filter option 3'b000: bypass 3'b001: 1-2^(-8) 3'b010: 1-2^(-9) 3'b011: 1-2^(-10) 3'b100: 1-2^(-11) 3'b101: 1-2^(-12) other: bypass	0x5
7	Reserved			
8	reg_rxadc_igr_init	R/W	i2s keep silence when filter with initial value	0x0
9	reg_rxadc_clk_force_en	R/W	force clock enable	0x0
10	reg_rxadc_fsm_src_sel	R/W	FSM trigger source 0: adc0_vld 1: adc1_vld	0x0
31:11	Reserved			

rxadc_status

Table 20.34: rxadc_status, Offset Address: 0x108

Bits	Name	Access	Description	Reset
0	reg_rxadc_cic0_init_done	RO	cic_d_0 init done	
1	reg_rxadc_fir1_0_init_done	RO	fir1_0 init done	
2	reg_rxadc_fir2_0_init_done	RO	fir2_0 init done	
3	Reserved			
4	reg_rxadc_cic1_init_done	RO	cic_d_1 init done	
5	reg_rxadc_fir1_1_init_done	RO	fir1_1 init done	
6	reg_rxadc_fir2_1_init_done	RO	fir2_1 init done	
7	Reserved			
10:8	reg_rxadc_fsm	RO	ADC FSM state	
31:11	Reserved			

rxadc_ana0

Table 20.35: rxadc_ana0, Offset Address: 0x110

Bits	Name	Access	Description	Reset
12:0	reg_gstepl_rxpga	R/W	PGA feedback resistance selection 2dB/step default:20Kohm	0x0001
13	reg_g6dbl_rxpga	R/W	PGA input resistance selection 0:20Kohm 1:10Kohm	0x0
15:14	reg_gainl_rxadc	R/W	PGA input resistance selection 0:20Kohm(0dB) 1:10Kohm(6dB) 2: 5Kohm(12dB) 3: 2.5Kohm(18dB)	0x0
28:16	reg_gstepr_rxpga	R/W	PGA feedback resistance selection 2dB/step default:20Kohm	0x0001
29	reg_g6dbr_rxpga	R/W	PGA input resistance selection 0:20Kohm 1:10Kohm	0x0
31:30	reg_gainr_rxadc	R/W	PGA input resistance selection 0:20Kohm(0dB) 1:10Kohm(6dB) 2: 5Kohm(12dB) 3: 2.5Kohm(18dB)	0x0

rxadc_ana1

Table 20.36: rxadc_ana1, Offset Address: 0x114

Bits	Name	Access	Description	Reset
15:0	reg_gainl_status	RO	[12:0]: gstep1 [13]: g6dbl [15:14]: gainl	
31:16	reg_gainr_status	RO	[28:16]: gstepr [29]: g6dbr [31:30]: gainr	

rxadc_ana2

Table 20.37: rxadc_ana2, Offset Address: 0x118

Bits	Name	Access	Description	Reset
0	reg_mutel_rxpga	R/W	Enable pin of mute left channel, active high	0x0
1	reg_muter_rxpga	R/W	Enable pin of mute right channel, active high	0x0
15:2	Reserved			
16	reg_diff_en_rxpga	R/W	Enable pin of differential mode(Left channel only, VIN/ VINB=PAD_VINL/PAD_VINR	0x0
17	reg_tristate_rxpga	R/W	?	0x0
31:18	Reserved			

rxadc_ana3

Table 20.38: rxadc_ana3, Offset Address: 0x11c

Bits	Name	Access	Description	Reset
0	reg_addi_rxadc	R/W	ADC opamp current +50%, active High	0x0
1	reg_cksel_rxadc	R/W	PGA enable control input, active High	0x0
2	reg_en_asar_i_rxadc	R/W	Enable pin of L channel quantizer	0x1
3	reg_en_asar_q_rxadc	R/W	Enable pin of Q channel quantizer	0x1
5:4	reg_dem_type_rxadc	R/W	DEM TYPE 0:rotation 1: min cell switching 2/3: NA	0x1
7:6	Reserved			
11:8	reg_ctune_rxadc	R/W	RXADC integrator CFB selection 12*135fF + (8/4/2/1)*135fF	0xc
12	reg_en_dither_rxadc	R/W	Enable pin of dithering	0x1
13	reg_rstsdm_rxadc	R/W	Enable pin of resetting integrator	0x0
14	reg_en_vcmt_rxadc	R/W	?	0x0
15	Reserved			
17:16	reg_vldo0p9_rxadc	R/W	0.9V LDO output selection 00:0.85V 01:0.9V 10:0.95V 11:1.0V	0x1
19:18	reg_vldo12_rxadc	R/W	1.2V LDO output selection 00:1.1V 01:1.15V 10:1.2V 11:1.25V	0x1
21:20	reg_rnlvl_rxadc	R/W	Dither option	0x0
31:22	Reserved			

rxadc_ana4

Table 20.39: rxadc_ana4, Offset Address: 0x120

Bits	Name	Access	Description	Reset
0	reg_da_en_rxpga_status	RO	DA_EN_RXPGA status	
1	reg_da_end2us_rxpga_status	RO	DA_END2US_RXPGA status	
2	reg_da_en_rxadc_status	RO	DA_EN_RXADC status	
3	reg_da_en_audbias_status	RO	DA_EN_AUDBIAS status	
15:4	Reserved			
18:16	reg_ad_dol_rxadc	RO	Left channel 3-bits output	
19	Reserved			
22:20	reg_ad_dor_rxadc	RO	Right channel 3-bits output	
31:23	Reserved			

20.2.3.3 Audio DAC Register Description

txdac_ctrl0

Table 20.40: txdac_ctrl0, Offset Address: 0x000

Bits	Name	Access	Description	Reset
0	reg_txdac_en	R/W	audio dac enable	0x0
1	reg_i2s_rx_en	R/W	audio dac i2s output enable	0x0
31:2	Reserved			

txdac_ctrl1

Table 20.41: txdac_ctrl1, Offset Address: 0x004

Bits	Name	Access	Description	Reset
1:0	reg_txdac_cic_opt	R/W	CIC decimation filter option 0: upsample ratio 64 1: upsample ratio 128 2: upsample ratio 256 3: upsample ratio 512	0x0
3:2	Reserved			
5:4	reg_txdac_dem_type	R/W	DEM TYPE 0:rotation 1: min cell switching 2/3: thermal code	0x1
7:6	Reserved			
8	reg_txdac_dsm_opt	R/W	DSM order option 0: order2 1: order1	0x0
9	reg_txdac_zcd_en	R/W	enable zero corssing function move from reg_spare0	0x0
10	reg_txdac_fsm_src_sel	R/W	FSM trigger source 0: i2s1_vld 1: i2s0_vld	0x1
11	Reserved			
14:12	reg_txdac_dither_opt	R/W	Dither option 0: disable others: weight = LSB/(2^(n-1))	0x0
31:15	Reserved			

txdac_status

Table 20.42: txdac_status, Offset Address: 0x008

Bits	Name	Access	Description	Reset
2:0	reg_txdac_fsm	RO	DAC main fsm state	
3	Reserved			
6:4	reg_txdac_afe_fsm	RO	DAC AFE fsm state	
7	Reserved			
16:8	reg_txdac_gain0	RO	DAC L-channel gain	
19:17	Reserved			
28:20	reg_txdac_gain1	RO	DAC R-channel gain	
31:29	Reserved			

txdac_afe0

Table 20.43: txdac_afe0, Offset Address: 0x00c

Bits	Name	Access	Description	Reset
5:0	reg_txdac_init_dly_cnt	R/W	DAC AFE initialize delay, min>2us	0x28
7:6	Reserved			
15:8	reg_txdac_value_tick	R/W	DAC AFE tick value for initialize/stop value ramp	0x0
27:16	reg_txdac_gain_tick	R/W	DAC AFE tick value for gain ramp	0x800
31:28	Reserved			

txdac_afe1

Table 20.44: txdac_afe1, Offset Address: 0x010

Bits	Name	Access	Description	Reset
8:0	reg_txdac_gain_ub_0	R/W	channel 0, L-channel, DAC gain upper bound 0x00: gain=0 0xff: gain=1-2^(-8), default value to prevent DSM overflow 0x100: gain=1	0xff
15:9	Reserved			
24:16	reg_txdac_gain_ub_1	R/W	channel 1, R-channel, DAC gain upper bound 0x00: gain=0 0xff: gain=1-2^(-8), default value to prevent DSM overflow 0x100: gain=1	0xff
27:25	Reserved			
28	reg_txdac_ramp_bp	R/W	bypass initial ramp procedure	0x0
31:29	Reserved			

txdac_ana0

Table 20.45: txdac_ana0, Offset Address: 0x020

Bits	Name	Access	Description	Reset
0	reg_addi_txdac	R/W	na	0x0
3:1	Reserved			
5:4	reg_tsel_txdac	R/W	2'b00: NA 2'b01: VCM 2'b10: VDD15A 2'b11: undefined	0x0
7:6	Reserved			
9:8	reg_vsel_txdac	R/W	1.5V LDO output selection 2'b00:1.35V 2'b01:1.4V 2'b10:1.45V 2'b11:1.5V	0x3
31:10	Reserved			

txdac_ana1

Table 20.46: txdac_ana1, Offset Address: 0x024

Bits	Name	Access	Description	Reset
0	reg_da_en_txdac_ow_val	R/W	DA_EN_TXDAC overwrite value	0x0
1	reg_da_end2us_txdac_ow_val	R/W	DA_END2US_TXDAC overwrite value	0x0
15:2	Reserved			
16	reg_da_en_txdac_ow_en	R/W	DA_EN_TXDAC overwrite enable	0x0
17	reg_da_end2us_txdac_ow_en	R/W	DA_END2US_TXDAC overwrite enable	0x0
31:18	Reserved			

CHAPTER

TWENTYONE

PERIPHERALS

21.1 I2C

21.1.1 Overview

This chip is equipped with 6 I2C controllers (5 in Active Domain, 1 in No-die Domain), which can be individually configured as Master/Slave. For IO configuration, please refer to Function pin mux for configuration.

21.1.2 Function Description

The I2C controller has the following features:

- Supports standard address (7bit) and extended address (10bit).
- The transfer rate supports standard mode (100kbit/s) and fast mode (400kbit/s).
- Support General Call and Start Byte functions.
- CBUS devices are not supported.
- Support DMA operation.
- Support 64 x 8bit TX FIFO and 64 x 8bit RX FIFO.

21.1.3 Functional Block Diagram

I2C functional block diagram is the functional block diagram of the I2C module. IIC_CLK is the module clock, and the chip supports 25MHz or 100MHz. The CPU selects various I2C modes and timings through the APB bus configuration register, writes TXFIFO, reads RXFIFO, and triggers FSM to send and receive SDA/SCL related IO signals. System DMA can also be used with I2C. DMA_IF, and write TXFIFO and read RXFIFO through the APB bus to send and receive I2C signals.

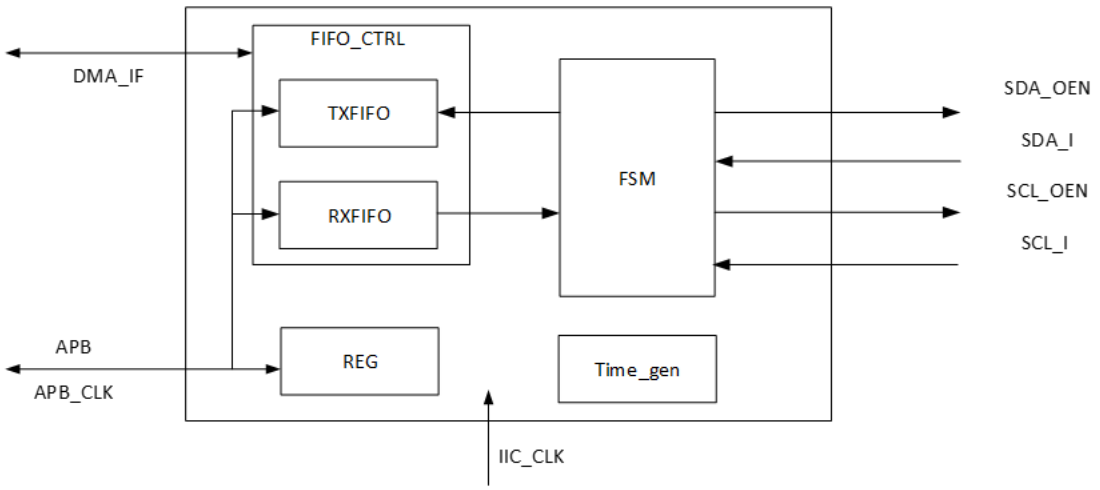


Diagram 21.1: I2C functional block diagram

21.1.4 I2C Protocol Timing

The chip I2C supports the general standard I2C protocol timing as shown in *I2C Protocol Timing*.

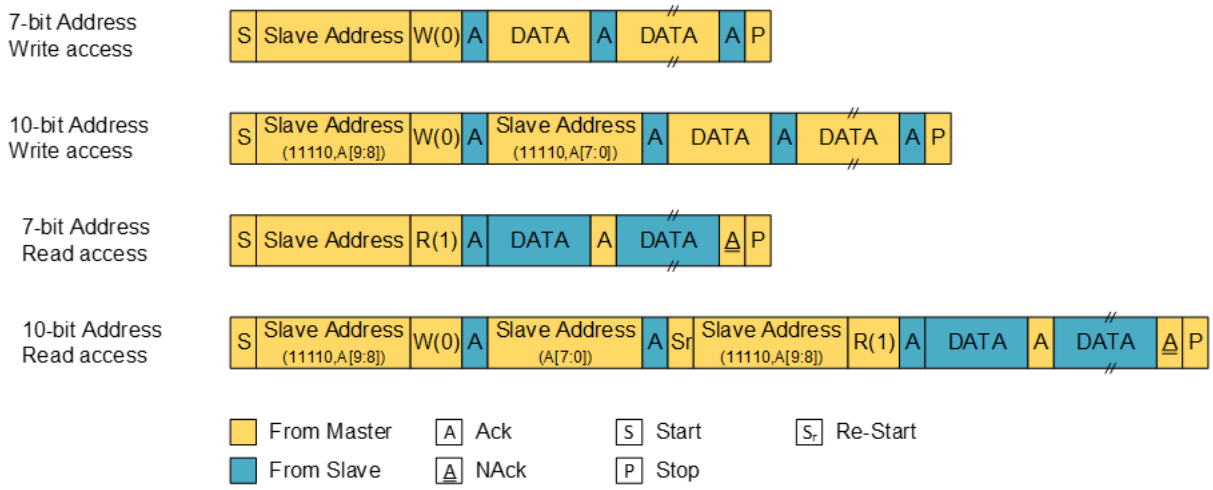


Diagram 21.2: I2C Protocol Timing

21.1.5 Way of Working

21.1.5.1 Configure I2C Clock and Timing Parameters

1. CLK_DIV CRG register chapter, configure clk_byp_0_31 to select IIC_CLK as the 25MHz default clock source or 100MHz z clock source.
2. The module should be in disabled state when configuring related timing configuration. IC_ENABLE needs to be set to 0, and query IC_ENABLE_STATUS[0] to confirm it is 0.
3. Refer to the table *I2C clock selection and related register configuration relationship table*, select according to the I2C clock, and configure the I2C timing count register.

Table 21.1: I2C clock selection and related register configuration relationship table

Register	25M IIC_CLK	100M IIC_CLK	Description
IC_SS_SCL_HCNT	115	460	SCL high level time counting in standard speed mode
IC_SS_SCL_LCNT	135	540	SCL low level time counting in standard speed mode
IC_FS_SCL_HCNT	21	90	SCL high level time counting in fast speed mode
IC_FS_SCL_LCNT	42	160	SCL low level time counting in fast speed mode
IC_SDA_HOLD	1	1	SDA hold time count, relative to SCL negative edge
IC_SDA_SETUP	6	25	SDA time count, relative to the positive edge of SCL
IC_FS_SPKLEN	2	5	I2C glitch suppression time count

21.1.5.2 Data transfer in non-DMA mode

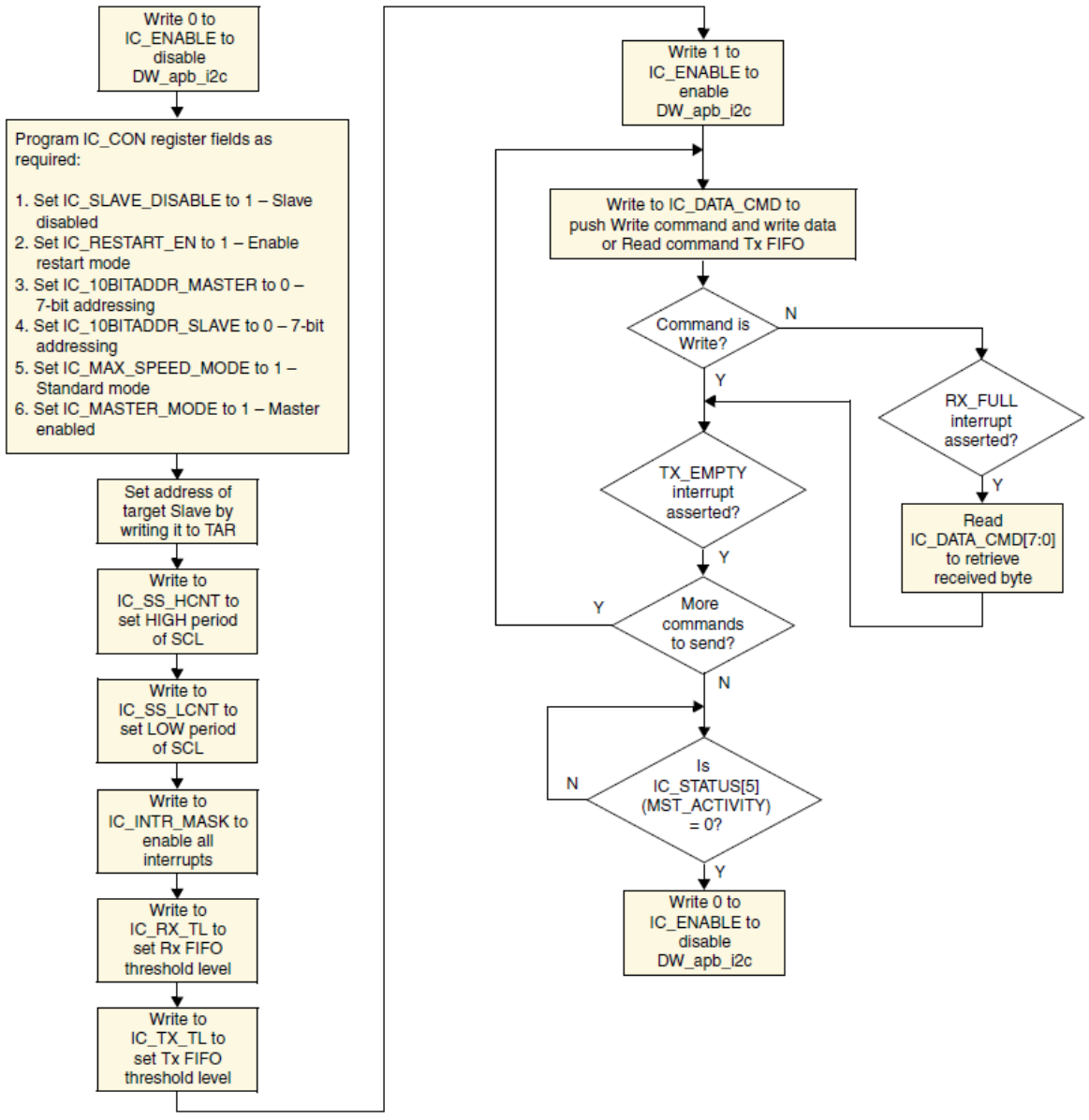


Diagram 21.3: I2C Data transfer software process in non-DMA mode

21.1.5.3 Data transfer in DMA mode

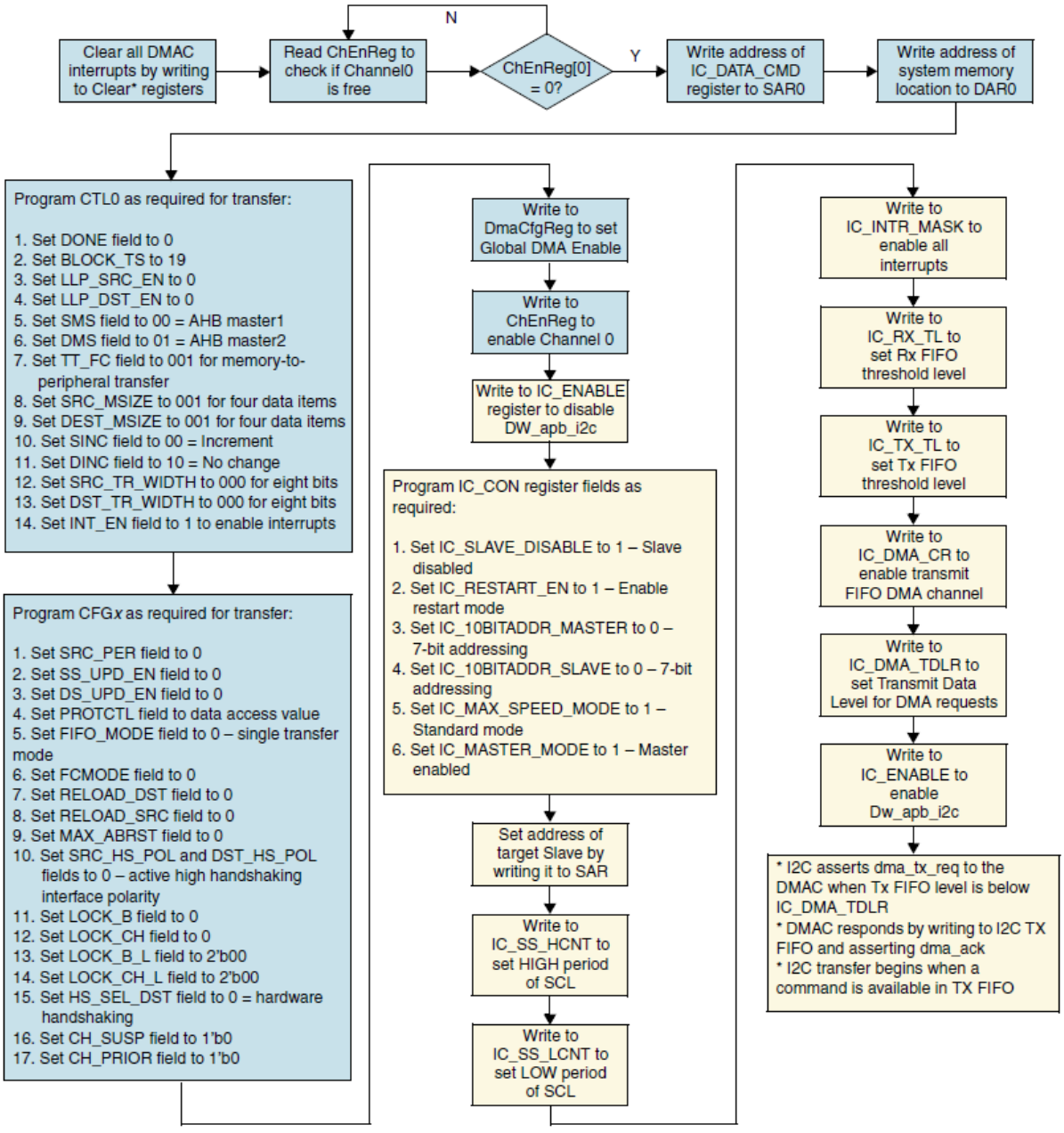


Diagram 21.4: I2C Data transmission software process in DMA mode

21.1.6 I2C Register Overview

Includes 6 I2C, 5 Active Domain, 1 No-die Domain. Their base addresses are as follows. Each I2C consists of a set of control registers, each set identically defined.

Table 21.2: Base addresses for 6 sets I2C module

GPIO Module	Base Address
I2C0	0x04000000
I2C1	0x04010000
I2C2	0x04020000
I2C3	0x04030000
I2C4	0x04040000
RTCSYS_I2C	0x0502B000

Table 21.3: I2C Registers Overview

Name	Address Offset	Description
IC_CON	0x000	I2C Control
IC_TAR	0x004	I2C Target Address
IC_SAR	0x008	I2C Slave Address
IC_DATA_CMD	0x010	I2C Rx/Tx Data Buffer and Command
IC_SS_SCL_HCNT	0x014	Standard speed I2C Clock SCL High Count
IC_SS_SCL_LCNT	0x018	Standard speed I2C Clock SCL Low Count
IC_FS_SCL_HCNT	0x01c	Fast speed I2C Clock SCL High Count
IC_FS_SCL_LCNT	0x020	Fast speed I2C Clock SCL Low Count
IC_INTR_STAT	0x02c	I2C Interrupt Status
IC_INTR_MASK	0x030	I2C Interrupt Mask
IC_RAW_INTR_STAT	0x034	I2C Raw Interrupt Status
IC_RX_TL	0x038	I2C Receive FIFO Threshold
IC_TX_TL	0x03c	I2C Transmit FIFO Threshold
IC_CLR_INTR	0x040	Clear Combined and Individual Interrupts
IC_CLR_RX_UNDER	0x044	Clear RX_UNDER Interrupt
IC_CLR_RX_OVER	0x048	Clear RX_OVER Interrupt
IC_CLR_TX_OVER	0x04c	Clear TX_OVER Interrupt
IC_CLR_RD_REQ	0x050	Clear RD_REQ Interrupt
IC_CLR_TX_ABRT	0x054	Clear TX_ABRT Interrupt
IC_CLR_RX_DONE	0x058	Clear RX_DONE Interrupt
IC_CLR_ACTIVITY	0x05c	Clear ACTIVITY Interrupt
IC_CLR_STOP_DET	0x060	Clear STOP_DET Interrupt
IC_CLR_START_DET	0x064	Clear START_DET Interrupt
IC_CLR_GEN_CALL	0x068	Clear GEN_CALL Interrupt
IC_ENABLE	0x06c	I2C Enable
IC_STATUS	0x070	I2C Status register
IC_TXFLR	0x074	Transmit FIFO Level Register
IC_RXFLR	0x078	Receive FIFO Level Register
IC_SDA_HOLD	0x07c	SDA hold time length register
IC_TX_ABRT_SOURCE	0x080	I2C Transmit Abort Status Register
IC_SLV_DATA_NACK_ONLY	0x084	Generate SLV_DATA_NACK Register
IC_DMA_CR	0x088	DMA Control Register for transmit and receive handshaking interface
IC_DMA_TDLR	0x08c	DMA Transmit Data Level
IC_DMA_RDLR	0x090	DMA Receive Data Level
IC_SDA_SETUP	0x094	I2C SDA Setup Register
IC_ACK_GENERAL_CALL	0x098	I2C ACK General Call Register
IC_ENABLE_STATUS	0x09c	I2C Enable Status Register

continues on next page

Table 21.3 – continued from previous page

Name	Address Offset	Description
IC_FS_SPKLEN	0x0a0	ISS and FS spike suppression limit
IC_HS_SPKLEN	0x0a4	HS spike suppression limit

21.1.7 I2C Register Description

21.1.7.1 IC_CON

Table 21.4: RTC_IC_CON, Offset Address: 0x000

Bits	Name	Access	Description	Reset
0	MASTER_MODE	R/W	enable master mode	0x1
2:1	SPEED	R/W	1: standard mode (~100 kbit/s) 2: fast mode (~400 kbit/s)	0x3
3	IC_10BITADDR_SLAVE	R/W	enable 10bit slave address mode	0x1
4	IC_10BITADDR_MASTER	R/W	enable 10bit master address mode	0x1
5	IC_RESTART_EN	R/W	enable I2C master to be able generate restart	0x1
6	IC_SLAVE_DISABLE	R/W	0: slave is enabled 1: slave is disabled	0x1
31:7	Reserved			

21.1.7.2 IC_TAR

Table 21.5: IC_TAR, Offset Address: 0x004

Bits	Name	Access	Description	Reset
9:0	IC_TAR	R/W	I2C Target Address Register	0x55
10	GC_OR_START	R/W	If bit SPECIAL is set to 1, then this bit indicates whether a General Call or START byte command 0: general call 1: start byte	0x0
11	SPECIAL	R/W	Used to issue General Call or START BYTE	0x0
31:12	Reserved			

21.1.7.3 IC_SAR

Table 21.6: IC_SAR, Offset Address: 0x008

Bits	Name	Access	Description	Reset
9:0	IC_SAR	R/W	I2C Slave Address Register	0x55
31:10	Reserved			

21.1.7.4 IC_DATA_CMD

Table 21.7: IC_DATA_CMD, Offset Address: 0x010

Bits	Name	Access	Description	Reset
7:0	DAT	R/W	transmitted or received data port	0x0
8	CMD	R/W	0: Write 1: Read	0x0
9	STOP	R/W	issue stop	0x0
10	RESTART	R/W	issue restart	0x0
31:11	Reserved			

21.1.7.5 IC_SS_SCL_HCNT

Table 21.8: IC_SS_SCL_HCNT, Offset Address: 0x014

Bits	Name	Access	Description	Reset
15:0	IC_SS_SCL_HCNT	R/W	Standard Speed I2C Clock SCL High Count Register	0x0190
31:16	Reserved			

21.1.7.6 IC_SS_SCL_LCNT

Table 21.9: IC_SS_SCL_LCNT, Offset Address: 0x018

Bits	Name	Access	Description	Reset
15:0	IC_SS_SCL_LCNT	R/W	Standard Speed I2C Clock SCL Low Count Register	0x01d6
31:16	Reserved			

21.1.7.7 IC_FS_SCL_HCNT

Table 21.10: IC_FS_SCL_HCNT, Offset Address: 0x01c

Bits	Name	Access	Description	Reset
15:0	IC_FS_SCL_HCNT	R/W	Fast Speed I2C Clock SCL High Count Register	0x003C
31:16	Reserved			

21.1.7.8 IC_FS_SCL_LCNT

Table 21.11: IC_FS_SCL_LCNT, Offset Address: 0x020

Bits	Name	Access	Description	Reset
15:0	IC_FS_SCL_LCNT	R/W	Fast Speed I2C Clock SCL Low Count Register	0x0082
31:16	Reserved			

21.1.7.9 IC_INTR_STAT

Table 21.12: IC_INTR_STAT, Offset Address: 0x02c

Bits	Name	Access	Description	Reset
0	R_RX_UNDER	RO	corresponding masked interrupt staus, please reference I2C Raw Interrupt Status	
1	R_RX_OVER	RO	corresponding masked interrupt staus, please reference I2C Raw Interrupt Status	
2	R_RX_FULL	RO	corresponding masked interrupt staus, please reference I2C Raw Interrupt Status	
3	R_TX_OVER	RO	corresponding masked interrupt staus, please reference I2C Raw Interrupt Status	
4	R_TX_EMPTY	RO	corresponding masked interrupt staus, please reference I2C Raw Interrupt Status	
5	R_RD_REQ	RO	corresponding masked interrupt staus, please reference I2C Raw Interrupt Status	
6	R_TX_ABRT	RO	corresponding masked interrupt staus, please reference I2C Raw Interrupt Status	
7	R_RX_DONE	RO	corresponding masked interrupt staus, please reference I2C Raw Interrupt Status	
8	R_ACTIVITY	RO	corresponding masked interrupt staus, please reference I2C Raw Interrupt Status	
9	R_STOP_DET	RO	corresponding masked interrupt staus, please reference I2C Raw Interrupt Status	
10	R_START_DET	RO	corresponding masked interrupt staus, please reference I2C Raw Interrupt Status	
11	R_GEN_CALL	RO	corresponding masked interrupt staus, please reference I2C Raw Interrupt Status	
31:12	Reserved			

21.1.7.10 IC_INTR_MASK

Table 21.13: IC_INTR_MASK, Offset Address: 0x030

Bits	Name	Access	Description	Reset
0	M_RX_UNDER	R/W	corresponding interrupt staus mask, please reference I2C Raw Interrupt Status	0x1
1	M_RX_OVER	R/W	corresponding interrupt staus mask, please reference I2C Raw Interrupt Status	0x1
2	M_RX_FULL	R/W	corresponding interrupt staus mask, please reference I2C Raw Interrupt Status	0x1
3	M_TX_OVER	R/W	corresponding interrupt staus mask, please reference I2C Raw Interrupt Status	0x1
4	M_TX_EMPTY	R/W	corresponding interrupt staus mask, please reference I2C Raw Interrupt Status	0x1
5	M_RD_REQ	R/W	corresponding interrupt staus mask, please reference I2C Raw Interrupt Status	0x1
6	M_TX_ABRT	R/W	corresponding interrupt staus mask, please reference I2C Raw Interrupt Status	0x1
7	M_RX_DONE	R/W	corresponding interrupt staus mask, please reference I2C Raw Interrupt Status	0x1
8	M_ACTIVITY	R/W	corresponding interrupt staus mask, please reference I2C Raw Interrupt Status	0x0
9	M_STOP_DET	R/W	corresponding interrupt staus mask, please reference I2C Raw Interrupt Status	0x0
10	M_START_DET	R/W	corresponding interrupt staus mask, please reference I2C Raw Interrupt Status	0x0
11	M_GEN_CALL	R/W	corresponding interrupt staus mask, please reference I2C Raw Interrupt Status	0x1
31:12	Reserved			

21.1.7.11 IC_RAW_INTR_STAT

Table 21.14: IC_RAW_INTR_STAT, Offset Address: 0x034

Bits	Name	Access	Description	Reset
0	IST_RX_UNDER	RO	when receive buffer is empty by reading from the IC_DATA_CMD register	
1	IST_RX_OVER	RO	receive buffer is oveflow (64Bytes)	
2	IST_RX_FULL	RO	receive buffer reaches or goes above the RX_TL threshold	
3	IST_TX_OVER	RO	transmit buffer is oveflow (64Bytes)	
4	IST_TX_EMPTY	RO	transmit buffer is at or below the TX_TL threshold	
5	IST_RD_REQ	RO	In slave mode, I2C hold SCL and wait for the response from processor	
6	IST_TX_ABRT	RO	In master or slave mode, when transmitter is unable to complete the action	
7	IST_RX_DONE	RO	In slave-transmitter mode, a NACK is received	
8	IST_ACTIVITY	RO	I2C activity is detected	
9	IST_STOP_DET	RO	STOP occurred	
10	IST_START_DET	RO	START or RESTART occurred	
11	IST_GEN_CALL	RO	General Call address is received	
31:12	Reserved			

21.1.7.12 IC_RX_TL

Table 21.15: IC_RX_TL, Offset Address: 0x038

Bits	Name	Access	Description	Reset
7:0	RX_TL	R/W	Receive FIFO Threshold Level	0x0
31:8	Reserved			

21.1.7.13 IC_TX_TL

Table 21.16: IC_TX_TL, Offset Address: 0x03c

Bits	Name	Access	Description	Reset
7:0	TX_TL	R/W	Transmit FIFO Threshold Level	0x0
31:8	Reserved			

21.1.7.14 IC_CLR_INTR

Table 21.17: IC_CLR_INTR, Offset Address: 0x040

Bits	Name	Access	Description	Reset
0	CLR_INTR	RO	read to clear corresponding all raw staus	
31:1	Reserved			

21.1.7.15 IC_CLR_RX_UNDER

Table 21.18: IC_CLR_RX_UNDER, Offset Address: 0x044

Bits	Name	Access	Description	Reset
0	CLR_RX_UNDER	RO	read to clear corresponding interupt raw staus, please reference I2C Raw Interrupt Status	
31:1	Reserved			

21.1.7.16 IC_CLR_RX_OVER

Table 21.19: IC_CLR_RX_OVER, Offset Address: 0x048

Bits	Name	Access	Description	Reset
0	CLR_RX_OVER	RO	read to clear corresponding interupt raw staus, please reference I2C Raw Interrupt Status	
31:1	Reserved			

21.1.7.17 IC_CLR_TX_OVER

Table 21.20: IC_CLR_TX_OVER, Offset Address: 0x04c

Bits	Name	Access	Description	Reset
0	CLR_TX_OVER	RO	read to clear corresponding interupt raw staus, please reference I2C Raw Interrupt Status	
31:1	Reserved			

21.1.7.18 IC_CLR_RD_REQ

Table 21.21: IC_CLR_RD_REQ, Offset Address: 0x050

Bits	Name	Access	Description	Reset
0	CLR_RD_REQ	RO	read to clear corresponding interupt raw staus, please reference I2C Raw Interrupt Status	
31:1	Reserved			

21.1.7.19 IC_CLR_TX_ABRT

Table 21.22: IC_CLR_TX_ABRT, Offset Address: 0x054

Bits	Name	Access	Description	Reset
0	CLR_TX_ABRT	RO	read to clear corresponding interupt raw staus, please reference I2C Raw Interrupt Status	
31:1	Reserved			

21.1.7.20 IC_CLR_RX_DONE

Table 21.23: IC_CLR_RX_DONE, Offset Address: 0x058

Bits	Name	Access	Description	Reset
0	CLR_RX_DONE	RO	read to clear corresponding interupt raw staus, please reference I2C Raw Interrupt Status	
31:1	Reserved			

21.1.7.21 IC_CLR_ACTIVITY

Table 21.24: IC_CLR_ACTIVITY, Offset Address: 0x05c

Bits	Name	Access	Description	Reset
0	CLR_ACTIVITY	RO	read to clear corresponding interupt raw staus, please reference I2C Raw Interrupt Status	
31:1	Reserved			

21.1.7.22 IC_CLR_STOP_DET

Table 21.25: IC_CLR_STOP_DET, Offset Address: 0x060

Bits	Name	Access	Description	Reset
0	CLR_STOP_DET	RO	read to clear corresponding interupt raw staus, please reference I2C Raw Interrupt Status	
31:1	Reserved			

21.1.7.23 IC_CLR_START_DET

Table 21.26: IC_CLR_START_DET, Offset Address: 0x064

Bits	Name	Access	Description	Reset
0	CLR_START_DET	RO	read to clear corresponding interupt raw staus, please reference I2C Raw Interrupt Status	
31:1	Reserved			

21.1.7.24 IC_CLR_GEN_CALL

Table 21.27: IC_CLR_GEN_CALL, Offset Address: 0x068

Bits	Name	Access	Description	Reset
0	CLR_GEN_CALL	RO	read to clear corresponding interupt raw staus, please reference I2C Raw Interrupt Status	
31:1	Reserved			

21.1.7.25 IC_ENABLE

Table 21.28: IC_ENABLE, Offset Address: 0x06c

Bits	Name	Access	Description	Reset
0	ENABLE	R/W	Enables I2C controller	0x0
31:1	Reserved			

21.1.7.26 IC_STATUS

Table 21.29: IC_STATUS, Offset Address: 0x070

Bits	Name	Access	Description	Reset
0	ST_ACTIVITY	RO	I2C Activity Status.	
1	ST_TFNF	RO	Transmit FIFO Not Full	
2	ST_TFE	RO	Transmit FIFO Completely Empty	
3	ST_RFNE	RO	Receive FIFO Not Empty	
4	ST_RFF	RO	Receive FIFO Completely Full	
5	ST_MST_ACTIVITY	RO	Master FSM Activity Status	
6	ST_SLV_ACTIVITY	RO	Slave FSM Activity Status	
31:7	Reserved			

21.1.7.27 IC_TXFLR

Table 21.30: IC_TXFLR, Offset Address: 0x074

Bits	Name	Access	Description	Reset
6:0	TXFLR	RO	I2C Transmit FIFO Level	
31:7	Reserved			

21.1.7.28 IC_RXFLR

Table 21.31: IC_RXFLR, Offset Address: 0x078

Bits	Name	Access	Description	Reset
6:0	RXFLR	RO	I2C Receive FIFO Level Register	
31:7	Reserved			

21.1.7.29 IC_SDA_HOLD

Table 21.32: IC_SDA_HOLD, Offset Address: 0x07c

Bits	Name	Access	Description	Reset
15:0	IC_SDA_HOLD	R/W	Sets the required SDA hold time in units of IP clock.	0x1
31:16	Reserved			

21.1.7.30 IC_TX_ABRT_SOURCE

Table 21.33: IC_TX_ABRT_SOURCE, Offset Address: 0x080

Bits	Name	Access	Description	Reset
15:0	TX_ABRT_SOURCE	RO	I2C Transmit Abort Source Register	
31:16	Reserved			

21.1.7.31 IC_SLV_DATA_NACK_ONLY

Table 21.34: IC_SLV_DATA_NACK_ONLY, Offset Address: 0x084

Bits	Name	Access	Description	Reset
0	NACK	R/W	generate a NACK in slave-receiver mode	0x0
31:1	Reserved			

21.1.7.32 IC_DMA_CR

Table 21.35: IC_DMA_CR, Offset Address: 0x088

Bits	Name	Access	Description	Reset
0	RDMAE	R/W	Receive DMA Enable	0x0
1	TDMAE	R/W	Transmit DMA Enable	0x0
31:2	Reserved			

21.1.7.33 IC_DMA_TDLR

Table 21.36: IC_DMA_TDLR, Offset Address: 0x08c

Bits	Name	Access	Description	Reset
5:0	DMATDL	R/W	the dma_tx_req signal is generated when the number of valid data entries in the transmit FIFO is equal to or below this field value	0x0
31:6	Reserved			

21.1.7.34 IC_DMA_RDLR

Table 21.37: IC_DMA_RDLR, Offset Address: 0x090

Bits	Name	Access	Description	Reset
5:0	DMARDL	R/W	dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or more than this field value + 1	0x0
31:6	Reserved			

21.1.7.35 IC_SDA_SETUP

Table 21.38: IC_SDA_SETUP, Offset Address: 0x094

Bits	Name	Access	Description	Reset
0	SDA_SETUP	R/W	SDA Setup time config register	0x64
31:1	Reserved			

21.1.7.36 IC_ACK_GENERAL_CALL

Table 21.39: IC_ACK_GENERAL_CALL, Offset Address: 0x098

Bits	Name	Access	Description	Reset
0	ACK_GEN_CALL	R/W	When set to 1, DW_apb_i2c responds with a ACK when it receives a General Call. When set to 0, the IP does not generate General Call interrupts	0x1
31:1	Reserved			

21.1.7.37 IC_ENABLE_STATUS

Table 21.40: IC_ENABLE_STATUS, Offset Address: 0x09c

Bits	Name	Access	Description	Reset
0	IC_EN	RO	I2C Enable Status Register	
1	SLV_DISABLED_WHILE_BUSY	RO	Slave Disabled While Busy (Transmit, Receive)	
2	SLV_RX_DATA_LOST	RO	Slave Received Data Lost.	
31:3	Reserved			

21.1.7.38 IC_FS_SPKLEN

Table 21.41: ic_fs_spklen, Offset Address: 0x0a0

Bits	Name	Access	Description	Reset
7:0	IC_FS_SPKLEN	R/W	I2C SS and FS Spike Suppression Limit Register	0x5
31:8	Reserved			

21.1.7.39 IC_HS_SPKLEN

Table 21.42: IC_HS_SPKLEN, Offset Address: 0x0a4

Bits	Name	Access	Description	Reset
7:0	IC_HS_SPKLEN	R/W	I2C HS Spike Suppression Limit Register	0x1
31:8	Reserved			

21.2 UART

21.2.1 Overview

UART (Universal Asynchronous Receiver Transmitter) is an asynchronous serial communication interface. Its main function is to convert data from peripheral devices to serial and then transfer it to the internal bus, and to convert data from parallel to serial and then output it to external devices. The main function of UART is to interface with the UART of an external chip to achieve communication between the two chips.

This chip provides 5 UART controllers. The relevant overview is as follows. Note: Please refer to the specific pin output definition before use. Due to different chip packages (QFN/BGA), some functions may not be exported. For pin definition, refer to *PinMux and PinCtrl*.

Table 21.43: UART IO pin information

Controller	Mode
UART0	2-line UART
UART1	2/4-line UART
UART2	2/4-line UART
UART3	2/4-line UART
UART4	2-line UART

21.2.2 Features

The UART module has the following features:

- Supports 64 x 8bit transmit FIFO and 64 x 8bit receive FIFO.
- Supports programmable bit width of data bits and stop bits. The data bits can be set to 5/6/7/8 bits through the program.
- The stop bit can be set to 1bit, 1.5bit or 2bit through programming.
- Supports odd, even checksum or no checksum.
- Supports programmable transmission rate settings.
- Supports receive FIFO interrupt, transmit FIFO interrupt, and error interrupt.
- Supports initial interrupt status query and post-mask interrupt status query.
- Support DMA operation.

21.2.3 Function Description

21.2.3.1 Application Block Diagram

UART is a universal point-to-point physical layer transmission protocol that can be used to interface with various systems, including PCs and various peripheral chips, and can be used as a communication interface between chips.

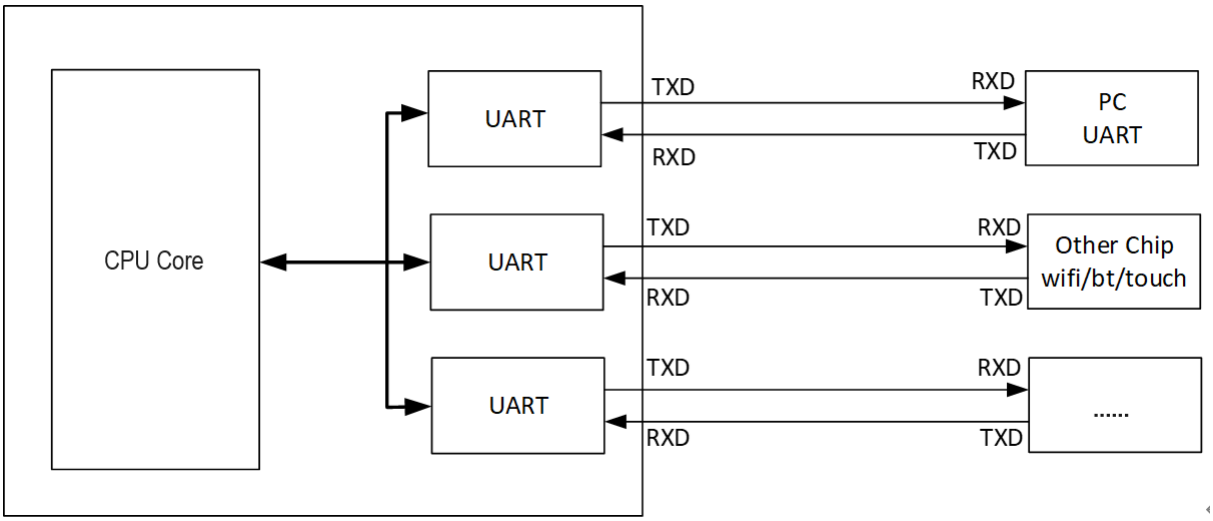


Diagram 21.5: UART application block diagram

21.2.3.2 Functional Principle

- Baud Rate

  Since the UART interface does not have a reference clock and is an asynchronous transmission method, both parties need to use the same transmission speed, that is, the baud rate (buadrate) for communication. If there is an error, the error rate needs to be small enough to avoid misinformation. The rate of 1 bit is called baudrate. Typical baud rates are 300, 1200, 2400, 9600, 19200, 38400, 115200bps, etc.

- Frame Structure

  The UART transmission data structure is in frames. The frame structure includes start signal, data signal, check bit and end signal.

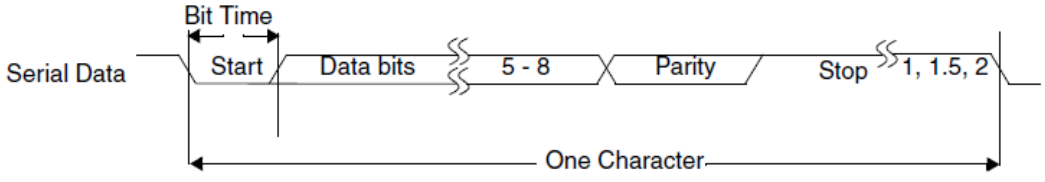


Diagram 21.6: UART transfer data structure

- Start signal (start bit)

The start signal is the mark of the beginning of a frame. The very beginning of initiating a frame transmission is to send a low-level signal bit on TXD. On RXD, if a low level signal bit is received in the idle state, it is judged as receiving the start of a detection transmission.

- Data signal (data bit)

  The data bit width can be adjusted according to different application requirements, and can be 5/6/7/8 bit data bit width. Typically 8-bit data width.

- Parity bit

  The check bit is a 1-bit error correction signal. The check bits of the UART include odd parity check, even parity and fixed check bits. It also supports the enable and disable of the check bit. For detailed description, please see the LCR register.

- End signal (stop bit)

  The end signal is the stop bit of the frame, supporting 1-bit, 1.5-bit and 2-bit stop bits. To send the end signal of a frame is to send TXD high level to complete the transmission and enter the idle state. After receiving a frame and counting the check bits, the end signal needs to be received.

21.2.4 Way of Working

21.2.4.1 Baud Rate Configuration

- UART working clock (UART_SCLK) configuration

  You can refer to the CLK_DIV CRG register description to configure clk_sel_0_9~ clk_sel_0_13 to select the working clock of uart0~uart4. The default is 1: XTAL 25MHz. If configured as 0, the UART PLL divided clock source is selected. The PLL frequency division clock source is preset to 187.5MHz. If necessary, you can configure the frequency division register div_clk_187p5m to adjust the PLL frequency division clock to 1500/NMHz, up to 187.5MHz.

- UART baud rate configuration

  DLL and DLH are the baud rate division control registers inside the UART controller. DLH is the high 8 bits and DLL is the low 8 bits. Before configuring DLH and DLL, LCR[7] must be configured to 1. At this time, the registers RBR_THR_DLL(DLL) and IER_DLH(DLH) can be configured.

  After the configuration is completed, the baud rate is set. The formula is:

$$Baud\,rate = \frac{UART\_SCLK}{16 * (256 * DLH + DLL)}$$

- Taking UART SCLK 25MHz as an example and configuring a baud rate of 115200, the formula is:

$$(256 * DLH + DLL) = \frac{25M}{16 * 115200} = 13.5$$

  If you choose to configure DLL as 14 and DLH as 0, the actual baud rate is:

$$Baud\,rate = \frac{25M}{16 * 14} = 111607$$

  The one-bit time error is:

$$Bit\ Error = \frac{(115200 - 114286)}{115200} = 3.12\%$$

  The accumulated time error of one frame is: $Frame\ Error =$ 3.12%*10 = 31.2%

- Taking UART SCLK 187.5MHz as an example, configure a baud rate of 115200, and the formula is:

$$(256 * DLH + DLL) = \frac{187.5M}{16 * 115200} = 101.7$$

If you choose to configure DLL as 102 and DLH as 0, the actual baud rate is:

$$Baud\ rate = \frac{187.5M}{16 * 102} = 114890$$

The one-bit time error is:

$$Bit\ Error = \frac{(115200 - 114890)}{115200} = 0.27\%$$

The accumulated time error of one frame is: $Frame\ Error =$ 3.12%*10 = 2.7%

21.2.4.2 Data Sending Flow Chart

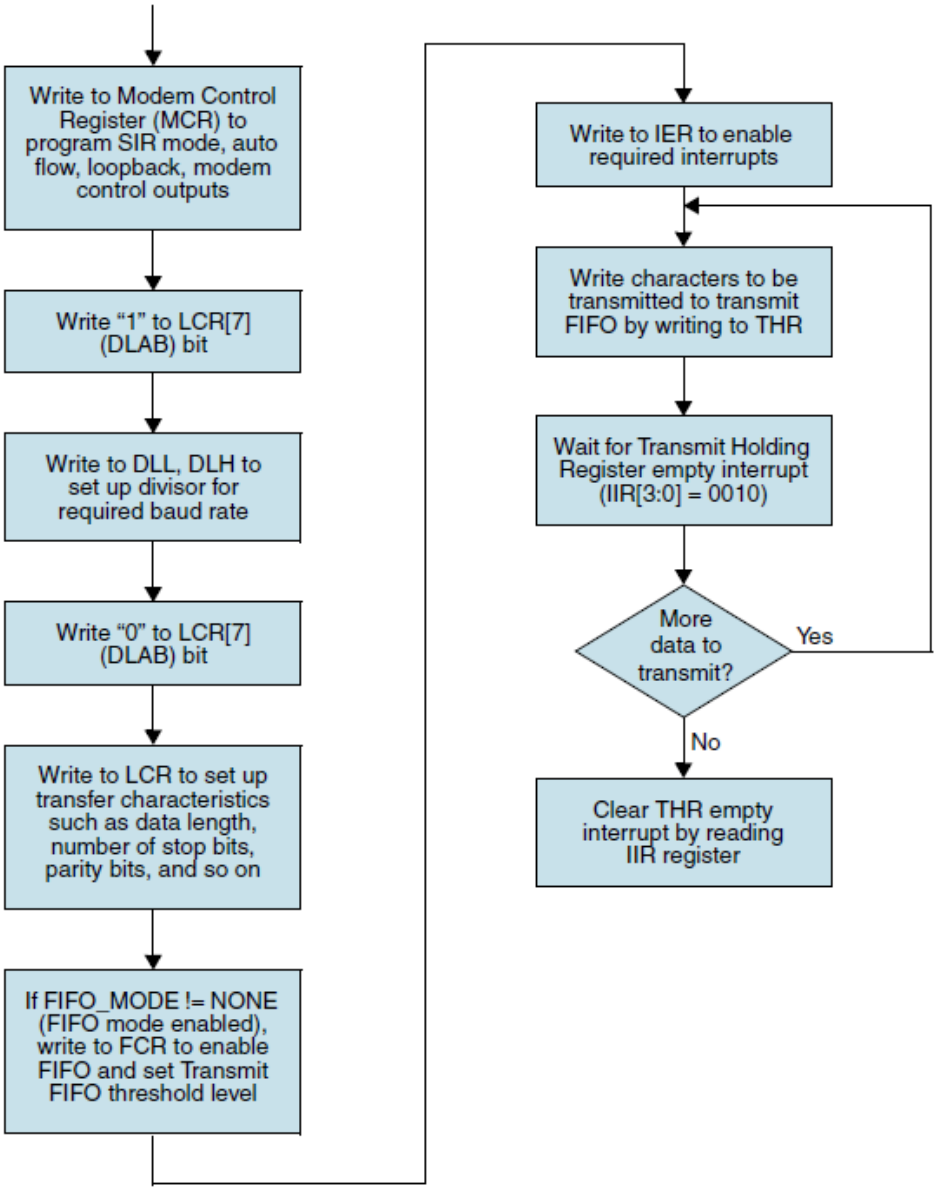


Diagram 21.7: UART data sending flow chart

21.2.4.3 Data Receiving Flow Chart

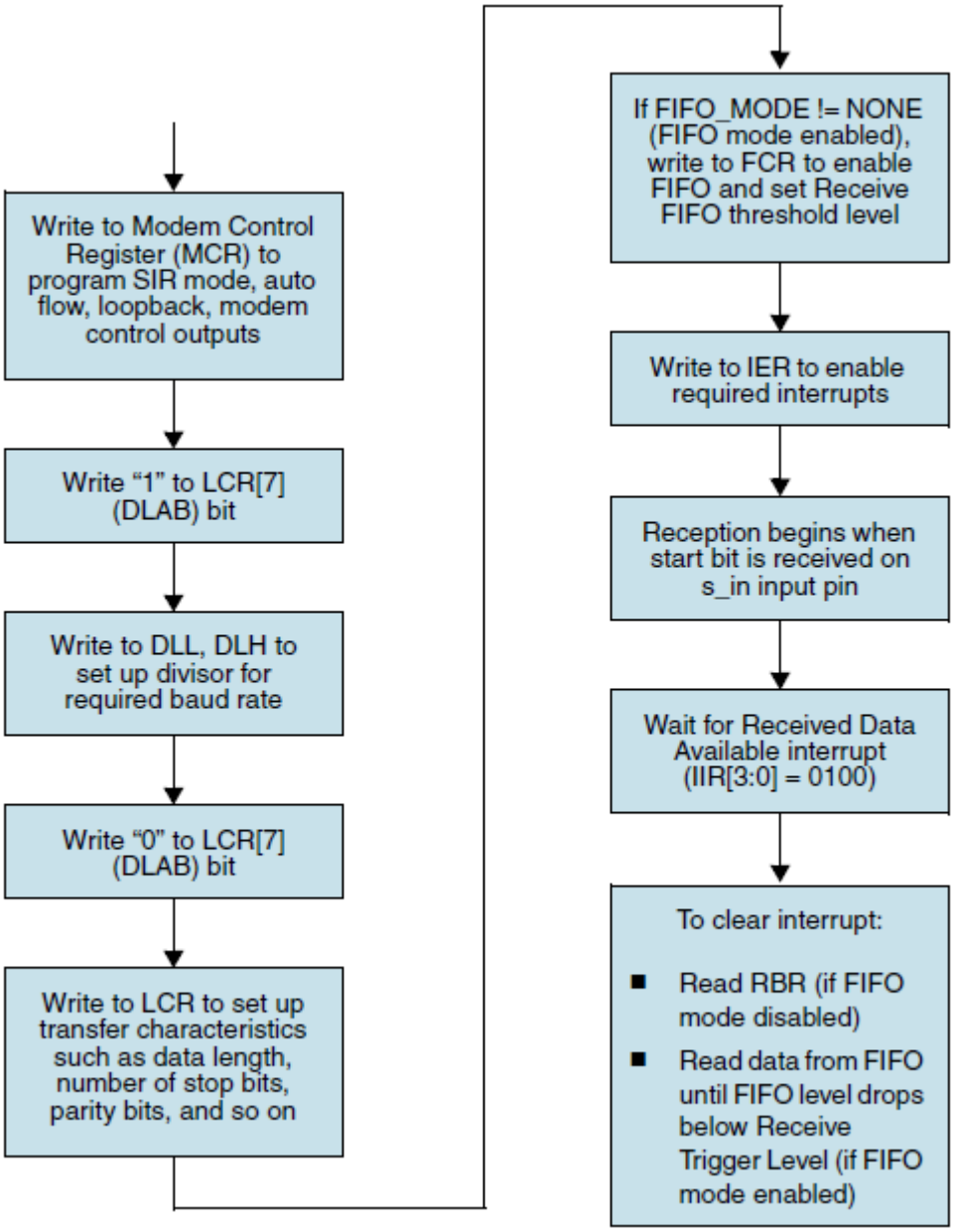


Diagram 21.8: UART data Receiving Fflow chart

21.2.4.4 Data transfer in interrupt or query mode

Initialization steps

1. Write 1 to LCR[7]. Enable configuration of Divisor Latch Access.
2. Write the corresponding configuration values to the RBR_THR_DLL and IER_DLH registers to configure the transmission baud rate.

3. Write 0 to LCR[7].
4. Configure LCR and set the corresponding UART working mode
5. Configure FCR and set the corresponding transmit and receive FIFO thresholds.
6. If you use the interrupt mode, you need to set IER and enable the corresponding interrupt signal;

Data sending

1. When LCR[7] is 0, write the transmission data to RBR_THR_DLL (Transmit Holding Register) to start data transmission.
2. If the query method is used, detect the TX_FIFO status by reading USR[1] (Transmit FIFO not full) and TFL (Transmit FIFO Level), and decide whether to continue writing data to RBR_THR_DLL based on the status of TX_FIFO;
3. If the interrupt mode is used, detect the corresponding interrupt status bit; decide whether to continue writing data to RBR_THR_DLL.
4. Determine whether the UART has completed sending all data by detecting USR[2] (Transmit FIFO Empty).

Data reception

1. If the query method is used, the RX_FIFO status is detected by reading USR[3] (Receive FIFO Not Empty) and RFL (Receive FIFO Level), and based on the status of RX_FIFO, it is decided whether to read RBR_THR_DLL (Receive Buffer Register) to obtain the data.
2. If the interrupt mode is used, the corresponding interrupt status bit detection is used to determine whether to read RBR_THR_DLL (Receive Buffer Register) and obtain the data.

21.2.4.5 Data transfer in DMA mode

Initialization steps

1. Write 1 to LCR[7]. Enable configuration of Divisor Latch Access
2. Write the corresponding configuration values to the RBR_THR_DLL and IER_DLH registers to configure the transmission baud rate.
3. Write 0 to LCR[7].
4. Configure LCR and set the corresponding UART working mode
5. Configure FCR and set the corresponding transmit and receive FIFO thresholds.
6. Close ETBEI/ERBFI in IER;

Data sending

1. Configure system DMA channel mapping. Refer to the system DMA channel mapping and configure the selected UART controller TX/RX request line number to the corresponding system DMA channel. For example: UART0 TX configures system DMA channel 3, then sdma_dma_ch_remap0[29:24]=9. After the configuration is completed, update_dma_remp_0_3 needs to be configured to make the configuration effective.
2. Configure the system DMA data channel, including data transmission source and destination addresses, data transmission number, transmission type and other parameters. Please refer to the DMA controller chapter for specific configuration.
3. Use the system DMA interrupt report to determine whether the data transmission is completed.

Data reception

1. Configure system DMA channel mapping. Refer to the DMA channel mapping and configure the selected UART controller TX/RX request line number to the corresponding system DMA channel. For example: UART0 RX configures system DMA channel 1, then sets sdma_dma_ch_remap0[13:8] = 8. After the configuration is completed, update_dma_remp_0_3 needs to be configured to make the configuration effective.
2. Configure the system DMA data channel, including data transmission source and destination addresses, data receiving area address, data transmission number, transmission type and other parameters. Please refer to the DMA controller chapter for specific configuration.
3. Determine whether data reception is completed through system DMA interrupt reporting.

21.2.5 UART Register Overview

Includes 6 UARTs, 5 Active Domains, and 1 No-die Domain. Their base addresses are as follows. Each UART consists of a set of control registers, each set identically defined.

Table 21.44: Base addresses for 6 sets of UART module

GPIO Module	Base Address
UART0	0x04140000
UART1	0x04150000
UART2	0x04160000
UART3	0x04170000
UART4	0x041C0000
RTCSYS_UART	0x05022000

Table 21.45: UART Registers Overview

Name	Address Offset	Description
RBR_THR_DLL	0x000	Receive Buffer,Transmit Holding or Divisor Latch Low byte Register
IER_DLH	0x004	Interrupt Enable or Divisor Latch high byte Register
FCR_IIR	0x008	FIFO Control or Interrupt Identification Register
LCR	0x00c	Line Control Register
MCR	0x010	Modem Control Register
LSR	0x014	Line Status Register
MSR	0x018	Modem Status Register
LPDLL	0x020	Low Power Divisor Latch (Low) Register
LPDLH	0x024	Low Power Divisor Latch (High) Register
SRBR_STHR	0x030	Shadow Receive/Trasnmit Buffer Register
FAR	0x070	FIFO Access Register
TFR	0x074	Transmit FIFO Read
RFW	0x078	Receive FIFO Write
USR	0x07c	UART Status Register
TFL	0x080	Transmit FIFO Level
RFL	0x084	Receive FIFO Level
SRR	0x088	Software Reset Register
SRTS	0x08c	Shadow Request to Send
SBCR	0x090	Shadow Break Control Register
SDMAM	0x094	Shadow DMA Mode
SFE	0x098	Shadow FIFO Enable
SRT	0x09c	Shadow RCVR Trigger
STET	0x0a0	Shadow TX Empty Trigger
HTX	0x0a4	Halt TX
DMASA	0x0a8	DMA Software Acknowledge

21.2.6 UART Register Description

21.2.6.1 RBR_THR_DLL

Table 21.46: RBR_THR_DLL, Offset Address: 0x000

Bits	Name	Access	Description	Reset
7:0	RBR_THR_DLL	R/W	LCR[7] bit = 0 : (R) Receive Buffer Register ,Data byte received on the serial input port (W)Transmit Holding Register,Data to be transmitted on the serial output port LCR[7] bit = 1 : Lower 8 bits of a 16-bit Divisor Latch register that contains the baud rate divisor for the UART	0x0
31:8	Reserved			

21.2.6.2 IER_DLH

Table 21.47: IER_DLH, Offset Address: 0x004

Bits	Name	Access	Description	Reset
7:0	IER_DLH	R/W	LCR[7] bit = 0 : IER[0] : Enable Received Data Available Interrupt. IER[1] : Enable Transmit Holding Register Empty Interrupt. IER[2] : Enable Receiver Line Status Interrupt. IER[3] : Enable Modem Status Interrupt. IER[7] : Programmable THRE Interrupt Mode Enable LCR[7] bit = 1 : Upper 8-bits of a 16-bit Divisor Latch register that contains the baud rate divisor for the UART.	0x0
31:8	Reserved			

21.2.6.3 FCR_IIR

Table 21.48: FCR_IIR, Offset Address: 0x008

Bits	Name	Access	Description	Reset
7:0	FCR_IIR	R/W	(R) interrupt Identification Register [3:0] Interrupt ID 0000 : modem status 0001 : no interrupt pending 0010 : THR empty 0100 : received data available 0110 : receiver line status 0111 : busy detect 1100 : character timeout [7:6] FIFOs Enabled 00 - disabled 11 - enable (W) FIFO Control Register [0] FIFO Enable [1] RCVR FIFO Reset [2] XMIT FIFO Reset [3] DMA Mode 0 - mode 0, single DMA data transfers at a time 1 - mode 1, multi DMA data transfers are made continuously [5:4] TX Empty 00 - FIFO empty 01 - 2 characters in the FIFO 10 - FIFO ¼ full 11 - FIFO ½ full [7:6] RCVR Trigger 00 - 1 character in the FIFO 01 - FIFO ¼ full 10 - FIFO ½ full 11 - FIFO 2 less than full	0x1
31:8	Reserved			

21.2.6.4 LCR

Table 21.49: LCR, Offset Address: 0x00c

Bits	Name	Access	Description	Reset
7:0	LCR	R/W	Line Control Register [1:0] Data Length Select. (00:5 bits,01:6 bits,10:7 bits,11:8 bits) [2] Number of stop bits. (0:1 stop bit,1:1.5 stop bits when Data Length Select is 0, else 2 stop bits) [3] Parity Enable [4] Even Parity Select [5] Stick Parity [6] Break Control Bit [7] Divisor Latch Access Bit	0x0
31:8	Reserved			

21.2.6.5 MCR

Table 21.50: MCR, Offset Address: 0x010

Bits	Name	Access	Description	Reset
7:0	MCR	R/W	Modem Control Register [0] reserved [1] Request to Send. This is used to directly control the Request to Send (rts_n) output [2] reserved [3] reserved [4] reserved [5] Auto Flow Control Enable. [6] reserved	0x0
31:8	Reserved			

21.2.6.6 LSR

Table 21.51: LSR, Offset Address: 0x014

Bits	Name	Access	Description	Reset
7:0	LSR	RO	Line Status Register [0] Data Ready bit. there is at least one character in the RBR or the receiver FIFO. [1] Overrun error bit. This is used to indicate the occurrence of an overrun error. [2] Parity Error bit. [3] Framing Error bit.. [4] Break Interrupt bit. [5] Transmit Holding Register Empty bit. [6] Transmitter Empty bit. [7] Receiver FIFO Error bit.	
31:8	Reserved			

21.2.6.7 MSR

Table 21.52: MSR, Offset Address: 0x018

Bits	Name	Access	Description	Reset
7:0	MSR	RO	Modem Status Register [0] Delta Clear to Send. [1] reserved [2] reserved [3] reserved [4] CTS [5] reserved [6] reserved [7] reserved	
31:8	Reserved			

21.2.6.8 LPDLL

Table 21.53: LPDLL, Offset Address: 0x020

Bits	Name	Access	Description	Reset
7:0	LPDLL	R/W	LCR[7] bit = 1 : Low Power Divisor Latch (Low) Register	0x0
31:8	Reserved			

21.2.6.9 LPDLH

Table 21.54: LPDLH, Offset Address: 0x024

Bits	Name	Access	Description	Reset
7:0	LPDLH	R/W	LCR[7] bit = 1 : Low Power Divisor Latch (High) Register	0x0
31:8	Reserved			

21.2.6.10 SRBR_STHR

Table 21.55: SRBR_STHR, Offset Address: 0x030

Bits	Name	Access	Description	Reset
7:0	SRBR_STHR	R/W	LCR[7] bit = 0 : (R) Shadow Receive Buffer Register (W) Shadow Transmit Holding Register	0x0
31:8	Reserved			

21.2.6.11 FAR

Table 21.56: FAR, Offset Address: 0x070

Bits	Name	Access	Description	Reset
0	FAR	R/W	FIFO Access Register,This register is use to enable a FIFO access mode for testing	0x0
31:1	Reserved			

21.2.6.12 TFR

Table 21.57: TFR, Offset Address: 0x074

Bits	Name	Access	Description	Reset
7:0	TFR	R/W	Transmit FIFO Read. These bits are only valid when FIFO access mode is enabled	0x0
31:8	Reserved			

21.2.6.13 RFW

Table 21.58: RFW, Offset Address: 0x078

Bits	Name	Access	Description	Reset
9:0	RFW	R/W	Receive FIFO Write. These bits are only valid when FIFO access mode is enabled [7:0] Receive FIFO Write Data. [8] Receive FIFO Parity Error. [9] Receive FIFO Framing Error.	0x0
31:10	Reserved			

21.2.6.14 USR

Table 21.59: USR, Offset Address: 0x07c

Bits	Name	Access	Description	Reset
4:0	USR	RO	UART Status Register [0] UART Busy. [1] Transmit FIFO Not Full. [2] Transmit FIFO Empty. [3] Receive FIFO Not Empty. [4] Receive FIFO Full.	
31:5	Reserved			

21.2.6.15 TFL

Table 21.60: TFL, Offset Address: 0x080

Bits	Name	Access	Description	Reset
5:0	TFL	RO	Transmit FIFO Level. This is indicates the number of data entries in the transmit FIFO.	
31:6	Reserved			

21.2.6.16 RFL

Table 21.61: TFL, Offset Address: 0x084

Bits	Name	Access	Description	Reset
5:0	RFL	RO	Receive FIFO Level. This is indicates the number of data entries in the receive FIFO.	
31:6	Reserved			

21.2.6.17 SRR

Table 21.62: SRR, Offset Address: 0x088

Bits	Name	Access	Description	Reset
2:0	SRR	R/W	Software Reset Register [0] UART Reset. [1] RCVR FIFO Reset. [2] XMIT FIFO Reset.	0x0
31:3	Reserved			

21.2.6.18 SRTS

Table 21.63: SRTS, Offset Address: 0x08c

Bits	Name	Access	Description	Reset
0	SRTS	R/W	Shadow Request to Send. This is a shadow register for the RTS bit (MCR[1])	0x0
31:1	Reserved			

21.2.6.19 SBCR

Table 21.64: SBCR, Offset Address: 0x090

Bits	Name	Access	Description	Reset
0	SBCR	R/W	Shadow Break Control Bit. This is a shadow register for the Break bit (LCR[6]).	0x0
31:1	Reserved			

21.2.6.20 SDMAM

Table 21.65: SDMAM, Offset Address: 0x094

Bits	Name	Access	Description	Reset
0	SDMAM	R/W	Shadow DMA Mode. This is a shadow register for the DMA mode bit (FCR[3]).	0x0
31:1	Reserved			

21.2.6.21 SFE

Table 21.66: SFE, Offset Address: 0x098

Bits	Name	Access	Description	Reset
0	SFE	R/W	Shadow FIFO Enable. This is a shadow register for the FIFO enable bit (FCR[0]).	0x0
31:1	Reserved			

21.2.6.22 SRT

Table 21.67: SRT, Offset Address: 0x09c

Bits	Name	Access	Description	Reset
1:0	SRT	R/W	Shadow RCVR Trigger. This is a shadow register for the RCVR trigger bits (FCR[7:6]).	0x0
31:2	Reserved			

21.2.6.23 STET

Table 21.68: STET, Offset Address: 0x0a0

Bits	Name	Access	Description	Reset
1:0	STET	R/W	Shadow TX Empty Trigger. This is a shadow register for the TX empty trigger bits (FCR[5:4]).	0x0
31:2	Reserved			

21.2.6.24 HTX

Table 21.69: HTX, Offset Address: 0x0a4

Bits	Name	Access	Description	Reset
0	HTX	R/W	This register is use to halt transmissions for testing,	0x0
31:1	Reserved			

21.2.6.25 DMASA

Table 21.70: DMASA, Offset Address: 0x0a8

Bits	Name	Access	Description	Reset
0	DMASA	R/W	This register is use to perform a DMA software acknowledge if a transfer needs to be terminated due to an error condition.	0x0
31:1	Reserved			

21.3 SPI

21.3.1 Overview

The system is configured with 4 SPI controller modules, which can be used as Master to conduct synchronous serial communication with external devices to achieve serial-to-parallel and parallel-to-serial conversion of data.

21.3.2 Features

The features of the SPI controller module are as follows:

- Supports three serial peripheral interface protocols: Motorola SPI (full duplex), TI SSP (full duplex), and NS MicroWire (half duplex)
- Independent receive/transmit FIFO
- Programmable data frame length: 4~16 bits
- SPI interface clock frequency programmable
- Support DMA operation mode
- Support internal loopback test mode
- The working reference clock can be set to 187.5MHz or 100MHz, and the output SPI_SCK supports a maximum of 46.875MHz

21.3.3 Function Description

21.3.3.1 Typical Application

The application block diagram when the SPI master connects to the external slave is as shown in the chart *SPI application block diagram*

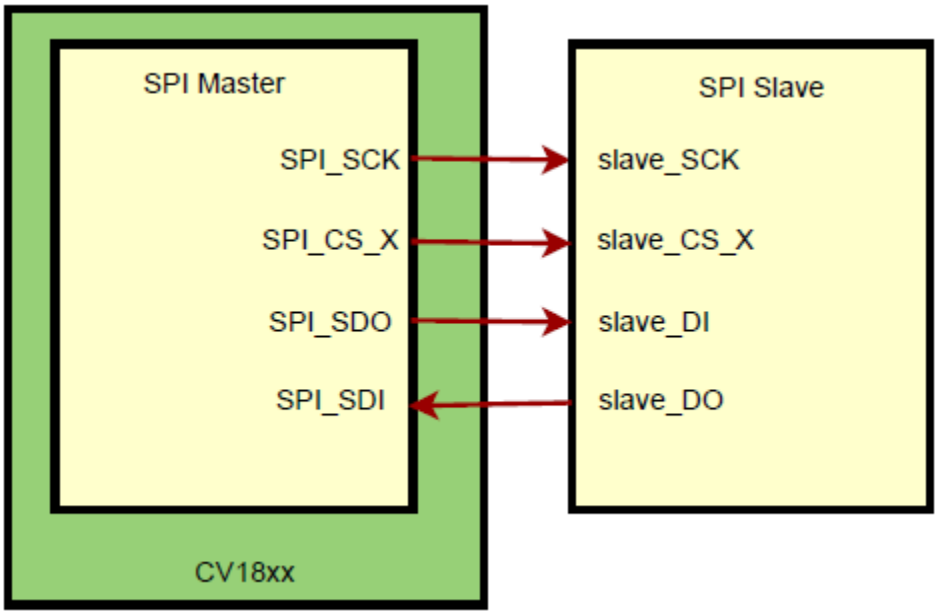


Diagram 21.9: SPI application block diagram

21.3.4 Way of Working

21.3.4.1 Operating Mode

The working modes of SPI are divided into:

- Data transmission in interrupt or query mode
- Data transfer in DMA mode

21.3.4.2 Clock

The SPI controller module reference clock can be set to 187.5MHz or 100MHz.

The output SPI_SCK supports up to 46.875MHz.

The calculation is as follows:

Output SPI_SCK = SPI working reference clock / BAUDR

SPI working reference clock : 187.5MHz or 100MHz.

BAUDR register : Set an even number between 2 and 65534.

Calculation example:

SPI working reference clock = 187.5MHz and BAUDR =4

Output SPI_SCK = 187.5MHz / 4 = 46.875MHz

21.3.4.3 Interrupt Handling

The SPI controller module has 6 interrupts, the first 5 of which are active high maskable independent interrupt sources.

- RXFINTR

  Receive FIFO interrupt request. This interrupt is set when there is RXFTLR+1 or more valid data in the receive FIFO.

- RXOINTR

  When the receive FIFO is full and new data needs to be written to the FIFO, FIFO Overflow will occur and this interrupt is set. At this time data is written to the receive shift register instead of the FIFO.

- RXUINTR

  When the receive FIFO is read empty and no new data is written to the receive FIFO before a new read request occurs, FIFO Underflow will be caused and the interrupt will be set. The values read at this time are all 0. This interrupt can be cleared by reading register RXUICR.

- TXOINTR

  When the transmit FIFO is full and new data needs to be written to the FIFO, FIFO Overflow will occur and this interrupt will be set.

- TXEINTR

  Send FIFO interrupt request. This interrupt is set when there is TXFTLR or less valid data in the transmit FIFO.

- SPI_INTR

  The combined interrupt is the result of the "OR" operation of the above five interrupts. To mask this interrupt, register IMR must be set to mask the above 5 interrupts. This interrupt is set if any of the above 5 independent interrupts is set and enabled.

21.3.4.4 Initialization

The SPI controller module initialization steps are as follows:

- Step 1: Set register SPIENR to "0", stop the SPI module.
- Step 2: Configure the register BAUDR to set the output clock frequency division divisor. The set value must be an even number.
- Step 3: Set register CTRLR0 to configure parameters such as transmission data bit width and transmission frame format.
- Step 4: In DMA operation mode, configure the register DMACR to enable the DMA function of SPI. When operating in DMA mode, the interrupt-related registers should be set to disable interrupt signal generation.
- Step 5: In interrupt operation mode, set the register IMR to generate the corresponding interrupt signal.
- Step 6: Set register SPIENR to "1" to enable the SPI module.

21.3.4.5 SPI data transmission process

- The process when the SPI master connects to the external SPI/SSP slave is shown in the diagram *Data transmission process when connecting to external SPI/SSP slave*.

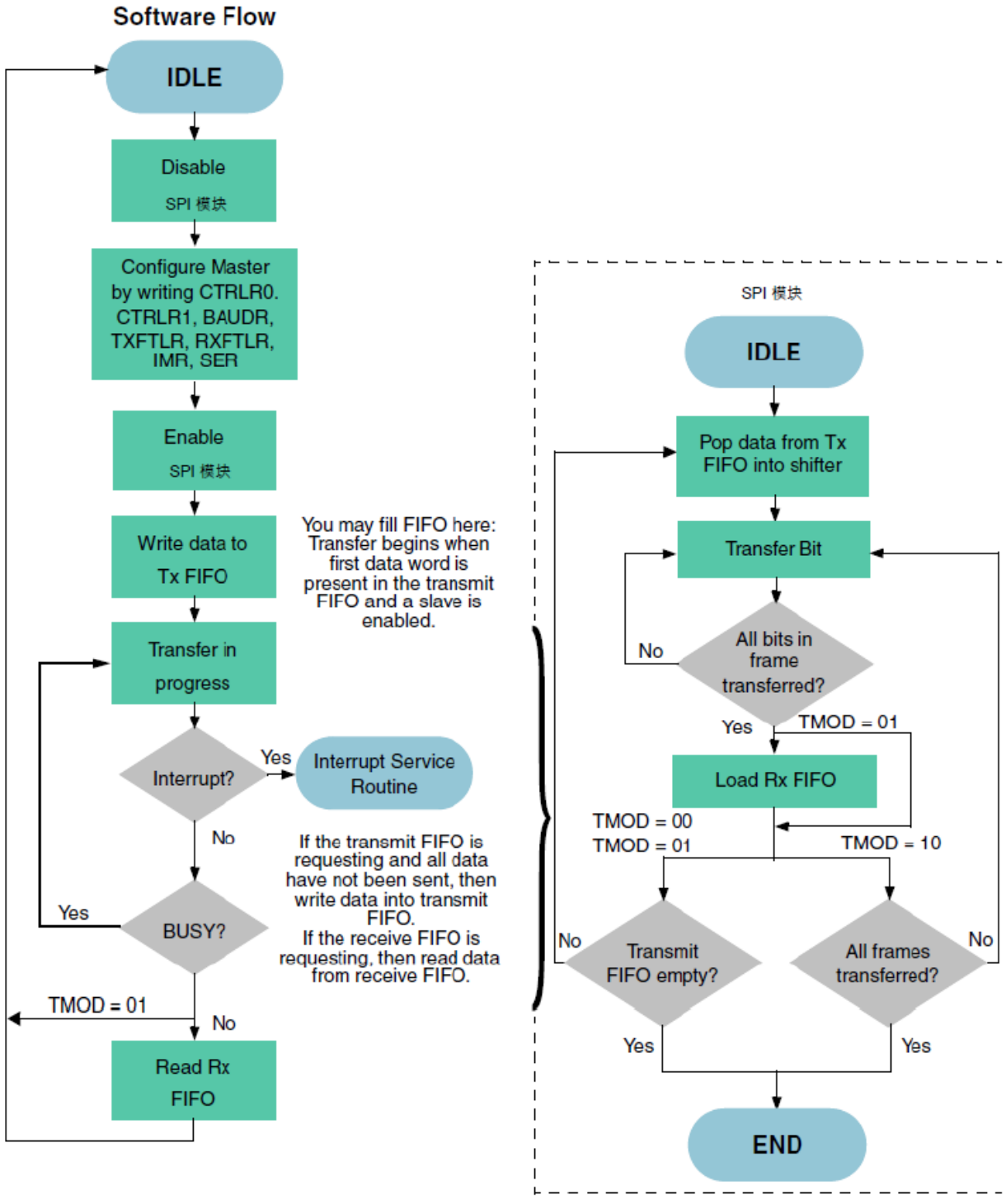


Diagram 21.10: Data transmission process when connecting to external SPI/SSP slave

- The process when the SPI master connects to the external Microwire slave is shown in the diagram *Data transmission process when connecting to external Microwire slave*.

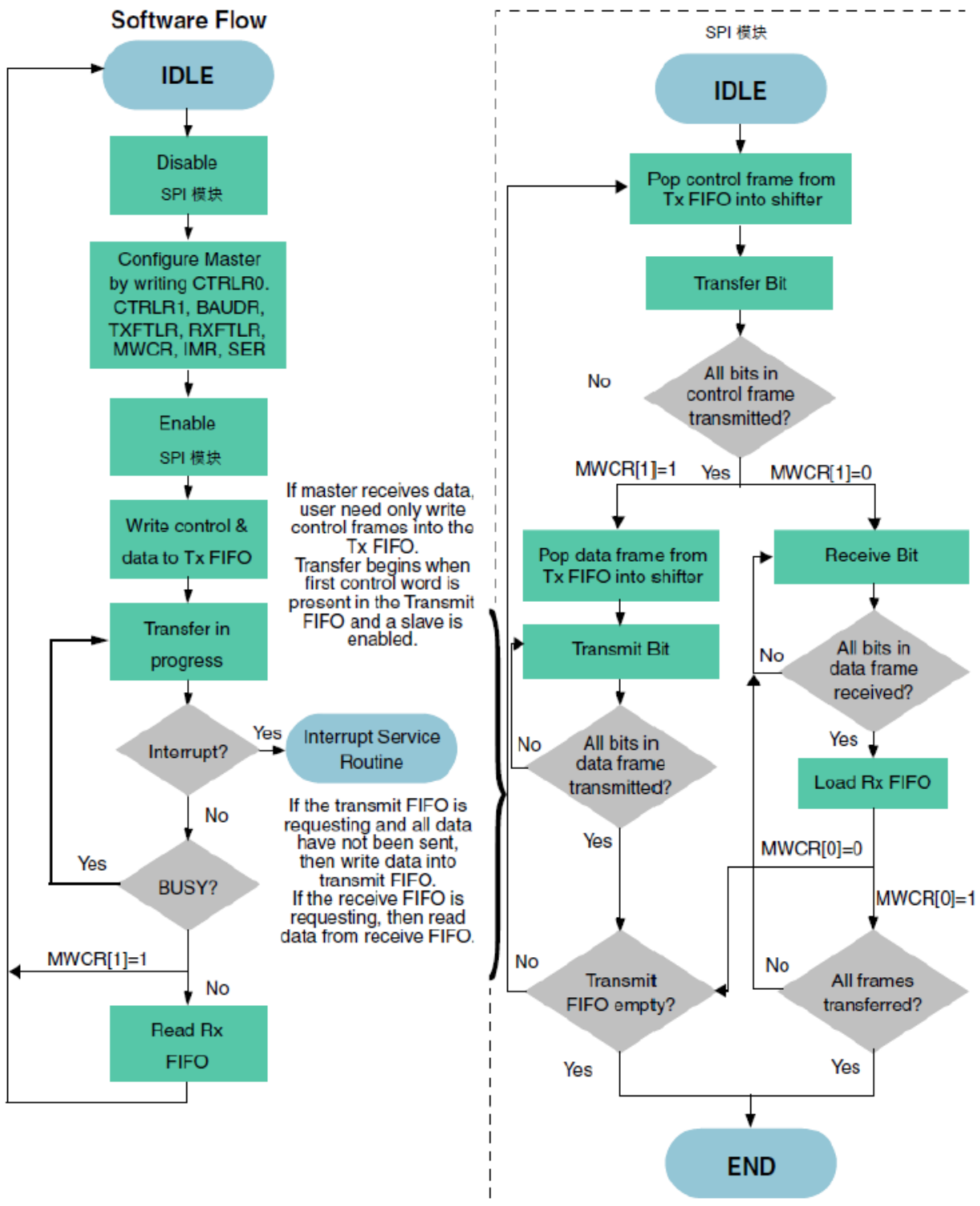


Diagram 21.11: Data transmission process when connecting to external Microwire slave

21.3.4.6 Data transfer in DMA mode

The SPI module uses two DMA channels, one for transmitting and one for receiving. The relevant registers for SPI DMA mode settings are DMACR, DMATDLR, and DMARDLR.

The steps to enable SPI DMA mode are as follows:

- Step 1: Get two DMA channels.
- Step 2: Set register DMACR [1:0] to enable SPI DMA transmission and reception.
- Step 3: Set register SPIENR to “1” to enable SPI.
- Step 4: Send data:
  1. Configure the control registers related to the sending DMA channel.
  2. Start the DMA controller and respond to the SPI send FIFO request.
  3. Use the DMA controller interrupt report to determine whether the transmission is completed. If it is completed, close the SPI transmission DMA function.
- Step 5: Receive data:
  1. Configure the control registers related to the receive DMA channel.
  2. Start the DMA controller and respond to the DMA request of the SPI receive FIFO.
  3. Use the DMA controller interrupt report to determine whether the data reception is completed. If it is completed, close the SPI receiving DMA function.
- Step 6: Set register SPIENR to “0” to stop SPI.

21.3.5 Three serial peripheral bus timing diagrams

21.3.5.1 Motorola SPI interface

The following figures represent the various data transmission formats of Motorola SPI. Among them, SCPH represents the SPI_SCK phase, and SCPOL represents the SPI_SCK polarity, which is set through the register CTRLR0[7:6].

(A) SCPH = 0

In this mode, SPI_CS_X is set to high level when in idle state and set to low level when transmitting. SPI_SCK is different through the SCPOL setting. SCPOL = 0, it is set to low level when in idle state. During transmission, data is captured on the rising edge of the clock. SCPOL = 1, set to high level when in idle state. , when transmitting, the data is captured on the falling edge of the clock.

- The single frame transmission format is shown in the diagram *Motorola SPI single frame transmission format (SCPH = 0)*.

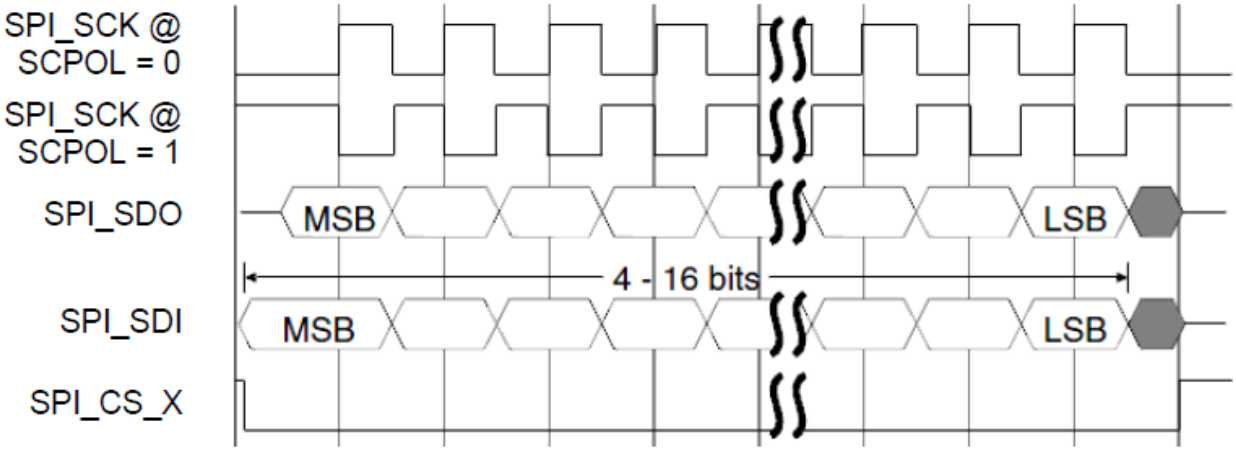


Diagram 21.12: Motorola SPI single frame transmission format (SCPH = 0)

- The continuous frame transmission format is shown in the diagram *Motorola SPI continuous frame transmission format (SCPH = 0)*.

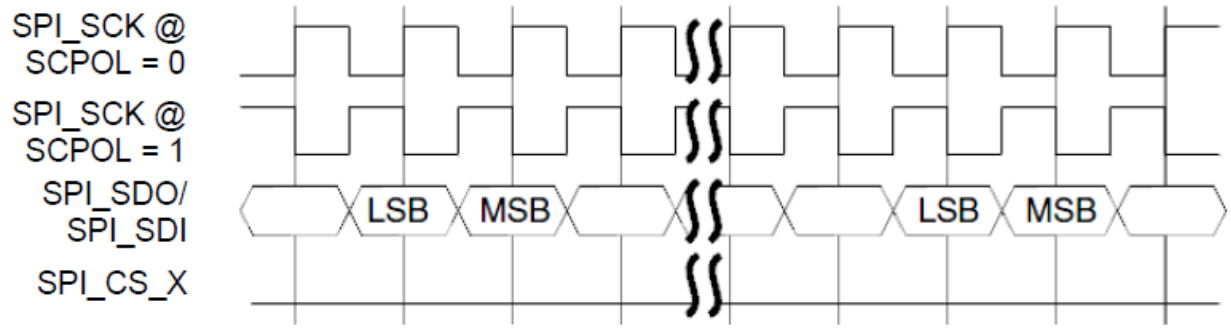


Diagram 21.13: Motorola SPI continuous frame transmission format (SCPH = 0)

(B) SCPH = 1

In this mode, SPI_CS_X is set to high level when in idle state and set to low level when transmitting. SPI_SCK is different through the SCPOL setting. SCPOL = 0, it is set to low level when in idle state, and data is captured on the falling edge of the clock during transmission. SCPOL = 1, set to high level when in idle state. , when transmitting, the data is captured on the rising edge of the clock.

- The single frame transmission format is shown in the diagram *Motorola SPI single frame transmission format (SCPH = 1)*.

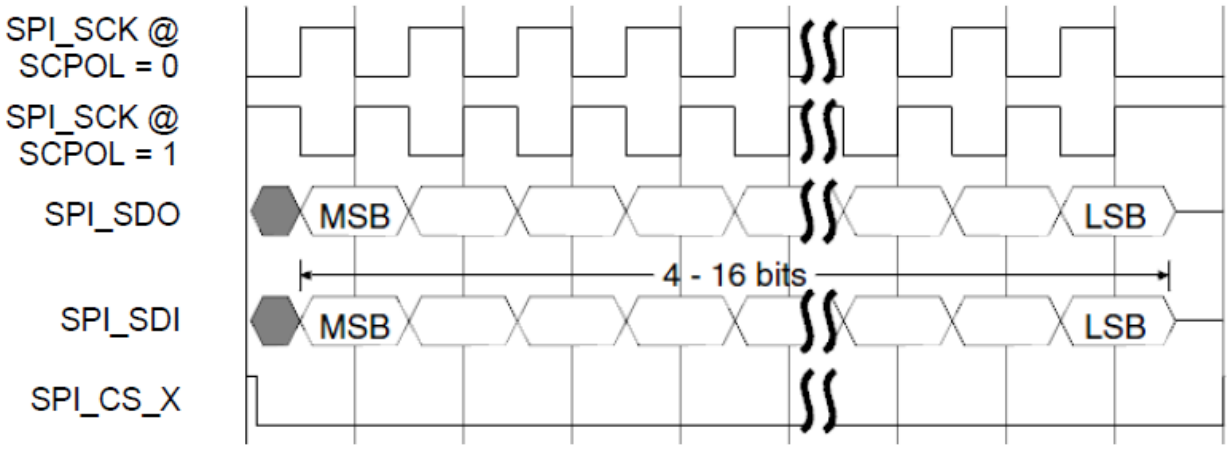


Diagram 21.14: Motorola SPI single frame transmission format (SCPH = 1)

- The continuous frame transmission format is shown in the diagram *Motorola SPI continuous frame transmission format (SCPH = 1)*.

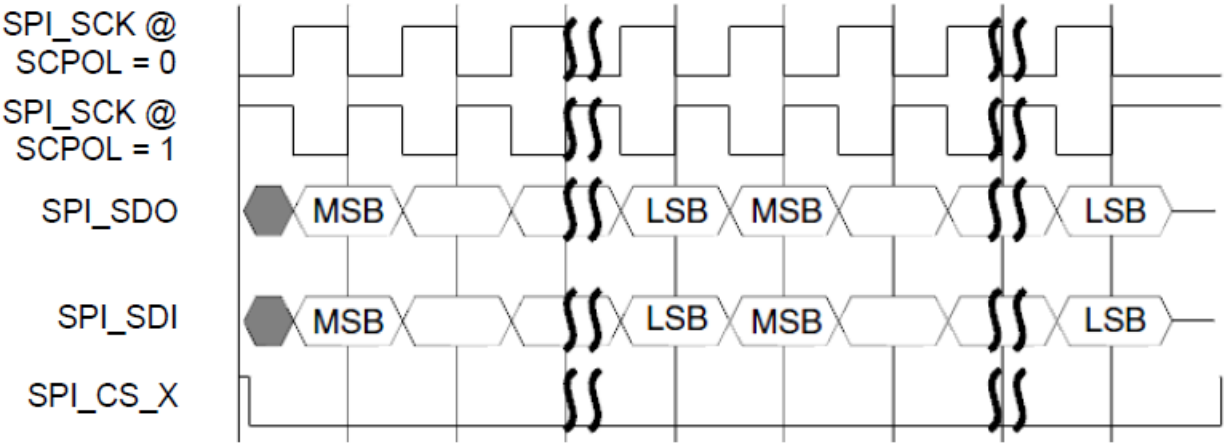


Diagram 21.15: Motorola SPI continuous frame transmission format (SCPH = 1)

21.3.5.2 TI Synchronous Serial Interface

In SSP mode, SPI_CS_X is set to high level when in idle state and set to low level during transmission. SPI_SCK is set to low level when in idle state, and data is captured on the falling edge of the clock during transmission.

The following figures represent the TI SSP data transfer format.

- The single frame transmission format is shown in the diagram *TI SSP single frame transmission format*.

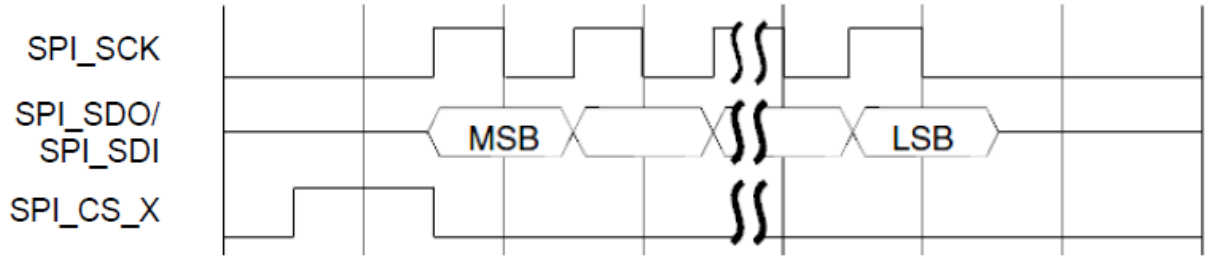


Diagram 21.16: TI SSP single frame transmission format

- The continuous frame transmission format is shown in the diagram *TI SSP continuous frame transmission format*.

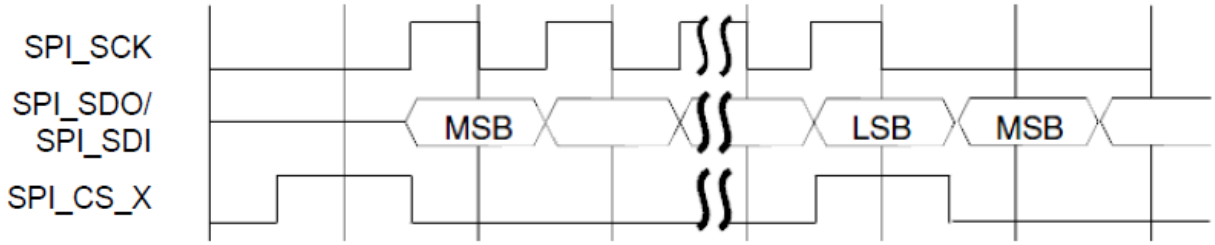


Diagram 21.17: TI SSP continuous frame transmission format

21.3.5.3 National Semiconductor Microwire Interface

In Microwire mode, SPI_CS_X is set to high level when in idle state and set to low level during transmission. SPI_SCK is set to low level when in idle state, and data is captured on the rising edge of the clock during transmission.

When transmitting data in this mode, a control word must be added first, and the external chip then responds to the data word required by the Master based on the control word. The control word length can be set through register CTRLR0[15:12], and other related parameters can be set through register MWCR.

The following figures represent the NS Microwire data transmission format.

- The single frame transmission format is shown in the diagram *NS Microwire Single frame transmission format*.

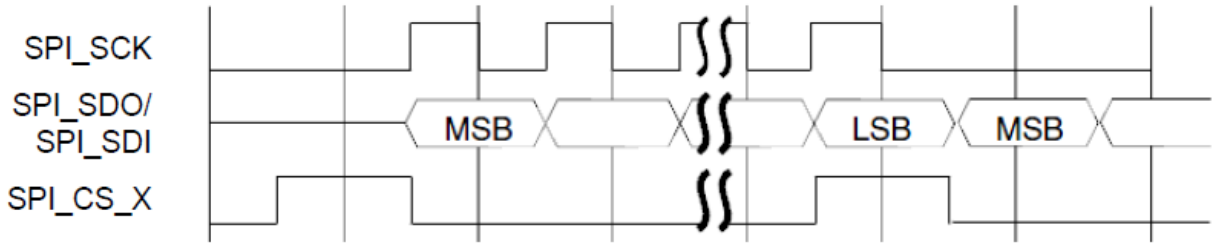


Diagram 21.18: NS Microwire Single frame transmission format

- The continuous frame transmission format is shown in the diagram *NS Microwire continuous frame transmission format*.

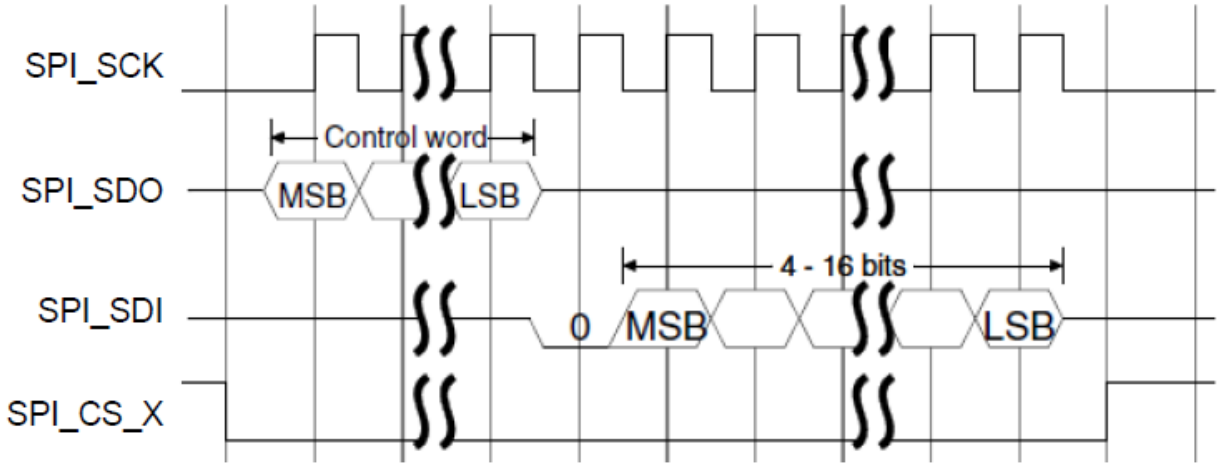


Diagram 21.19: NS Microwire continuous frame transmission format

21.3.6 Register Overview

The four sets of SPI module base addresses of the chip are shown in the table *Base addresses for 4 sets of SPI module*.

Table 21.71: Base addresses for 4 sets of SPI module

GPIO Module	Base Address
SPI0	0x04180000
SPI1	0x04190000
SPI2	0x041A0000
SPI3	0x041B0000

Table *Registes Overview (SPI0)* is the offset address and definition of the registers of the first group of SPI modules (SPI0). SPI0 ~ SPI3 have the same register definitions.

Table 21.72: Registes Overview (SPI0)

Name	Address Offset	Description
CTRLR0	0x000	Control Register 0
CTRLR1	0x004	Control Register 1
SPIENR	0x008	SPI Enable Register
MWCR	0x00c	Microwire Control Register
SER	0x010	Slave Enable Register
BAUDR	0x014	Baud Rate Select
TXFTLR	0x018	Transmit FIFO Threshold Level
RXFTLR	0x01c	Receive FIFO Threshold Level
TXFLR	0x020	Transmit FIFO Level Register
RXFLR	0x024	Receive FIFO Level Register
SR	0x028	Status Register
IMR	0x02c	Interrupt Mask Register
ISR	0x030	Interrupt Status Register
RISR	0x034	Raw Interrupt Status Register
TXOICR	0x038	Transmit FIFO Overflow Interrupt Clear Register
RXOICR	0x03c	Receive FIFO Overflow Interrupt Clear Register
RXUICR	0x040	Receive FIFO Underflow Interrupt Clear Register
MSTICR	0x044	Multi-Master Interrupt Clear Register
ICR	0x048	Interrupt Clear Register
DMACR	0x04c	DMA Control Register
DMATDLR	0x050	DMA Transmit Data Level
DMARDLR	0x054	DMA Receive Data Level
DR (36 groups)	0x060	Data Register
RX_SAMPLE_DLY	0x0f0	Rx Sample Delay Register

21.3.7 Register Description

21.3.7.1 CTRLR0

Table 21.73: CTRLR0, Offset Address: 0x000

Bits	Name	Access	Description	Reset
15:0	CTRLR0	R/W	[15:12] Control Frame Size. Selects the length of the control word for the Microwire frame format. • 0000 1-bit control word • 0001 2-bit control word • 0010 3-bit control word • 0011 4-bit control word • 0100 5-bit control word • 0101 6-bit control word • 0110 7-bit control word • 0111 8-bit control word • 1000 9-bit control word • 1001 10-bit control word • 1010 11-bit control word • 1011 12-bit control word • 1100 13-bit control word • 1101 14-bit control word • 1110 15-bit control word • 1111 16-bit control word [11] Shift Register Loop. Used for testing purposes only. When internally active, connects the transmit shift register output to the receive shift register input. Can be used in both serialslave and serial-master modes. • 0 - Normal Mode Operation • 1 - Test Mode Operation [10] only for slave mode.	0x7

To be continued ……

Table 21.74: CTRLR0, Offset Address: 0x000 (continued)

Bits	Name	Access	Description	Reset
			[9:8] Transfer Mode. Selects the mode of transfer for serial communication. This field does not affect the transfer duplicity. Only indicates whether the receive or transmit data are valid. In transmit-only mode, data received from the external device is not valid and is not stored in the receive FIFO memory; it is overwritten on the next transfer. In receive-only mode, transmitted data are not valid. After the first write to the transmit FIFO, the same word is retransmitted for the duration of the transfer. In transmit-and-receive mode, both transmit and receive data are valid. The transfer continues until the transmit FIFO is empty. Data received from the external device are stored into the receive FIFO memory, where it can be accessed by the host processor. In eeprom-read mode, receive data is not valid while control data is being transmitted. When all control data is sent to the EEPROM, receive data becomes valid and transmit data becomes invalid. All data in the transmit FIFO is considered control data in this mode. • 00 - Transmit & Receive • 01 - Transmit Only • 10 - Receive Only • 11 - EEPROM Read [7] Serial Clock Polarity. Valid when the frame format (FRF) is set to Motorola SPI. Used to select the polarity of the inactive serial clock, which is held inactive when the SPI master is not actively transferring data on the serial bus. • 0 - Inactive state of serial clock is low • 1 - Inactive state of serial clock is high	

To be continued ……

Table 21.75: CTRLR0, Offset Address: 0x000 (continued)

Bits	Name	Access	Description	Reset
			[6] Serial Clock Phase. Valid when the frame format (FRF) is set to Motorola SPI. The serial clock phase selects the relationship of the serial clock with the slave select signal. When SCPH = 0, data are captured on the first edge of the serial clock. When SCPH = 1, the serial clock starts toggling one cycle after the slave select line is activated, and data are captured on the second edge of the serial clock. • 0: Serial clock toggles in middle of first data bit • 1: Serial clock toggles at start of first data bit [5:4] Frame Format. Selects which serial protocol transfers the data. • 00 - Motorola SPI • 01 - Texas Instruments SSP • 10 - National Semiconductors Microwire • 11 - Reserved [3:0] Data Frame Size. Selects the data frame length. When the data frame size is programmed to be less than 16 bits, the receive data are automatically right-justified by the receive logic, with the upper bits of the receive FIFO zero-padded. You must right-justify transmit data before writing into the transmit FIFO. The transmit logic ignores the upper unused bits when transmitting the data. • 0000 Reserved - undefined operation • 0001 Reserved - undefined operation • 0010 Reserved - undefined operation • 0011 4-bit serial data transfer • 0100 5-bit serial data transfer • 0101 6-bit serial data transfer • 0110 7-bit serial data transfer • 0111 8-bit serial data transfer • 1000 9-bit serial data transfer • 1001 10-bit serial data transfer • 1010 11-bit serial data transfer • 1011 12-bit serial data transfer • 1100 13-bit serial data transfer • 1101 14-bit serial data transfer • 1110 15-bit serial data transfer • 1111 16-bit serial data transfer	
31:16	Reserved			

21.3.7.2 CTRLR1

Table 21.76: CTRLR1, Offset Address: 0x004

Bits	Name	Access	Description	Reset
15:0	CTRLR1	R/W	Number of Data Frames. When TMOD = 10 or TMOD = 11, this register field sets the number of data frames to be continuously received by the SPI. The SPI continues to receive serial data until the number of data frames received is equal to this register value plus 1, which enables you to receive up to 64 KB of data in a continuous transfer.	0x0
31:16	Reserved			

21.3.7.3 SPIENR

Table 21.77: SPIENR, Offset Address: 0x008

Bits	Name	Access	Description	Reset
0	SPIENR	R/W	SPI Enable. Enables and disables all SPI operations. When disabled, all serial transfers are halted immediately. Transmit and receive FIFO buffers are cleared when the device is disabled. It is impossible to program some of the SPI control registers when enabled. When disabled, the spi_sleep output is set (after delay) to inform the system that it is safe to remove the spi_clk, thus saving power consumption in the system.	0x0
31:1	Reserved			

21.3.7.4 MWCR

Table 21.78: MWCR, Offset Address: 0x00c

Bits	Name	Access	Description	Reset
2:0	MWCR	R/W	[2] Microwire Handshaking. Relevant only when the SPI is configured as a serial-master device. When configured as a serial slave, this bit field has no functionality. Used to enable and disable the “busy/ready” handshaking interface for the Microwire protocol. When enabled, the SPI checks for a ready status from the target slave, after the transfer of the last data/control bit, before clearing the BUSY status in the SR register. 0: handshaking interface is disabled 1: handshaking interface is enabled [1] Microwire Control. Defines the direction of the data word when the Microwire serial protocol is used. When this bit is set to 0, the data word is received by the SPI from the external serial device. When this bit is set to 1, the data word is transmitted from the SPI to the external serial device. [0] Microwire Transfer Mode. Defines whether the Microwire transfer is sequential or non-sequential. When sequential mode is used, only one control word is needed to transmit or receive a block of data words. When non-sequential mode is used, there must be a control word for each data word that is transmitted or received. 0 – non-sequential transfer 1 – sequential transfer	0x0
31:3	Reserved			

21.3.7.5 SER

Table 21.79: SER, Offset Address: 0x010

Bits	Name	Access	Description	Reset
0	SER	R/W	Slave Select Enable Flag. This register corresponds to a slave select line (ss_x_n]) from the SPI master. When this register is set (1), the slave select line from the master is activated when a serial transfer begins. It should be noted that setting or clearing this register have no effect on the corresponding slave select outputs until a transfer is started. Before beginning a transfer, you should enable this register that corresponds to the slave device with which the master wants to communicate. 1: Selected 0: Not Selected	0x0
31:1	Reserved			

21.3.7.6 BAUDR

Table 21.80: BAUDR, Offset Address: 0x014

Bits	Name	Access	Description	Reset
15:0	BAUDR	R/W	SPI Clock Divider(SCKDV). The LSB for this field is always set to 0 and is unaffected by a write operation, which ensures an even value is held in this register. If the value is 0, the serial output clock (sclk_out) is disabled. The frequency of the sclk_out is derived from the following equation: Fsclk_out/SCKDV = Fssi_clk where SCKDV is any even value between 2 and 65534. For example: for Fssi_clk = 3.6864MHz and SCKDV =2 Fsclk_out = 3.6864/2 = 1.8432MHz	0x0
31:16	Reserved			

21.3.7.7 TXFTLR

Table 21.81: TXFTLR, Offset Address: 0x018

Bits	Name	Access	Description	Reset
2:0	TXFTLR	R/W	Transmit FIFO Threshold. Controls the level of entries (or below) at which the transmit FIFO controller triggers an interrupt. The FIFO depth is 8; If you attempt to set this value greater than or equal to the depth of the FIFO, this field is not written and retains its current value. When the number of transmit FIFO entries is less than or equal to this value, the transmit FIFO empty interrupt is triggered.	0x0
31:3	Reserved			

21.3.7.8 RXFTLR

Table 21.82: RXFTLR, Offset Address: 0x01c

Bits	Name	Access	Description	Reset
2:0	RXFTLR	R/W	Receive FIFO Threshold. Controls the level of entries (or above) at which the receive FIFO controller triggers an interrupt. The FIFO depth is 8. If you attempt to set this value greater than the depth of the FIFO, this field is not written and retains its current value. When the number of receive FIFO entries is greater than or equal to this value + 1, the receive FIFO full interrupt is triggered.	0x0
31:3	Reserved			

21.3.7.9 TXFLR

Table 21.83: TXFLR, Offset Address: 0x020

Bits	Name	Access	Description	Reset
3:0	TXFLR	RO	Transmit FIFO Level. Contains the number of valid data entries in the transmit FIFO.	
31:4	Reserved			

21.3.7.10 RXFLR

Table 21.84: RXFLR, Offset Address: 0x024

Bits	Name	Access	Description	Reset
3:0	RXFLR	RO	Receive FIFO Level. Contains the number of valid data entries in the receive FIFO.	
31:4	Reserved			

21.3.7.11 SR

Table 21.85: SR, Offset Address: 0x028

Bits	Name	Access	Description	Reset
6:0	SR	RO	[6] Data Collision Error. This bit is set if the SPI master is actively transmitting when another master selects this device as a slave. This informs the processor that the last data transfer was halted before completion. This bit is cleared when read. 0 - No error 1 - Transmit data collision error [5] Transmission Error. Set if the transmit FIFO is empty when a transfer is started. Data from the previous transmission is resent on the txd line. This bit is cleared when read. 0 - No error 1 - Transmission error [4] Receive FIFO Full. When the receive FIFO is completely full, this bit is set. When the receive FIFO contains one or more empty location, this bit is cleared. 0 - Receive FIFO is not full 1 - Receive FIFO is full [3] Receive FIFO Not Empty. Set when the receive FIFO contains one or more entries and is cleared when the receive FIFO is empty. This bit can be polled by software to completely empty the receive FIFO. 0 - Receive FIFO is empty 1 - Receive FIFO is not empty [2] Transmit FIFO Empty. When the transmit FIFO is completely empty, this bit is set. When the transmit FIFO contains one or more valid entries, this bit is cleared. This bit field does not request an interrupt. 0 - Transmit FIFO is not empty 1 - Transmit FIFO is empty [1] Transmit FIFO Not Full. Set when the transmit FIFO contains one or more empty locations, and is cleared when the FIFO is full. 0 - Transmit FIFO is full 1 - Transmit FIFO is not full [0] SPI Busy Flag. When set, indicates that a serial transfer is in progress; when cleared indicates that the SPI is idle or disabled. 0 - SPI is idle or disabled 1 - SPI is actively transferring data	
31:7	Reserved			

21.3.7.12 IMR

Table 21.86: IMR, Offset Address: 0x02c

Bits	Name	Access	Description	Reset
5:0	IMR	R/W	[5] Multi-Master Contention Interrupt Mask. 0 - spi_mst_intr interrupt is masked 1 - spi_mst_intr interrupt is not masked [4] Receive FIFO Full Interrupt Mask 0 - spi_rxf_intr interrupt is masked 1 - spi_rxf_intr interrupt is not masked [3] Receive FIFO Overflow Interrupt Mask 0 - spi_rxo_intr interrupt is masked 1 - spi_rxo_intr interrupt is not masked [2] Receive FIFO Underflow Interrupt Mask 0 - spi_rxu_intr interrupt is masked 1 - spi_rxu_intr interrupt is not masked [1] Transmit FIFO Overflow Interrupt Mask 0 - spi_txo_intr interrupt is masked 1 - spi_txo_intr interrupt is not masked [0] Transmit FIFO Empty Interrupt Mask 0 - spi_txe_intr interrupt is masked 1 - spi_txe_intr interrupt is not masked	0x3F
31:6	Reserved			

21.3.7.13 ISR

Table 21.87: ISR, Offset Address: 0x030

Bits	Name	Access	Description	Reset
5:0	ISR	RO	[5] Multi-Master Contention Interrupt Status. 0 = spi_mst_intr interrupt not active after masking 1 = spi_mst_intr interrupt is active after masking [4] Receive FIFO Full Interrupt Status 0 = spi_rxf_intr interrupt is not active after masking 1 = spi_rxf_intr interrupt is full after masking [3] Receive FIFO Overflow Interrupt Status 0 = spi_rxo_intr interrupt is not active after masking 1 = spi_rxo_intr interrupt is active after masking [2] Receive FIFO Underflow Interrupt Status 0 = spi_rxu_intr interrupt is not active after masking 1 = spi_rxu_intr interrupt is active after masking [1] Transmit FIFO Overflow Interrupt Status 0 = spi_txo_intr interrupt is not active after masking 1 = spi_txo_intr interrupt is active after masking [0] Transmit FIFO Empty Interrupt Status 0 = spi_txe_intr interrupt is not active after masking 1 = spi_txe_intr interrupt is active after masking	
31:6	Reserved			

21.3.7.14 RISR

Table 21.88: RISR, Offset Address: 0x034

Bits	Name	Access	Description	Reset
5:0	RISR	RO	[5] Multi-Master Contention Raw Interrupt Status. 0 = spi_mst_intr interrupt is not active prior to masking 1 = spi_mst_intr interrupt is active prior masking [4] Receive FIFO Full Raw Interrupt Status 0 = spi_rxf_intr interrupt is not active prior to masking 1 = spi_rxf_intr interrupt is active prior to masking [3] Receive FIFO Overflow Raw Interrupt Status 0 = spi_rxo_intr interrupt is not active prior to masking 1 = spi_rxo_intr interrupt is active prior masking [2] Receive FIFO Underflow Raw Interrupt Status 0 = spi_rxu_intr interrupt is not active prior to masking 1 = spi_rxu_intr interrupt is active prior to masking [1] Transmit FIFO Overflow Raw Interrupt Status 0 = spi_txo_intr interrupt is not active prior to masking 1 = spi_txo_intr interrupt is active prior masking [0] Transmit FIFO Empty Raw Interrupt Status 0 = spi_txe_intr interrupt is not active prior to masking 1 = spi_txe_intr interrupt is active prior masking	
31:6	Reserved			

21.3.7.15 TXOICR

Table 21.89: TXOICR, Offset Address: 0x038

Bits	Name	Access	Description	Reset
0	TXOICR	RO	Clear Transmit FIFO Overflow Interrupt. This register reflects the status of the interrupt. A read from this register clears the spi_txo_intr interrupt; writing has no effect.	
31:1	Reserved			

21.3.7.16 RXOICR

Table 21.90: RXOICR, Offset Address: 0x03c

Bits	Name	Access	Description	Reset
0	RXOICR	RO	Clear Receive FIFO Overflow Interrupt. This register reflects the status of the interrupt. A read from this register clears the spi_rxo_intr interrupt; writing has no effect.	
31:1	Reserved			

21.3.7.17 RXUICR

Table 21.91: RXUICR, Offset Address: 0x040

Bits	Name	Access	Description	Reset
0	RXUICR	RO	Clear Receive FIFO Underflow Interrupt. This register reflects the status of the interrupt. A read from this register clears the spi_rxu_intr interrupt; writing has no effect.	
31:1	Reserved			

21.3.7.18 MSTICR

Table 21.92: MSTICR, Offset Address: 0x044

Bits	Name	Access	Description	Reset
0	MSTICR	RO	Clear Multi-Master Contention Interrupt. This register reflects the status of the interrupt. A read from this register clears the spi_mst_intr interrupt; writing has no effect.	
31:1	Reserved			

21.3.7.19 ICR

Table 21.93: ICR, Offset Address: 0x048

Bits	Name	Access	Description	Reset
0	ICR	RO	Clear Interrupts. This register is set if any of the interrupts below are active. A read clears the spi_txo_intr, spi_rxu_intr, spi_rxo_intr, and the spi_mst_intr interrupts. Writing to this register has no effect.	
31:1	Reserved			

21.3.7.20 DMACR

Table 21.94: DMACR, Offset Address: 0x04c

Bits	Name	Access	Description	Reset
1:0	DMACR	R/W	[1] Transmit DMA Enable. This bit enables/disables the transmit FIFO DMA channel. 0 = Transmit DMA disabled 1 = Transmit DMA enabled [0] Receive DMA Enable. This bit enables/disables the receive FIFO DMA channel 0 = Receive DMA disabled 1 = Receive DMA enabled	0x0
31:2	Reserved			

21.3.7.21 DMATDLR

Table 21.95: DMATDLR, Offset Address: 0x050

Bits	Name	Access	Description	Reset
2:0	DMATDLR	R/W	Transmit Data Level. This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of valid data entries in the transmit FIFO is equal to or below this field value, and TDMAE = 1.	0x0
31:3	Reserved			

21.3.7.22 DMARDLR

Table 21.96: DMARDLR, Offset Address: 0x054

Bits	Name	Access	Description	Reset
2:0	DMARDLR	R/W	Receive Data Level. This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or above this field value + 1, and RDMAE=1.	0x0
31:3	Reserved			

21.3.7.23 DR

Table 21.97: DR, Offset Address: 0x060

Bits	Name	Access	Description	Reset
15:0	DR	R/W	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. Read = Receive FIFO buffer Write = Transmit FIFO buffer Note : The DR register in the SPI occupies thirty-six 32-bit address locations of the memory map to facilitate AHB burst transfers. Writing to any of these address locations has the same effect as pushing the data from the pwdata bus into the transmit FIFO. Reading from any of these locations has the same effect as popping data from the receive FIFO onto the prdata bus.	0x0
31:16	Reserved			

21.3.7.24 RX_SAMPLE_DLY

Table 21.98: RX_SAMPLE_DLY, Offset Address: 0x0f0

Bits	Name	Access	Description	Reset
7:0	RX_SAMPLE_DLY	R/W	Receive Data (rxd) Sample Delay. This register is used to delay the sample of the rxd input signal. Each value represents a single ssi_clk delay on the sample of the rxd signal. NOTE: If this register is programmed with a value that exceeds the depth of the internal shift registers (DEPTH = 8), a zero (0) delay will be applied to the rxd sample.	0x0
31:8	Reserved			

21.4 eMMC/SD/SDIO Controller

21.4.1 Function Description

21.4.1.1 Functional Block Diagram

The EMMC/SD/SDIO controller (SDMMC controller for short) is used to handle operations such as data reading and writing of SD cards and eMMC, as well as external devices supported by the SDIO protocol (such as Bluetooth, WIFI, etc.). This chip provides three sets of SDMMC controllers. in:

- EMMC supports devices that comply with the eMMC4.1 and eMMC4.5 protocols.
- SDIO0 supports devices that comply with the Secure Digital Memory (SD 3.0) protocol.
- SDIO1 supports devices that comply with the Secure Digital I/O (SDIO 3.0) protocol.

The corresponding functional signals and pins of the three SDMMC controllers in the chip are as shown in the table below.

Table 21.99: Corresponding function signals and pin mapping table of SDMMC controller

SDMMC Controller	Functional Signal	Pin
eMMC	EMMC_CLK	EMMC_CLK
	EMMC_CMD	EMMC_CMD
	EMMC_DATA0	EMMC_DAT0
	EMMC_DATA1	EMMC_DAT1
	EMMC_DATA2	EMMC_DAT2
	EMMC_DATA3	EMMC_DAT3
	EMMC_RSTn	EMMC_RSTN
SDIO0	SD_CLK	SD0_CLK
	SD_CMD	SD0_CMD
	SD_DATA0	SD0_D0
	SD_DATA1	SD0_D1
	SD_DATA2	SD0_D2
	SD_DATA3	SD0_D3
	SD_CARD_DETECT	SD0_CD
	SD_POWER_EN	SD0_PWR_EN
SDIO1	SDIO_CLK	SD1_CLK
	SDIO_CMD	SD1_CMD
	SDIO_DATA0	SD1_D0
	SDIO_DATA1	SD1_D1
	SDIO_DATA2	SD1_D2
	SDIO_DATA3	SD1_D3

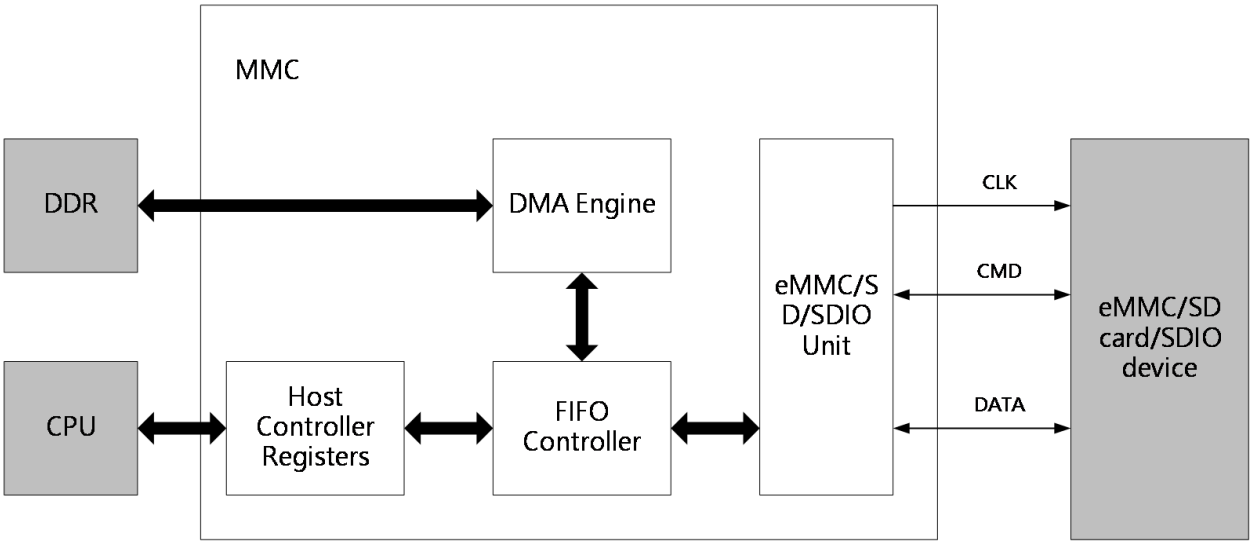


Diagram 21.20: SDMMC controller functional block diagram

SDMMC functions:

1. Support SD card, SDIO device and eMMC.
2. Transfer data between eMMC/SD/SDIO and system memory data through the internal DMA controller.
3. Supports CRC generation and checking of commands and data.
4. The frequency required between different modes can be generated through the internal frequency divider.
5. Provide a mechanism to turn off the internal clock and the clock on the interface to meet power saving requirements.
6. Provides 1-bit and 4-bit data transmission interfaces to communicate with devices.
7. Supports data read and write operations with block_size equal to 1~2048byte.
8. Support SDIO protocol, including interrupt interval, suspend, resume and read wait operations.
9. Supports AXI/AHB interface and can access system memory through internal DMA.
10. Supports AHB interface, which can access internal registers through CPU.

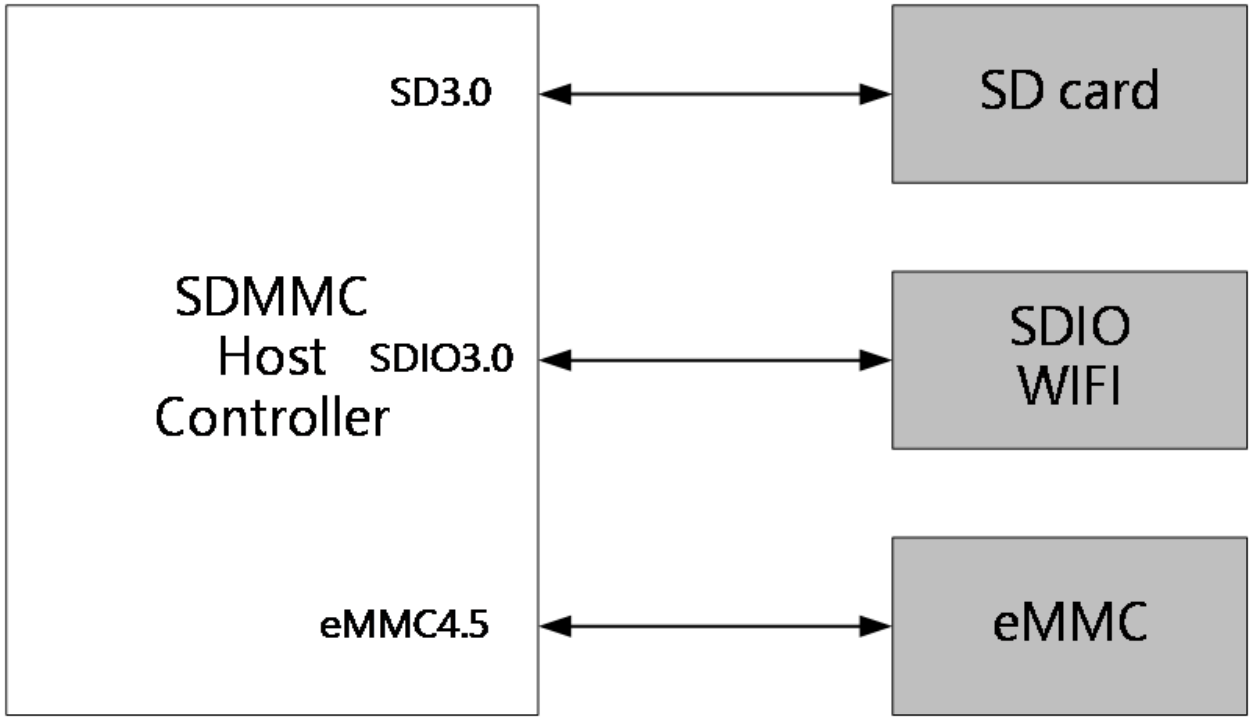


Diagram 21.21: typical application

21.4.1.2 Command and Response

The bus packet of eMMC/SD mainly consists of three parts: command, response and data.

The command and response packets are transmitted through the CMD signal line.

- Command packet

  The command packet is sent from the host to the device to indicate the start of an operation. The packet format consists of 48 bits including start bit, transmission bit, command number, command parameter, CRC verification code and end bit. As shown in *eMMC/SD/SDIO command format*.

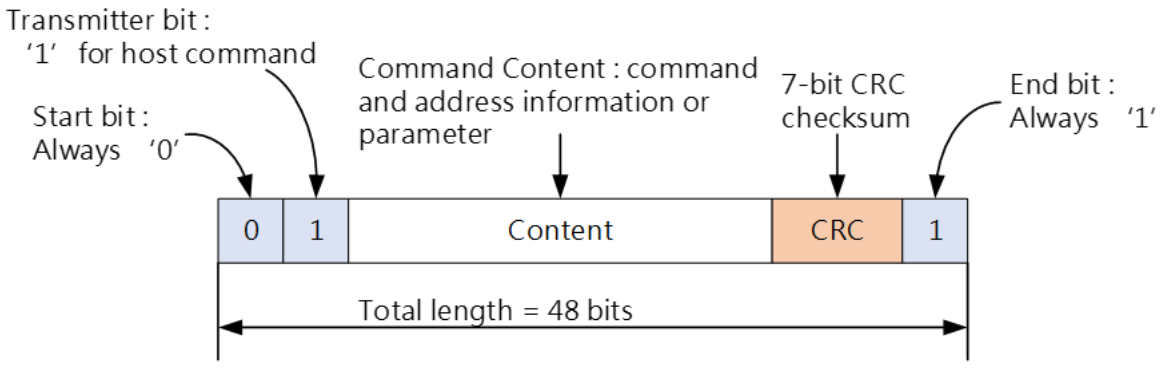


Diagram 21.22: eMMC/SD/SDIO command format

- Response Packet

  After receiving the command, the device will return a response according to different command categories to

display the status or parameters of the device. Its length is 48 bits or 136 bits. As shown in *eMMC/SD/SDIO response format*.

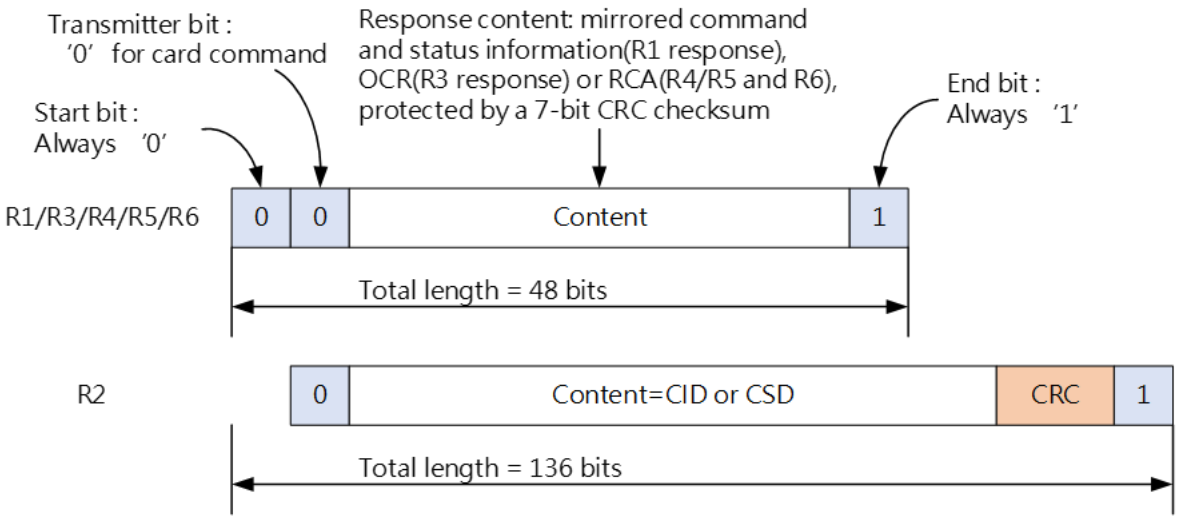


Diagram 21.23: eMMC/SD/SDIO response format

- Data packet:

  Data packets are used to exchange data between the host and the device. According to different needs, 1-bit (DATA0), 4-bit (DATA0-DATA3) or 7-bit (DATA0-DATA7) can be selected. In each clock interval, Each data signal line can choose to transmit (single data rate) or (dual data rate). The packet formats are shown in the diagram *eMMC/SD/SDIO 1-bit data packet format* ~ the diagram *8-bit dual data rate data packet format*.

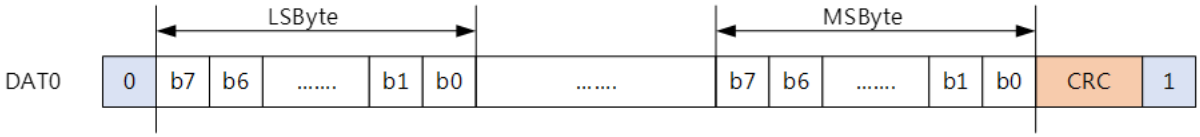


Diagram 21.24: eMMC/SD/SDIO 1-bit data packet format

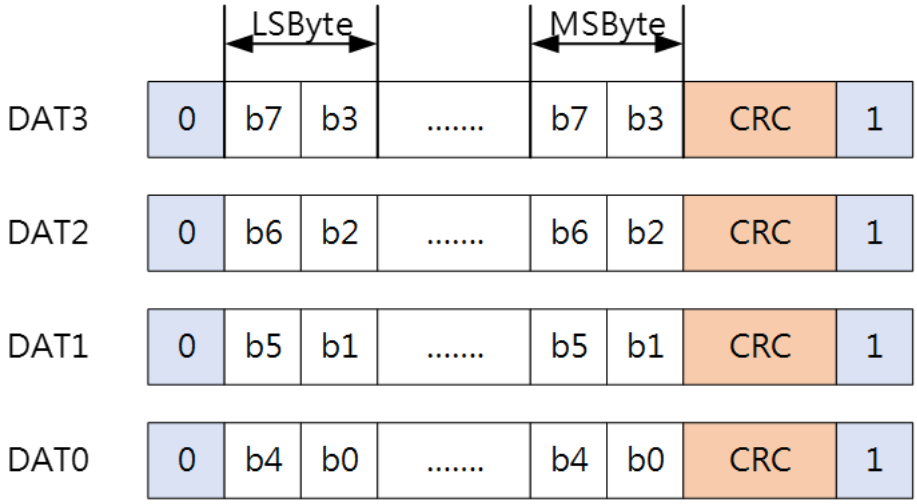


Diagram 21.25: eMMC/SD/SDIO 4-bit data packet format

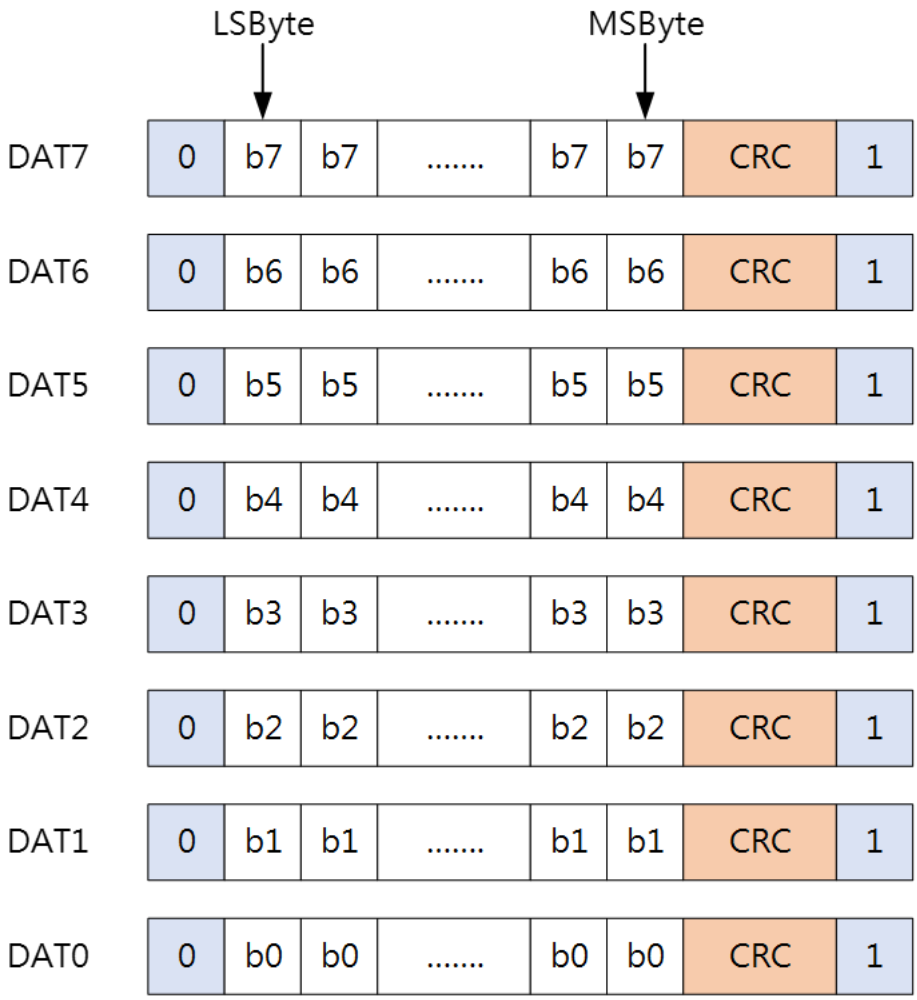


Diagram 21.26: eMMC/SD/SDIO 8-bit data packet format

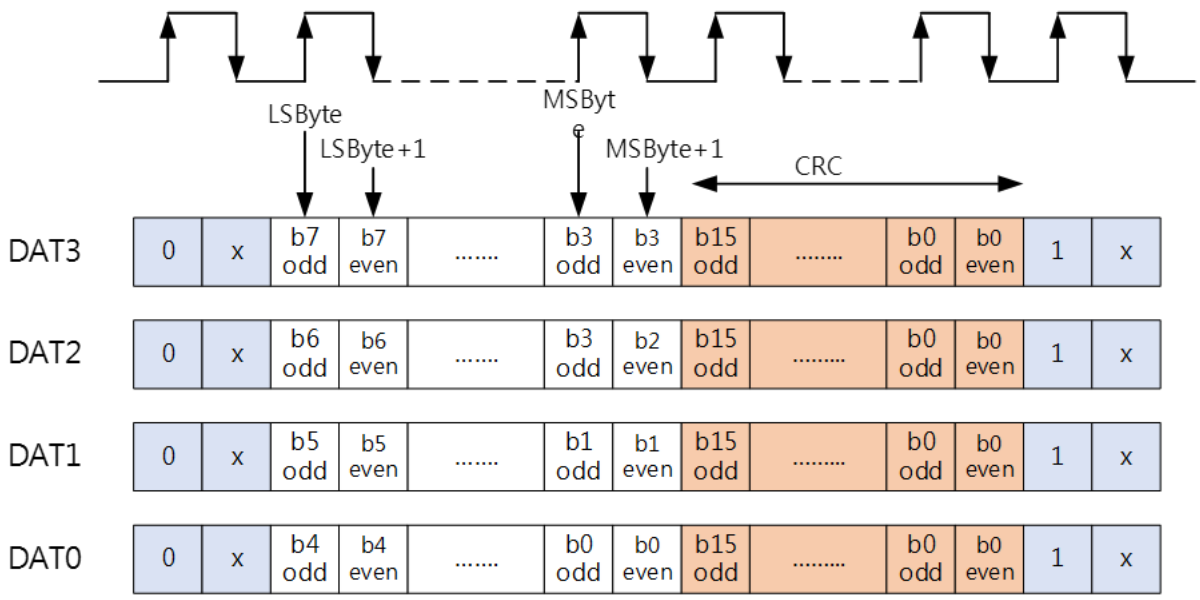


Diagram 21.27: 4-bit dual data rate data packet format

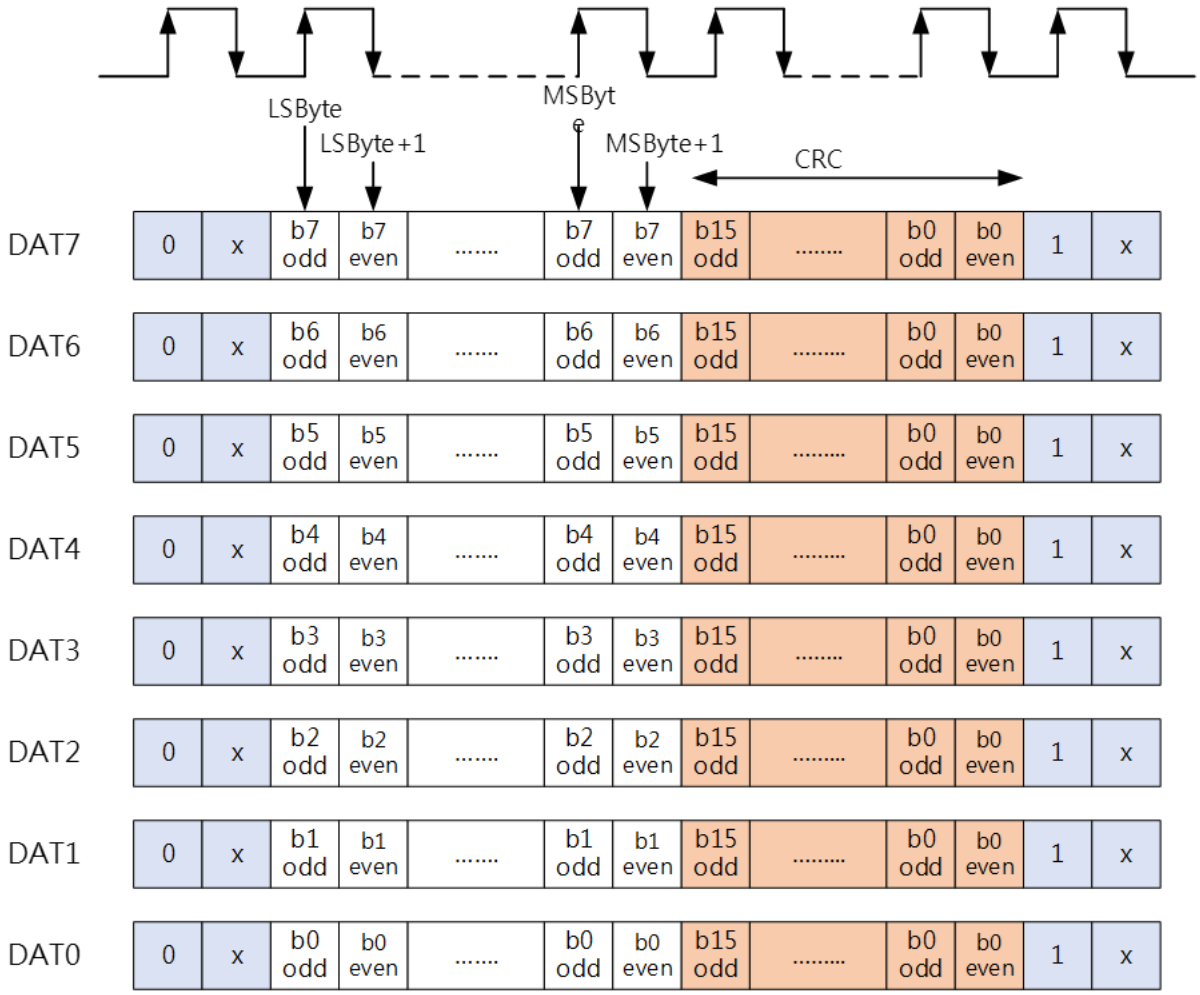


Diagram 21.28: 8-bit dual data rate data packet format

Depending on whether there is data transmission, instructions can be further divided into the following two types:

- Non-data transmission commands: Complete command transmission and receive responses through the signal line CMD.

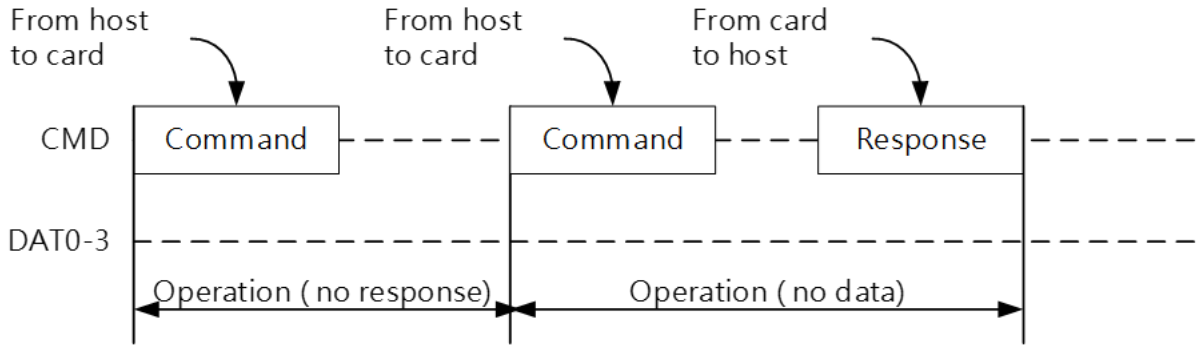


Diagram 21.29: Non-data transmission instructions: Complete instruction transmission and receive responses through the signal line CMD

- Data transmission instructions: In addition to the interaction on the signal line CMD, there is also data transmission on the data lines DAT0~DAT3

21.4.1.3 Data Transmission

The data transmission between the host and the device is mainly in blocks. In addition to the data, CRC check bits are also included to verify the correctness of the data. The more commonly used methods are single-block data reading and writing and multi-block data reading and writing. Compared with single-block data transmission, multi-block data transmission has higher efficiency. Among them, the block size of EMMC and SD card is 512byte. SDIO is special and can support block sizes of 1~2048byte. Users can define the block size value according to different devices.

(1) Single block and multi-block read operations are shown in the diagram *Single block and multi-block read operations*. Single block transmission consists of instructions, responses, data and CRC. Multi-block transfers end with a reliable STOP CMD to abort the transfer.

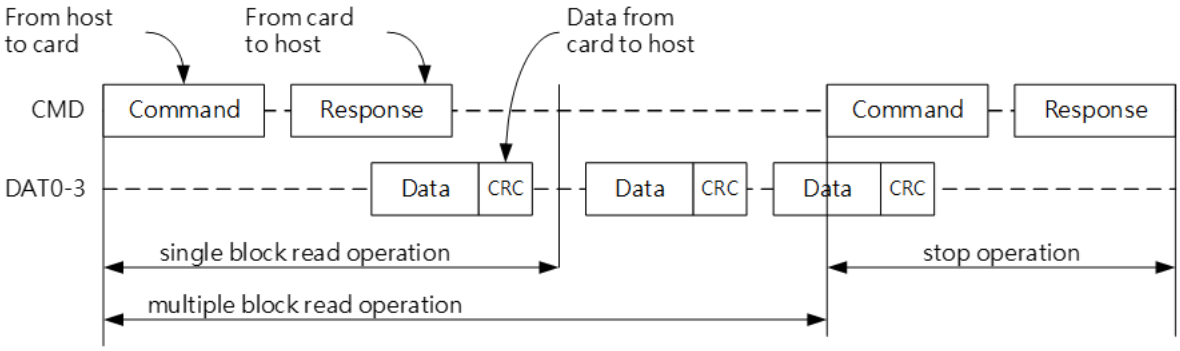


Diagram 21.30: Single block and multi-block read operations

(2) Single block and multi-block write operations are as shown in the diagram *Single block and multi-block write operations*. The transmission process will send a BUSY signal through the DAT0 signal line to notify the host that the writing device is in progress.

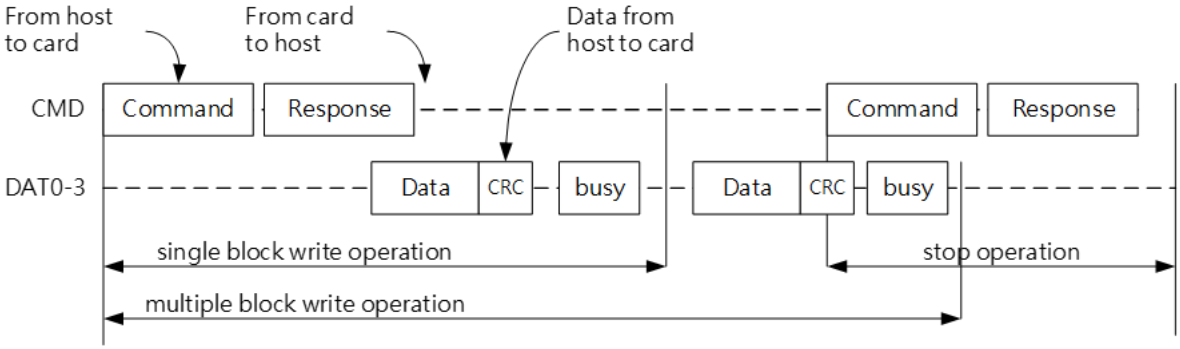


Diagram 21.31: Single block and multi-block write operations

21.4.1.4 Speed mode and voltage switching supported by SD3.0

- Voltage switching procedure (1.8V -> 3.3V)
  - Step 1: Set PWRSW to 3.3V. => sd_pwrsw_ctrl (0x030001F4) = 0x00000009 (reg_pwrsw_auto=1, reg_pwrsw_disc=0, reg_pwrsw_vsel=0(3.0v), reg_en_pwrsw=1)
  - Step 2: Wait 1ms to complete voltage switching.
- Voltage switching procedure (3.3V -> 1.8V)
  - Step 1: Set PWRSW to 1.8V. => sd_pwrsw_ctrl (0x030001F4) = 0x0000000B (reg_pwrsw_auto=1, reg_pwrsw_disc=0, reg_pwrsw_vsel=1(1.8v), reg_en_pwrsw=1)
  - Step 2: Wait 1ms to complete voltage switching.
- Support speed mode and voltage

  The speed modes and voltages supported by SD3.0 are as follows.

Table 21.100: SD3.0 supported speeds and voltages

Mode	Speed	Voltage
DS (default speed)	25Mhz	1.8V/3.3V
HS (high speed)	50Mhz	1.8V/3.3V
SDR12	25Mhz	1.8V
SDR25	50Mhz	1.8V
DRR50	50Mhz	1.8V
SDR50	100Mhz	1.8V
SDR104	187.5Mhz	1.8V

21.4.1.5 eMMC supported Speed Modes and Voltages

The speed modes and voltages supported by eMMC are as follows.

Table 21.101: eMMC supported speeds and voltages

Mode	Speed	Voltage
DS (default speed)	26Mhz	1.8V/3.3V
HS (high speed)	52Mhz	1.8V/3.3V
DRR52	52Mhz	1.8V
HS200	187.5Mhz	1.8V

21.4.2 Application Notes

21.4.2.1 Clock Shutdown

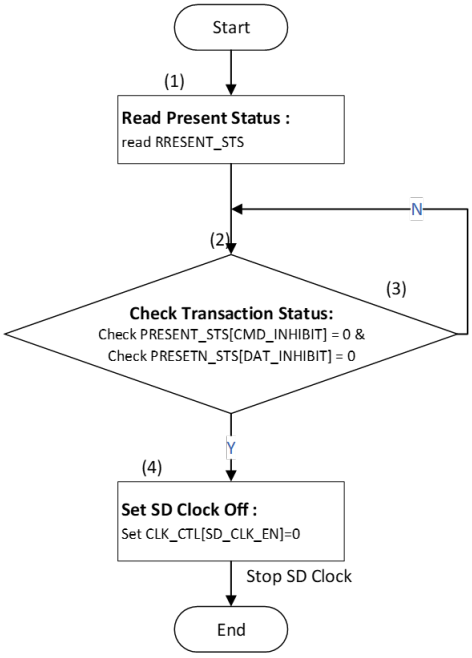


Diagram 21.32: clock shutdown program

As shown in the diagram *clock shutdown program* is a clock shutdown procedure. The host must ensure that no transmission is taking place on the bus in order to shut down the clock.

(1) Read the temporary register PRESENT_STS.

(2) Check whether the bits CMD_INHIBIT and DAT_INHIBIT are both 0.

(3) If any bit is not 0, it means that the transmission is still in progress and a delay is required.

(4) If both are 0, you can set CLK_CTL[SD_CLK_EN]=0 to turn off the clock.

21.4.2.2 Soft Reset

When the controller operates abnormally, reliably reset the configuration register (base address = 0x0300_3000) to perform a soft reset. The temporary register addresses used are as follows:

1. EMMC: SOFT_RSTN_0[reg_soft_reset_x_emmc] (address offset: 0x000, Bit15)
2. SDIO0: SOFT_RSTN_0[reg_soft_reset_x_sd0] (address offset: 0x000, Bit16)
3. SDIO1: SOFT_RSTN_0[reg_soft_reset_x_sd1] (address offset: 0x000, Bit17)

21.4.2.3 Interface Clock Configuration

Diagram *Clock configuration flow chart* is a flow chart for interface clock configuration. The SDMMC controller provides a frequency divider inside, allowing users to adjust the required clock frequency according to different protocols and speed modes. Its relationship is:

$F_{SD\_CLK\_OUT} = F_{INT\_CARD\_CLK} / (2 * clk\_divisor)$

When SDMMC changes the frequency, in addition to ensuring that no instructions and data are still being transmitted, it must also be set according to the steps of the interface clock configuration flow chart to avoid glitches in the clock output to the eMMC/SD device.

(1) Turn off the interface clock.

(2) Calculate the frequency division factor.

(3) Set the frequency division factor. Fill in the parameters calculated in (2) into CLK_CTL[FREQ_SEL], and start turning on the internal clock switch (CLK_CTL[INT_CLK_EN]=1).

(4) Check CLK_CTL[INT_CLK_STABLE] to confirm whether the frequency switching is completed.

(5) If it has not been completed (CLK_CTL[INT_CLK_STABLE]=0), delay and wait.

(6) If switching the clock frequency is completed, turn on the interface clock.

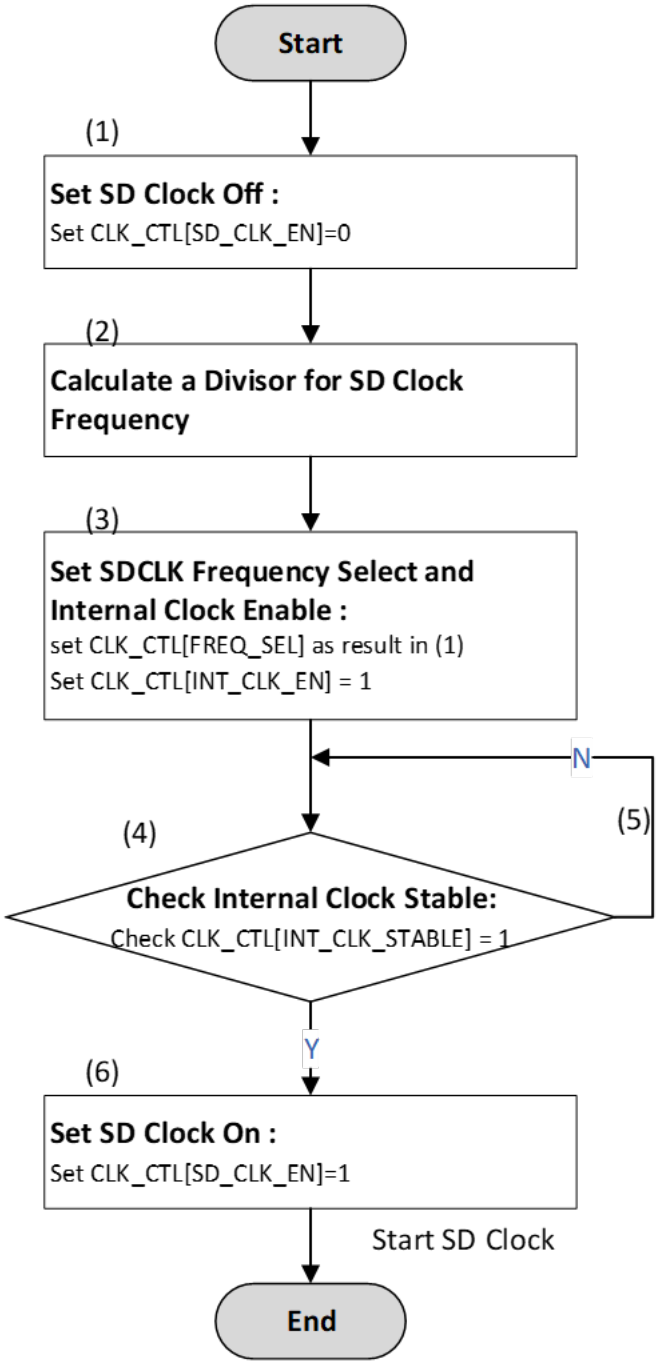


Diagram 21.33: Clock configuration flow chart

21.4.2.4 Non-Data Transfer Instructions

- Command transmission program

  The command transmission program is as shown in the diagram *command transfer program*.

  (1) First, you must check whether the register bit PRESENT[CMD_INHIBIT] is 0 to confirm whether the CMD Line is still in use.

  (2) If the CMD Line is idle, further confirm whether it is a command with busy. If it is not a busy instruction, you do not need to check the status of the DATA Line and directly execute step (5). Otherwise, if it is a busy instruction, you must execute step (3) to confirm whether it is an Abort instruction.

  (3) If it is an Abort command, it means that when the CMD line completes the transmission, the DATA line is also idle, and you can directly enter step (5); otherwise, if it is not an Abort command, you must execute step (4) to confirm that the DATA line is busy Whether it has been released.

  (4) Check whether the temporary register bit PRESENT[DAT_INHIBIT] is 0 to confirm whether the DATA Line is still in use. If it is still in use, wait until the transmission is completed, and then perform step (5).

  (5) Set the values of the ARGUMENT register and CMD register according to the instruction requirements.

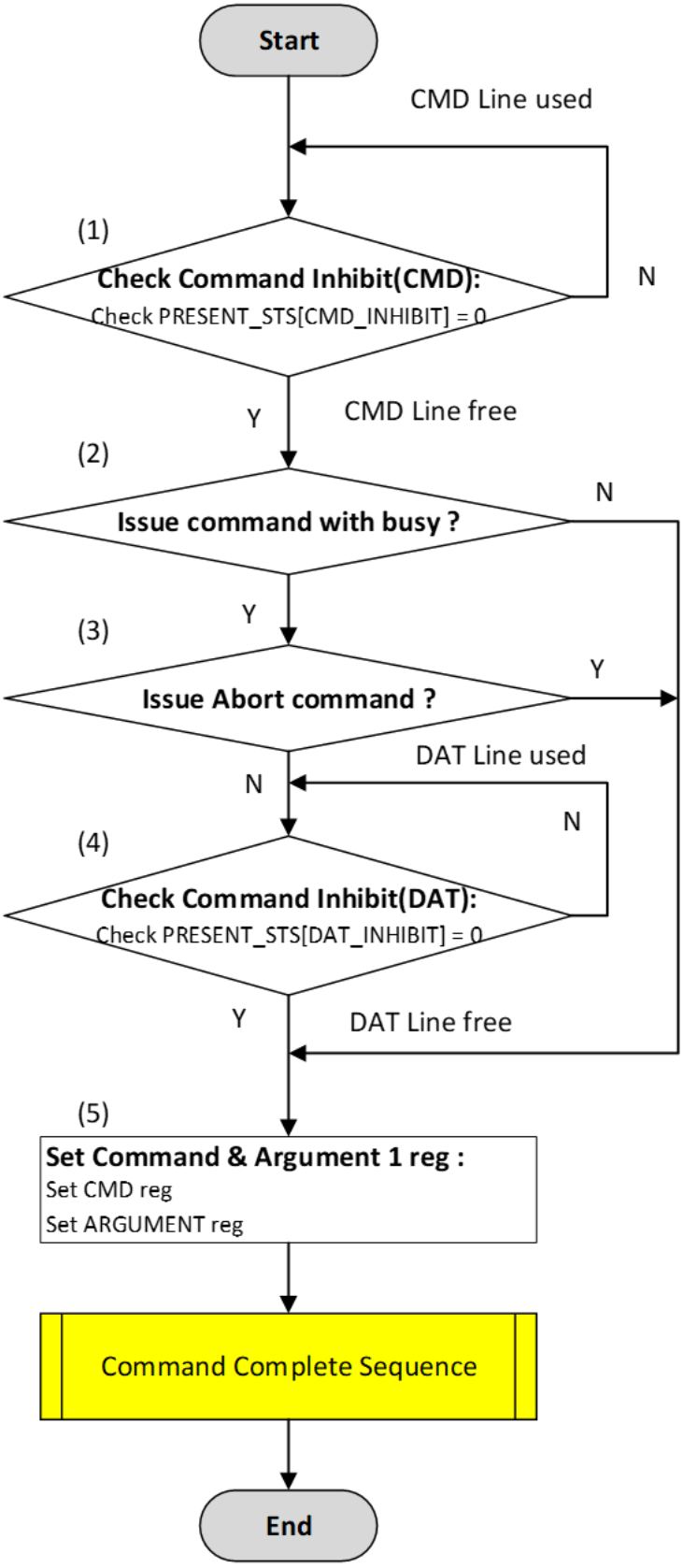


Diagram 21.34: command transfer program

- Command Complete Sequence

  The instruction completion procedure is as shown in the diagram *command completion program*.

  (1) First, wait for the interrupt NORM_INT_STS[CMD_CMPL] for Command completion.

  (2) After receiving the interrupt, set NORM_INT_STS[CMD_CMPL]=1 to clear the CMD_CMPL interrupt status.

  (3) Then read the RESP1_0, RESP3_2, RESP5_4, RESP7_6 and other temporary registers to obtain the response value.

  (4) If it is an instruction including data transfer, step (5) will be executed, otherwise jump to step (8).

  (5) Wait for data transfer interrupt NORM_INT_STS[XFER_CMPL].

  (6) After receiving the interrupt, set NORM_INT_STS[XFER_CMPL]=1 to clear the XFER_CMPL interrupt status.

  (7) Check RESP1_0, RESP3_2, RESP5_4, RESP7_6 and other temporary registers to confirm whether there is an error status. If there is no error status, perform step (8) and report no error. If there is an error, perform step (9) to report the error.

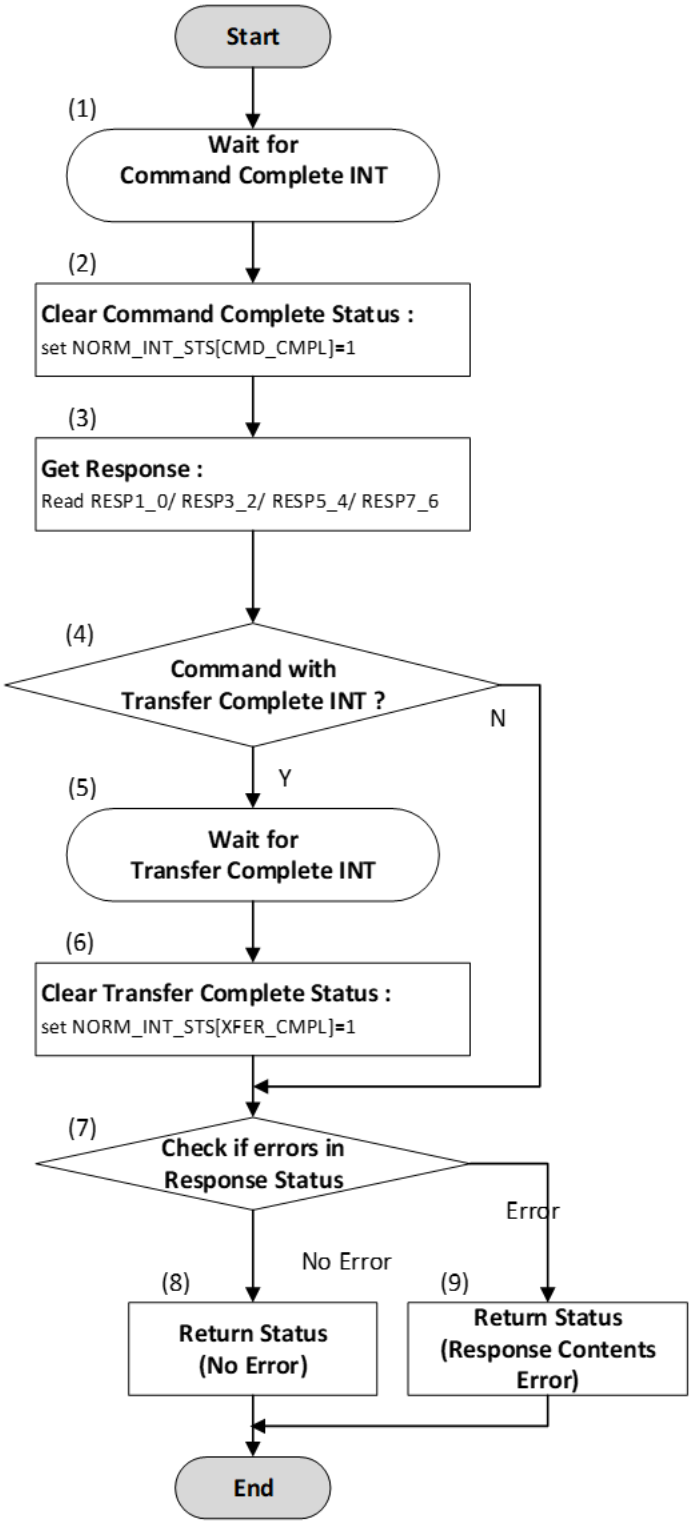


Diagram 21.35: command completion program

21.4.2.5 Stop or Abort Data Transfer

- Abort command program

  The abort command is completed by CMD12 for eMMC/SD devices, and CMD52 for SDIO devices. There are two main situations when using it:

  (1) Stop unlimited module data transmission.

  (2) Stop multi-module data transmission.

  The stop command program is as shown in the diagram: ref:*diagram_emmc_stop_procedure*. The detailed steps are as follows:

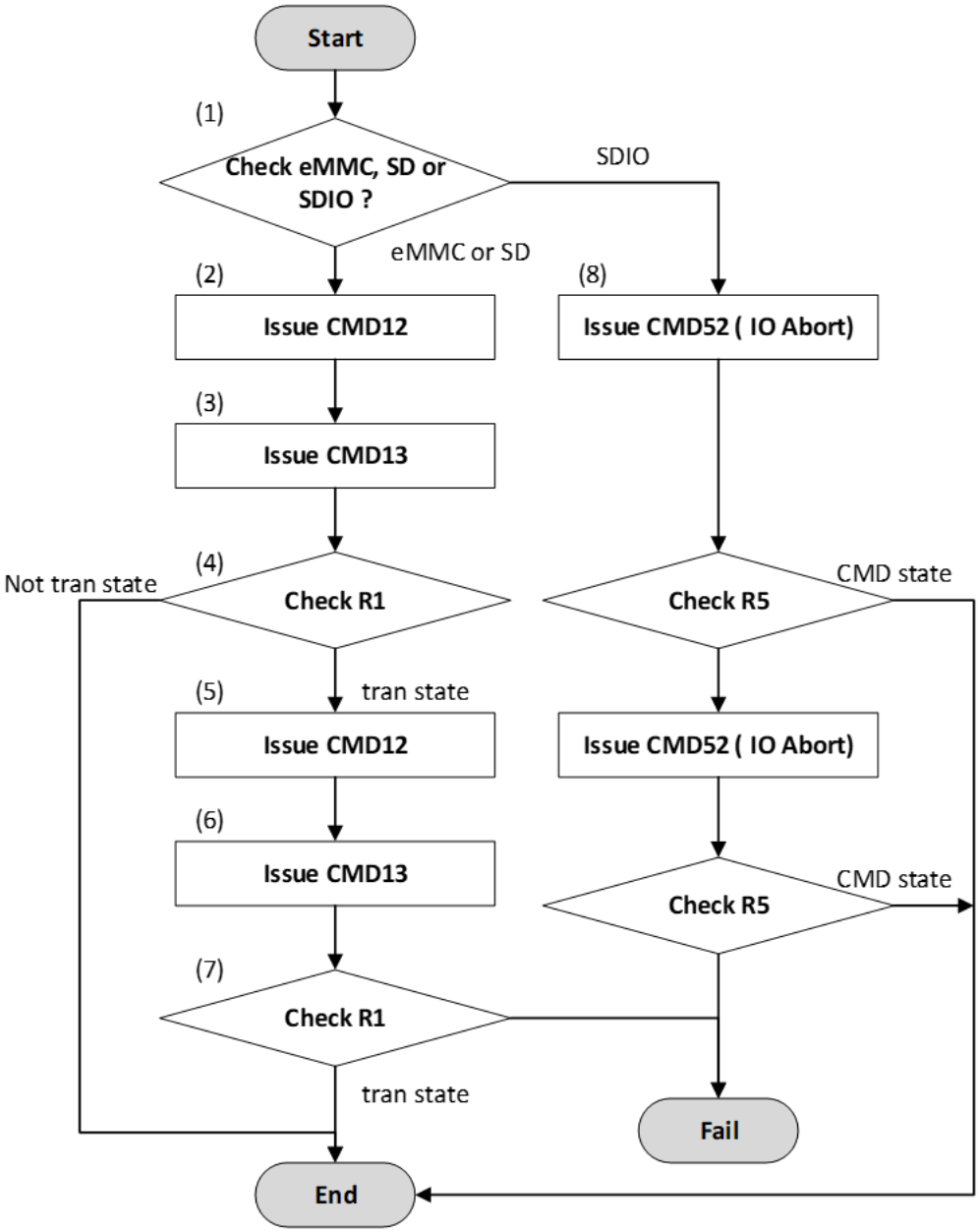


Diagram 21.36: abort command program

There are two ways of aborting instructions: synchronous abort instruction and asynchronous abort instruction.

- Asynchronous abort command program

  Diagram *Program chart of asynchronous abort instruction* is the program diagram of the non-synchronous stop instruction. Detailed steps are as follows:

  (1) Execute abort instructions according to different transmission modes.

  (2) Set SW_RST_CMD and SW_RST_DAT in the SW_RESET register to reset the CMD and DAT signal lines.

  (3) Check the bits SW_RESET[SW_RST_CMD] and SW_RESET[SW_RST_DAT] to confirm whether the reset is completed. If both are 0, end the program. If one of them is 1, return to step (3) and wait for a delay.

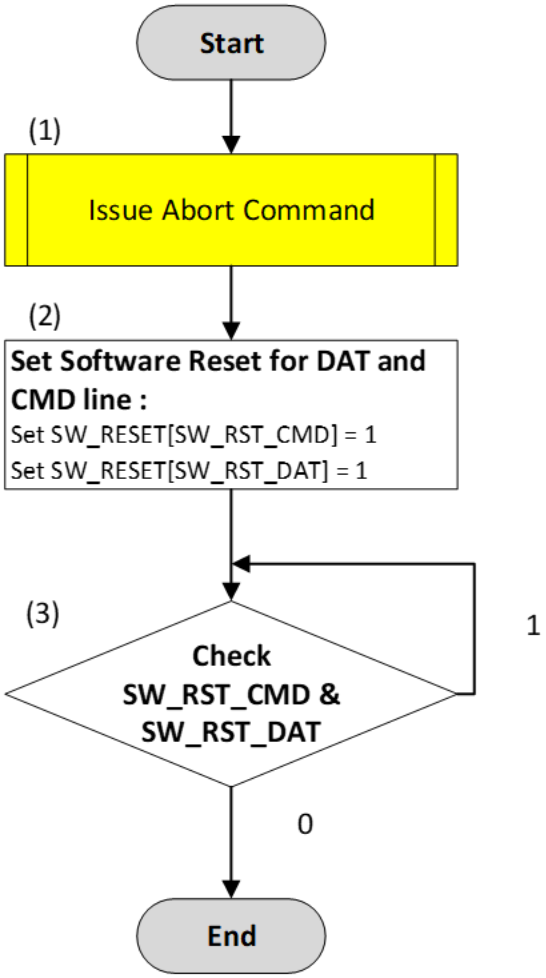


Diagram 21.37: Program chart of asynchronous abort instruction

- Synchronous abort command program

  Diagram *Program chart of synchronization abort instruction* is the program diagram of the synchronization stop instruction. Detailed steps are as follows:

  (1) Writing bit BG_CTL[STOP_BG_REQ] stops transmission in the Block Gap interval.

  (2) Interrupt NORM_INT_STS[XFER_CMPL] waiting for transfer completion.

(3) After receiving the interrupt, set NORM_INT_STS[XFER_CMPL]=1 to clear the XFER_CMPL interrupt status.

(4) Execute abort instructions according to different transmission modes.

(5) Set SW_RST_CMD and SW_RST_DAT in the SW_RESET register to reset the CMD and DAT signal lines.

(6) Check the bits SW_RESET[SW_RST_CMD] and SW_RESET[SW_RST_DAT] to confirm whether the reset is completed. If both are 0, end the program. If one of them is 1, return to step (6) to delay waiting.

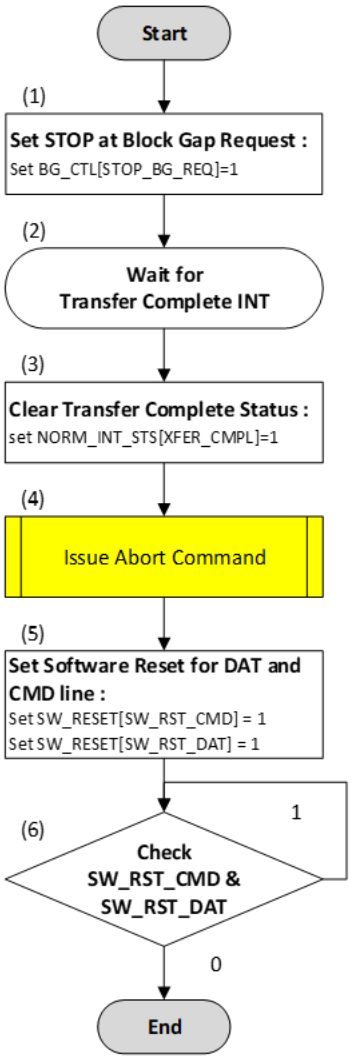


Diagram 21.38: Program chart of synchronization abort instruction

21.4.2.6 Non-DMA data Transfer Mode

The program for non-DMA data transfer mode is as shown in the diagram *Program for non-DMA data transfer mode*. Detailed steps are as follows:

(1) Write to the BLK_SIZE register to set the block size.

(2) Write to the BLK_CNT register to set the number of blocks.

(3) Write to the ARGUMENT register to set the command parameters.

(4) Write to the XFER_MODE register to set the transmission mode. The host can determine settings based on usage scenarios. Contains Single or Multiple Block Select, DMA Enable, Block Count Enable, Data Transfer Direction, Auto CMD Enable.

(5) Write to the CMD register to set the command and response types.

(6) Interrupt NORM_INT_STS[CMD_CMPL] waiting for Command completion.

(7) After receiving the interrupt, set NORM_INT_STS[CMD_CMPL]=1 to clear the CMD_CMPL interrupt status.

(8) Then read the RESP1_0, RESP3_2, RESP5_4, RESP7_6 and other temporary registers to obtain the response value.

(9) If it is a read operation, step (14) will be executed, if it is a write operation, step (10) will be executed.

(10) Waiting for the Buffer Write Ready interrupt NORM_INT_STS[BUF_WRDY].

(11) After receiving the interrupt, set NORM_INT_STS[BUFF_WRDY]=1 to clear the BUFF_WRDY interrupt status.

(12) Write the data you want to write to the device sequentially into the BUF_DATA register.

(13) If there are more blocks to be written, go back to step (10) until the last block is written, then go to step (18).

(14) Waiting for the Buffer Read Ready interrupt NORM_INT_STS[BUF_RRDY].

(15) After receiving the interrupt, set NORM_INT_STS[BUFF_RRDY]=1 to clear the BUFF_RRDY interrupt status.

(16) Read the data received from the device in sequence from the BUF_DATA register.

(17) If there are more blocks to be read, go back to step (14) until the last block is read, then go to step (18).

(18) Determine whether it is single module transmission, multi-module transmission or unlimited module transmission. If it is a single module or multi-module transmission, skip to step (19). If it is an infinite module transfer, jump to step (21) and perform the action of aborting the transfer.

(19) Interrupt NORM_INT_STS[XFER_CMPL] waiting for completion of data transfer.

(20) After receiving the interrupt, set NORM_INT_STS[CMD_XFER]=1 to clear the XFER_CMPL interrupt status.

(21) Execute the abort transfer procedure.

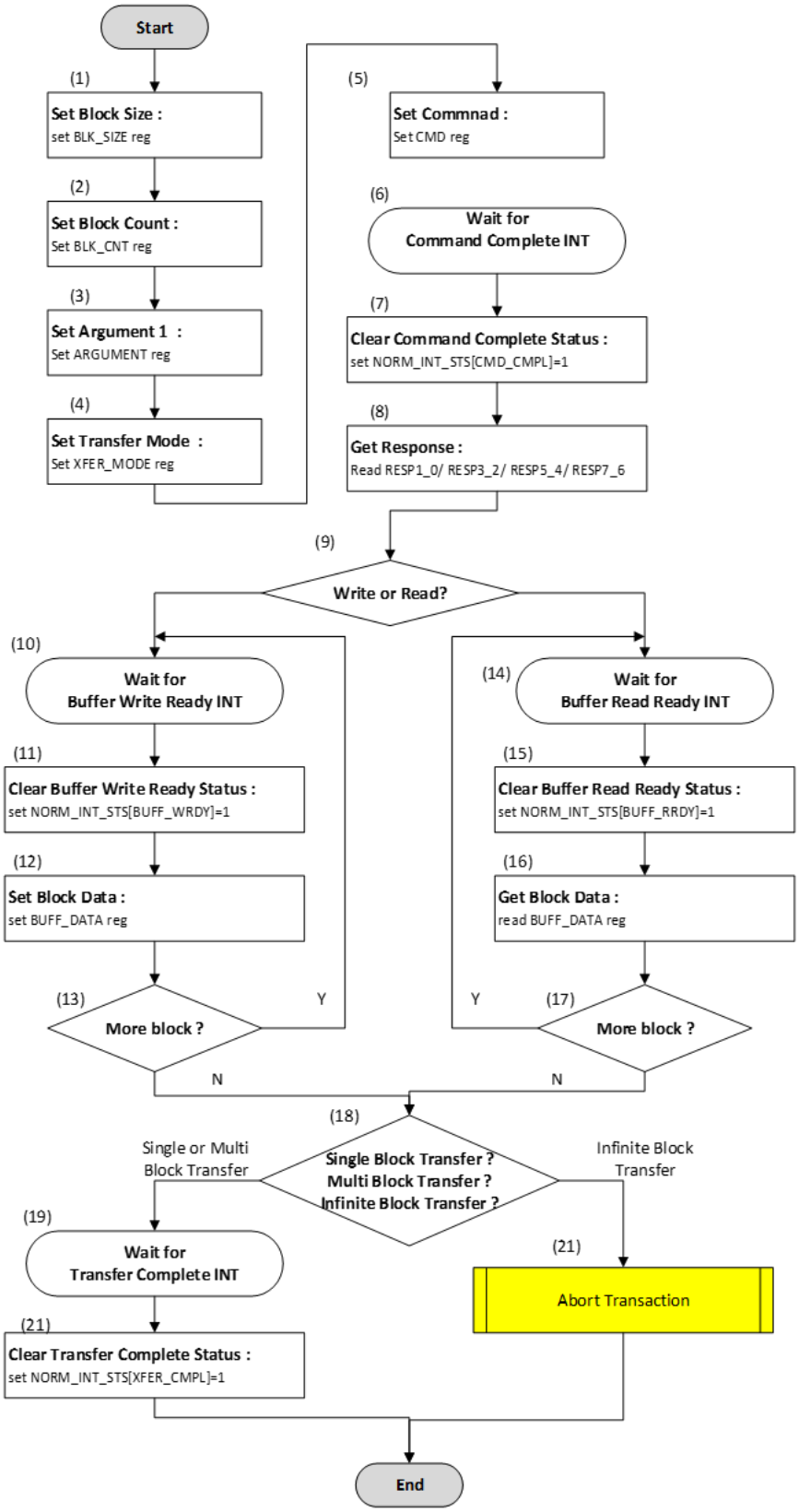


Diagram 21.39: Program for non-DMA data transfer mode

21.4.2.7 SDMA Data Transfer Mode

The SDMA data transfer mode program is shown in the diagram: ref:*diagram_sdma_data_transfer*, and the detailed steps are as follows:

(1) Write to the SDMA_SA register to set the starting address of the system memory used for data transmission.

(2) Write to the BLK_SIZE register to set the block size.

(3) Write to the BLK_CNT register to set the number of blocks.

(4) Write to the ARGUMENT register to set the command parameters.

(5) Write to the XFER_MODE register to set the transmission mode. The host can determine settings based on usage scenarios. Contains Single or Multiple Block Select, DMA Enable, Block Count Enable, Data Transfer Direction, Auto CMD Enable.

(6) Write to the CMD register to set the command and response types.

(7) Interrupt NORM_INT_STS[CMD_CMPL] waiting for Command completion.

(8) After receiving the interrupt, set NORM_INT_STS[CMD_CMPL]=1 to clear the CMD_CMPL interrupt status.

(9) Then read the RESP1_0, RESP3_2, RESP5_4, RESP7_6 and other temporary registers to obtain the response value.

(10) Wait for data transfer interrupt and DMA interrupt.

(11) Read the interrupt status register NORM_INT_STS to determine the interrupt type. Skip to step (12) if it is a DMA interrupt, or skip to step (14) if it is a data transfer interrupt.

(12) Set NORM_INT_STS[DMA_INT]=1 to clear the DMA_INT status value.

(13) Write to the SDMA_SA register to reset the starting address of the system memory for the next DMA. Then skip to step (10).

(14) Set NORM_INT_STS[DMA_INT]=1 and [NORM_INT_STS[XFER_CMPL]=1 to clear the DMA_INT and XFER_CMPL status values. Then end the program.

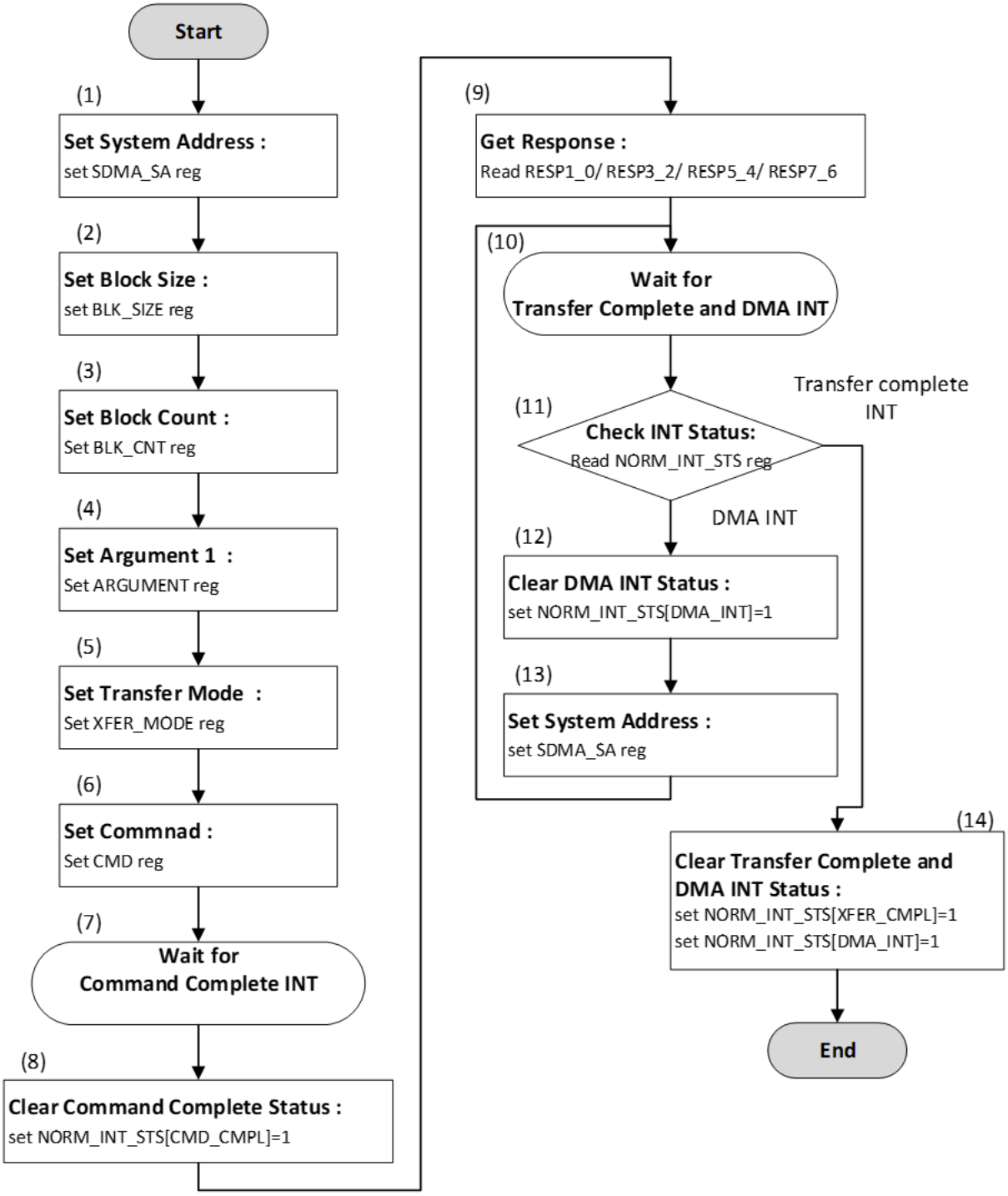


Diagram 21.40: SDMA data transfer mode program

21.4.2.8 ADMA Data Transfer Mode

The ADMA data transfer mode program is shown in the diagram *ADMA data transfer mode program*, and the detailed steps are as follows:

(1) Fill in the ADMA description table into the system memory.

(2) Write to the ADMA_SA_L and ADMA_SA_H registers to set the starting address of the system memory used by the description table.

(3) Write to the BLK_SIZE register to set the block size.

(4) Write to the BLK_CNT register to set the number of blocks.

(5) Write to the ARGUMENT register to set the command parameters.

(6) Write to the XFER_MODE register to set the transmission mode. The host can determine settings based on usage scenarios. Contains Single or Multiple Block Select, DMA Enable, Block Count Enable, Data Transfer Direction, Auto CMD Enable.

(7) Write to the CMD register to set the command and response types.

(8) Interrupt NORM_INT_STS[CMD_CMPL] waiting for Command completion.

(9) After receiving the interrupt, set NORM_INT_STS[CMD_CMPL]=1 to clear the CMD_CMPL interrupt status.

(10) Then read the RESP1_0, RESP3_2, RESP5_4, RESP7_6 and other temporary registers to obtain the response value.

(11) Wait for data transfer interrupt or ADMA Error interrupt.

(12) Read the interrupt status register NORM_INT_STS and ERR_INT_STS to determine the interrupt type. Skip to step (13) if it is an ADMA error interrupt, or step (15) if it is a data transfer interrupt.

(13) Set ERR_INT_STS[ADMA_ERR]=1 to clear the ADMA_ERR status value.

(14) Enter the ADMA Abort Transaction program (ADMA Abort Transaction) and execute the Abort Command to abort the data transfer with the device. If necessary, you can check the ADMA Error Status register to check the cause of the error.

(15) Set NORM_INT_STS[XFER_CMPL]=1 to clear the XFER_CMPL status value. Then end the program.

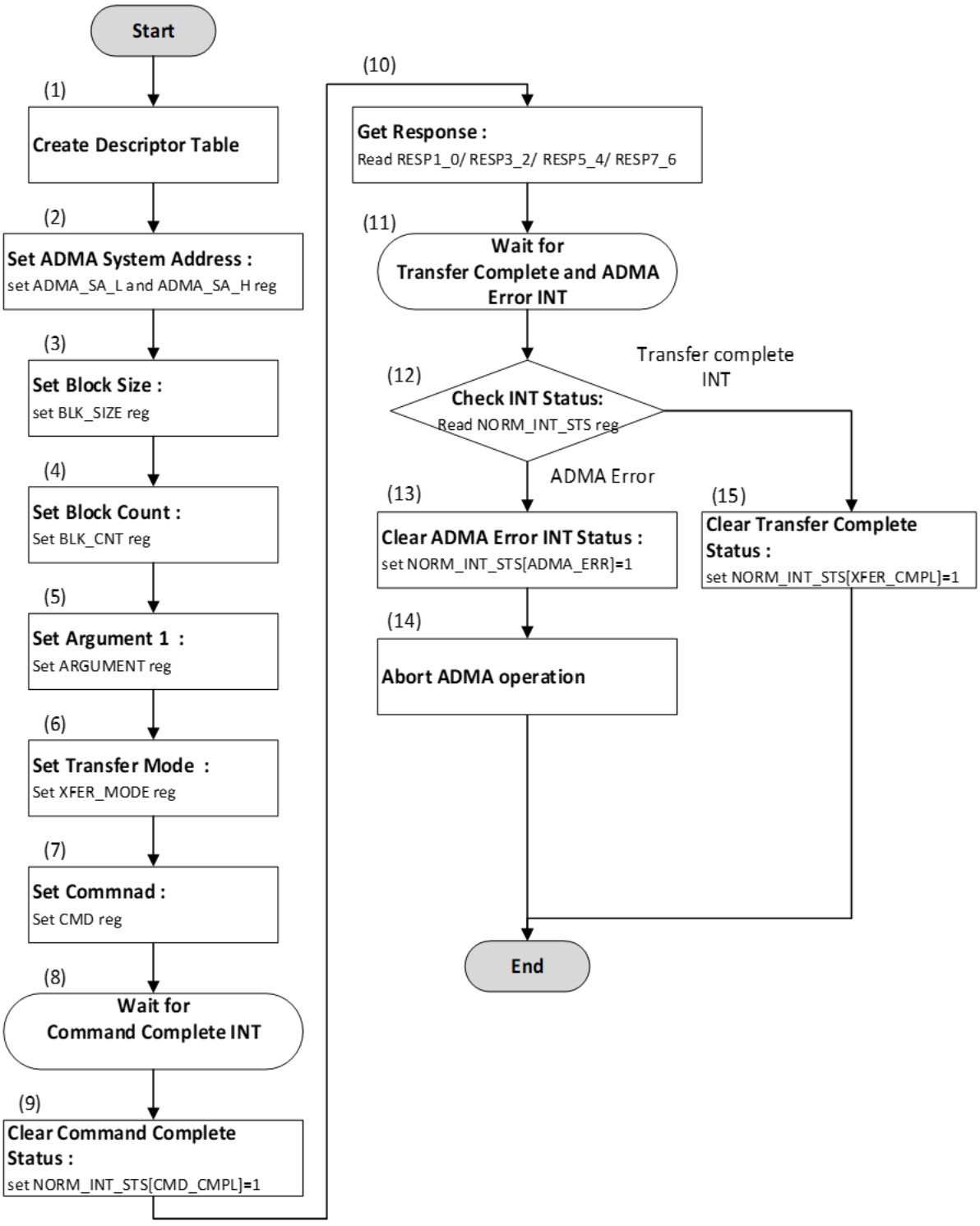


Diagram 21.41: ADMA data transfer mode program

21.4.3 Register Overview

Table *SDMMC Registers Overview* is the SDMMC register overview (the base address of EMMC is 0x0430_0000, SDIO0 is 0x0431_0000, SDIO1 is 0x0500_0000)

Table 21.102: SDMMC Registers Overview

Name	Address Offset	Description
SDMA_SADDR	0x000	SDMA System Memory Address/ Argument2
BLK_SIZE_AND_CNT	0x004	Block Size and Block Count Register
ARGUMENT	0x008	Argument 1 Register
XFER_MODE_AND_CMD	0x00c	Transfer Mode and Command Register
RESP31_0	0x010	Response Bit 31-0 Regsiter
RESP63_32	0x014	Response Bit 63-32 Regsiter
RESP95_64	0x018	Response Bit 95-64 Regsiter
RESP127_96	0x01c	Response Bit 127-96 Regsiter
BUF_DATA	0x020	Buffer Data Port Register
PRESENT_STS	0x024	Present State Register
HOST_CTL1_PWR_BG_WUP	0x028	Host Control 1 , Power, Block Gap and Wakeup Register
CLK_CTL_SWRST	0x02c	Clock and Reset Control Register
NORM_AND_ERR_INT_STS	0x030	Normal and Error Interrupt Status Register
NORM_AND_ERR_INT_STS_EN	0x034	Normal and Error Interrupt Status Enable Register
NORM_AND_ERR_INT_SIG_EN	0x038	Normal and Error Interrupt Signal Enable Register
AUTO_CMD_ERR_AND_HOST_CTL2	0x03c	Auto CMD Error Status Register and Host Control 2 register
CAPABILITIES1	0x040	Capabilities 1 Register
CAPABILITIES2	0x044	Capabilities 2 Register
FORCE_EVENT_ERR	0x050	Force Event Register for Auto CMD Error Status
ADMA_ERR_STS	0x054	ADMA Error Status Register
ADMA_SADDR_L	0x058	ADMA System Address Register for low 32-bit
ADMA_SADDR_H	0x05c	ADMA System Address Register for high 32-bit
PRESENT_VUL_INIT_DS	0x060	Present Value Register for Initialization and Default Speed
PRESENT_VUL_HS_SDR12	0x064	Present Value Register for High-speed and SDR12
PRESENT_VUL_SDR25_SDR50	0x068	Present Value Register for SDR25 and SDR50
PRESENT_VUL_SDR104_DDR50	0x06c	Present Value Register for SDR104 and DDR50
SLOT_INT_AND_HOST_VER	0x0fc	Slot Interrupt Status and Host Controller Version Register
EMMC_CTRL	0x200	MSHC Control register
EMMC_BOOT_CTL	0x204	eMMC Boot Control Register
CDET_TOUT_CTL	0x208	Card Detect Control Register

continues on next page

Table 21.102 – continued from previous page

Name	Address Offset	Description
MBIU_CTRL	0x20c	MBIU Control register
PHY_TX_RX_DLY	0x240	PHY tx and rx delay line register
PHY_DS_DLY	0x244	PHY DS delay line register
PHY_DLY_STS	0x248	PHY delay line status register
PHY_CONFIG	0x24c	PHY Configuration register

21.4.4 Register Description

The following is a detailed register description.

21.4.4.1 SDMA_SADDR

SDMA System Memory Address/ Argument2

Table 21.103: SDMA_SADDR, Offset Address: 0x000

Bits	Name	Access	Description	Reset
31:0	SDMA_SA	R/W	Physical system memory address used for DMA transfer and the second argument for Auto CMD23	0x0

21.4.4.2 BLK_SIZE_AND_CNT

Block Size and Block Count Register

Table 21.104: BLK_SIZE_AND_CNT, Offset Address: 0x004

Bits	Name	Access	Description	Reset
11:0	XFER_BLK_SIZE	R/W	Block Size of data transfer. • 0x1 : 1 byte • 0x2 : 2 bytes • ..... • 0x200 : 512 bytes • ..... • 0x800 : 2048 bytes	0x0
14:12	SDMA_BUF_BDARY	R/W	Host SDMA buffer Boundary • 0x0 (4K bytes) • 0x1 (8K bytes) • 0x2 (16K bytes) • 0x3 (32K bytes) • 0x4 (64K bytes) • 0x5 (128K bytes) • 0x6 (256K bytes) • 0x7 (512K bytes)	0x0
15	Reserved			
31:16	BLK_CNT	R/W	Blocks Count for Current Transfer	0x0

21.4.4.3 ARGUMENT

Argument 1 Register

Table 21.105: ARGUMENT, Offset Address: 0x008

Bits	Name	Access	Description	Reset
31:0	ARGUMENT	R/W	Command Argument 1	0x0

21.4.4.4 XFER_MODE_AND_CMD

Transfer Mode and Command Register

Table 21.106: XFER_MODE_AND_CMD, Offset Address: 0x00c

Bits	Name	Access	Description	Reset
0	DMA_ENABLE	R/W	DMA enable 1 : DMA Data Transfer 0 : No data transfer or Non DMA data transfer	0x0
1	BLK_CNT_ENABLE	R/W	Block Count Enable. This bit is used to enable the block count regiser, which is only relevant for multiple block transfers. 1 : Enable 0 : Disable	0x0
3:2	AUTO_CMD_ENABLE	R/W	Auto CMD enable. This field determines use of auto command funcitons 0x0 : Auto command Disabled 0x1 : Auto CMD12 Enable 0x2 : Auto CMD23 Enable 0x3 : Reserved	0x0
4	DAT_XFER_DIR	R/W	Data Transfer Direction Select 1 : Read (card to host) 0 : Write (host to card)	0x0
5	MULTI_BLK_SEL	R/W	Multi/single Block Select 1 : Multiple block transfer 0 : Single block transfer	0x0
6	RESP_TYPE	R/W	Response Type R1/R5 0x0 : R1 (Memory) 0x1 : R5 (SDIO)	0x0
7	RESP_ERR_CHK_ENABLE	R/W	Response Error Check Enable 1 : Enable 0 : Disable	0x0
8	RESP_INT_DISABLE	R/W	Response Interrupt Disable 1 : Disable 0 : Enable	0x0
15:9	Reserved			
17:16	RESP_TYPE_SEL	R/W	Response Type Select 0x0 : No Response 0x1 : Response Length 136 0x2 : Response Length 48 0x3 : Response Length 48 with busy	0x0
18	SUB_CMD_FLAG	R/W	Sub Command Flag 1 : Sub Command 0 : Main Command	0x0
19	CMD_CRC_CHK_ENABLE	R/W	Command CRC check enable 1 : Enable 0 : Disable	0x0

To be continued ……

Table 21.107: XFER_MODE_AND_CMD, Offset Address: 0x00c
(continued)

Bits	Name	Access	Description	Reset
20	CMD_IDX_CHK_ENABLE	R/W	Command Index Check Enable 1 : Enable 0 : Disable	0x0
21	DATA_PRESENT_SEL	R/W	Data Present Select. It is set to 0 for following : (1) Commands using only CMD line (ex. CMD52) (2) Commands with no data transfer but using busy signal on DAT0 (ex. R1b) (3) Resume command 1 : Data Present 0 : No Data Present	0x0
23:22	CMD_TYPE	R/W	Command Type 0x0 : Normal 0x1 : Suspend (CMD52 for writing “Bus Suspend” in CCCR) 0x2 : CMD52 for writing “Function Select” in CCCR) 0x3 : Abort (CMD12, CMD52 for writing “I/O Abort” in CCCR)	0x0
29:24	CMD_IDX	R/W	Command Index	0x0
31:30	Reserved			

21.4.4.5 RESP31_0

Response Bit 31-0 Regsiter

Table 21.108: RESP31_0, Offset Address: 0x010

Bits	Name	Access	Description	Reset
31:0	RESP31_0	RO	Command Response for RSP[39:8]	

21.4.4.6 RESP63_32

Response Bit 63-32 Regsiter

Table 21.109: RESP63_32, Offset Address: 0x014

Bits	Name	Access	Description	Reset
31:0	RESP63_32	RO	Command Response for RSP[71:40]	

21.4.4.7 RESP95_64

Response Bit 95-64 Regsiter

Table 21.110: RESP95_64, Offset Address: 0x018

Bits	Name	Access	Description	Reset
31:0	RESP95_64	RO	Command Response for RSP[103:72]	

21.4.4.8 RESP127_96

Response Bit 127-96 Regsiter

Table 21.111: RESP127_96, Offset Address: 0x01c

Bits	Name	Access	Description	Reset
31:0	RESP127_96	RO	Command Response for RSP[135:104]	

21.4.4.9 BUF_DATA

Buffer Data Port Register

Table 21.112: BUF_DATA, Offset Address: 0x020

Bits	Name	Access	Description	Reset
31:0	BUF_DATA	R/W	Buffer Data	0x0

21.4.4.10 PRESENT_STS

Present State Register

Table 21.113: PRESENT_STS, Offset Address: 0x024

Bits	Name	Access	Description	Reset
0	CMD_INHIBIT	RO	Command Inhibit (CMD) 1 : Cannot issue command 0 : Can issue command using only CMD line	
1	CMD_INHIBIT_DAT	RO	Command Inhibit (DAT) 1 : Cannot issue command wihich used the DAT line 0 : Can issue command using only DAT line	
2	DAT_LINE_ACTIVE	RO	DAT Line Active This bit indicates whether one of the DAT line on SD Bus is in use. 1 : DAT Line Active 0 : DAT Line Inactive	
3	RE_TUNE_REQ	RO	Re-Tuning Request 1 : Sampling clock need re-tuning 0 : Fixed or well tuned sampling clock	
7:4	Reserved			
8	WR_XFER_ACTIVE	RO	Write Transfer Active 1 : Transferring data 0 : No valid data	
9	RD_XFER_ACTIVE	RO	Read Transfer Active 1 : Transferring data 0 : No valid data	
10	BUF_WR_ENABLE	RO	Buffer Write Enable 1 : Enable 0 : Disable	
11	BUF_RD_ENABLE	RO	Buffer Read Enable 1 : Enable 0 : Disable	
15:12	Reserved			
16	CARD_INSERTED	RO	Card Inserted 1 : Card Inserted 0 : Reset or Debouncing or No card	
17	CARD_STABLE	RO	Card State Stable 1 : No Card or Inserted 0 : Reset or Debouncing	
18	CARD_CD_STS	RO	Card Detect Pin Level 1 : Card Present (SD_CD = 0) 0 : No Card Present (SD_CD = 1)	
19	CARD_WP_STS	RO	Write Protect Switch Pin Level 1 : Write enabled (SD_WP =0) 0 : Write protected (SD_WP = 1)	
23:20	DAT_3_0_STS	RO	DAT[3:0] Line Signal Level	
24	CMD_LINE_STS	RO	CMD Line Signal Level	
31:25	Reserved			

21.4.4.11 HOST_CTL1_PWR_BG_WUP

Host Control 1 Register

Table 21.114: HOST_CTL1_PWR_BG_WUP, Offset Address: 0x028

Bits	Name	Access	Description	Reset
0	LEC_CTL	R/W	LED Control This bit is used to caution the user not to remove the card while the SD card is being accessed. 1 : LED on 0 : LED off	0x0
1	DAT_XFER_WIDTH	R/W	Data Transfer Width. 1 : 4-bit mode 0 : 1-bit mode	0x0
2	HS_ENABLE	R/W	High Speed Enable 1 : High Speed Enable 0 : Normal Speed Enable	0x0
4:3	DMA_SEL	R/W	DMA Select. 0x0 : SDMA mode 0x1 : Reserved 0x2 : ADMA2 0x3 : ADMA2 or ADMA3	0x0
5	EXT_DAT_WIDTH	R/W	Extended Data Transfer Width 1 : 8-bit mode 0 : Selected by DAT_XFER_WIDTH	0x0
6	CRAD_DET_TEST	R/W	Card Detect Test Level 1 : Card Inserted 0 : No card	0x0
7	CARD_DET_SEL	R/W	Card Detect Signal Selection 1 : CARD_DET_TEST is selected (for test purpose) 0 : SD_CD Is selected	0x0
8	SD_BUS_PWR	R/W	SD Bus Power. 1 : Power on 0 : Power off	0x0
11:9	SD_BUS_VOL_SEL	R/W	SD Bus Voltage Select 111b : 3.3V 110b : 3.0V 101b : 1.8V 100b - 000b : Reserved	0x0
15:12	Reserved			
16	STOP_BG_REQ	R/W	Stop At Block Gap Request. This bit is used to stop executing read and write transaction at the next block gap for non-DMA, SDMA and ADMA transfers. 1 : Stop 0 : Transfer	0x0

To be continued ……

Table 21.115: HOST_CTL1_PWR_BG_WUP, Offset Address: 0x028
(continued)

Bits	Name	Access	Description	Reset
17	CONTINUE_REQ	R/W	Continue Request. This bit is used to restart a transaction, which was stoped using the STOP_BG_REQ. 1 : Restart 0 : Not affect	0x0
18	READ_WAIT	R/W	Read Wait Control 1 : Enable Read Wait Control 0 : Disable Read Wait Control	0x0
19	INT_BG	R/W	Interrupt At Block Gap 1 : Enable 0 : Disabel	0x0
23:20	Reserved			
24	WAKEUP_ON_CARD_INT	R/W	Wakeup Event Enable On Card Interrupt. 1 : Enable 0 : Disable	0x0
25	WAKEUP_ON_CARD_INSERT	R/W	Wakeup Event Enable On Card Insertion. 1 : Enable 0 : Disable	0x0
26	WAKEUP_ON_CARD_REMV	R/W	Wakeup Event Enable On Card Removal. 1 : Enable 0 : Disable	0x0
31:27	Reserved			

21.4.4.12 CLK_CTL_SWRST

Clock and Timeout Control Register

Table 21.116: CLK_CTL_SWRST, Offset Address: 0x02c

Bits	Name	Access	Description	Reset
0	INT_CLK_EN	R/W	Internal Clock Enable 1 : Oscillate 0 : Stop	0x0
1	INT_CLK_STABLE	RO	Internal Clock Stable. 1 : Ready 0 : Not Ready	
2	SD_CLK_EN	R/W	SD Clock Enable for Card 1 : Enable 0 : Disable	0x0
3	PLL_EN	R/W	PLL Enable 1 : Enable 0 : Disable	0x0
5:4	Reserved			
7:6	UP_FREQ_SEL	R/W	Upper Bits of SDCLK Frequency Select	0x0
15:8	FREQ_SEL	R/W	SDCLK Frequency Select	0x0
19:16	TOUT_CNT	R/W	Data Timeout Counter Value 0x0 : TMCLK x 2^13 0x1 : TMCLK x 2^14 ..... 0xe : TMCLK x 2^ 27 0xf : Reserved	0x0
23:20	Reserved			
24	SW_RST_ALL	R/W	Software Reset For All	0x0
25	SW_RST_CMD	R/W	Software Reset For CMD Line	0x0
26	SW_RST_DAT	R/W	Software Reset For DATA Line	0x0
31:27	Reserved			

21.4.4.13 NORM_AND_ERR_INT_STS

Normal and Error Interrupt Status Register

Table 21.117: NORM_AND_ERR_INT_STS, Offset Address: 0x030

Bits	Name	Access	Description	Reset
0	CMD_CMPL	RWC	Command Complete	
1	XFER_CMPL	RWC	Transfer Complete	
2	BG_EVENT	RWC	Block Gap Event	
3	DMA_INT	RWC	DMA Interrupt	
4	BUF_WRDY	RWC	Buffer Write Ready	
5	BUF_RRDY	RWC	Buffer Read Ready	
6	CARD_INSERT_INT	RWC	Card Insertion	
7	CARD_REMOV_INT	RWC	Card Removal	
8	CARD_INT	RO	Card Interrupt	
9	INT_A	RO	INT_A. This status is set if INT_A is enabled and INT_A pin is in low level	
10	INT_B	RO	INT_B. This status is set if INT_B is enabled and INT_B pin is in low level	

continues on next page

Table 21.117 – continued from previous page

Bits	Name	Access	Description	Reset
11	INT_C	RO	INT_C. This status is set if INT_C is enabled and INT_C pin is in low level	
12	RE_TUNE_EVENT	RO	Re-Tuning Event	
13	Reserved			
14	CQE_EVENT	RO	Command Queuing Event	
15	ERR_INT	RO	Error Interrupt	
16	CMD_TOUT_ERR	RWC	Command Timeout Error	
17	CMD_CRC_ERR	RWC	Command CRC Error	
18	CMD_ENDBIT_ERR	RWC	Command End Bit Error	
19	CMD_IDX_ERR	RWC	Command Index Error	
20	DAT_TOUT_ERR	RWC	Data Timeout Error	
21	DAT_CRC_ERR	RWC	Data CRC Error	
22	DAT_ENDBIT_ERR	RWC	Data End Bit Error	
23	CURR_LIMIT_ERR	RWC	Current Limit Error	
24	AUTO_CMD_ERR	RWC	Auto Command Error	
25	ADMA_ERR	RWC	ADMA Error	
26	TUNE_ERR	RWC	Tuning Error	
27	Reserved			
28	BOOT_ACK_ERR	RWC		
31:29	Reserved			

21.4.4.14 NORM_AND_ERR_INT_STS_EN

Normal and Error Interrupt Status Enable Register

Table 21.118: NORM_AND_ERR_INT_STS_EN, Offset Address: 0x034

Bits	Name	Access	Description	Reset
0	CMD_CMPL_EN	R/W	Command Complete Status Enable	0x0
1	XFER_CMPL_EN	R/W	Transfer Complete Status Enable	0x0
2	BG_EVENT_EN	R/W	Block Gap Event Status Enable	0x0
3	DMA_INT_EN	R/W	DMA Interrupt Status Enable	0x0
4	BUF_WRDY_EN	R/W	Buffer Write Ready Status Enable	0x0
5	BUF_RRDY_EN	R/W	Buffer Read Ready Status Enabel	0x0
6	CARD_INSERT_INT_EN	R/W	Card Insertion Status Enable	0x0
7	CARD_REMOV_INT_EN	R/W	Card Removal Status Enable	0x0
8	CARD_INT_EN	R/W	Card Interrupt Status Enable	0x0
9	INT_A_EN	R/W	INT_A Status Enable.	0x0
10	INT_B_EN	R/W	INT_B Status Enable.	0x0
11	INT_C_EN	R/W	INT_C Status Enable.	0x0
12	RE_TUNE_EVENT_EN	R/W	Re-Tuning Event Status Enable	0x0
13	Reserved			
14	CQE_EVENT_EN	R/W	Command Queuing Event Status Enable	0x0
15	ERR_INT_EN	R/W	Error Interrupt Status Enable	0x0
16	CMD_TOUT_ERR_EN	R/W	Command Timeout Error Status Enable	0x0
17	CMD_CRC_ERR_EN	R/W	Command CRC Error Status Enable	0x0
18	CMD_ENDBIT_ERR_EN	R/W	Command End Bit Error Status Enable	0x0

continues on next page

Table 21.118 – continued from previous page

Bits	Name	Access	Description	Reset
19	CMD_IDX_ERR_EN	R/W	Command Index Error Status Enable	0x0
20	DAT_TOUT_ERR_EN	R/W	Data Timeout Error Status Enable	0x0
21	DAT_CRC_ERR_EN	R/W	Data CRC Error Status Enable	0x0
22	DAT_ENDBIT_ERR_EN	R/W	Data End Bit Error Statue Enable	0x0
23	CURR_LIMIT_ERR_EN	R/W	Current Limit Error Status Enable	0x0
24	AUTO_CMD_ERR_EN	R/W	Auto Command Error Status Enable	0x0
25	ADMA_ERR_EN	R/W	ADMA Error Status Enable	0x0
26	TUNE_ERR_EN	R/W	Tuning Error Status Enable	0x0
27	Reserved			
28	BOOT_ACK_ERR_EN	R/W	Boot Ack Error Status Enable	0x0
31:29	Reserved			

21.4.4.15 NORM_AND_ERR_INT_SIG_EN

Normal and Error Interrupt Signal Enable Register

Table 21.119: NORM_AND_ERR_INT_SIG_EN, Offset Address: 0x038

Bits	Name	Access	Description	Reset
0	CMD_CMPL_SIG_EN	R/W	Command Complete Signal Enable	0x0
1	XFER_CMPL_SIG_EN	R/W	Transfer Complete Signal Enable	0x0
2	BG_EVENT_SIG_EN	R/W	Block Gap Event Signal Enable	0x0
3	DMA_INT_SIG_EN	R/W	DMA Interrupt Signal Enable	0x0
4	BUF_WRDY_SIG_EN	R/W	Buffer Write Ready Signal Enable	0x0
5	BUF_RRDY_SIG_EN	R/W	Buffer Read Ready Signal Enabel	0x0
6	CARD_INSERT_INT_SIG_EN	R/W	Card Insertion Signal Enable	0x0
7	CARD_REMOV_INT_SIG_EN	R/W	Card Removal Signal Enable	0x0
8	CARD_INT_SIG_EN	R/W	Card Interrupt Signal Enable	0x0
9	INT_A_SIG_EN	R/W	INT_A Signal Enable.	0x0
10	INT_B_SIG_EN	R/W	INT_B Signal Enable.	0x0
11	INT_C_SIG_EN	R/W	INT_C Signal Enable.	0x0
12	RE_TUNE_EVENT_SIG_EN	R/W	Re-Tuning EventSignal Enable	0x0
13	Reserved			
14	CQE_EVENT_SIG_EN	R/W	CQE EventSignal Enable	0x0
15	Reserved			
16	CMD_TOUT_ERR_SIG_EN	R/W	Command Timeout Error Signal Enable	0x0
17	CMD_CRC_ERR_SIG_EN	R/W	Command CRC Error Signal Enable	0x0
18	CMD_ENDBIT_ERR_SIG_EN	R/W	Command End Bit Error Signal Enable	0x0
19	CMD_IDX_ERR_SIG_EN	R/W	Command Index Error Signal Enable	0x0
20	DAT_TOUT_ERR_SIG_EN	R/W	Data Timeout Error Signal Enable	0x0
21	DAT_CRC_ERR_SIG_EN	R/W	Data CRC Error Signal Enable	0x0
22	DAT_ENDBIT_ERR_SIG_EN	R/W	Data End Bit Error Signal Enable	0x0
23	CURR_LIMIT_ERR_SIG_EN	R/W	Current Limit Error Signal Enable	0x0
24	AUTO_CMD_ERR_SIG_EN	R/W	Auto Command Error Signal Enable	0x0
25	ADMA_ERR_SIG_EN	R/W	ADMA Error Signal Enable	0x0
26	TUNE_ERR_SIG_EN	R/W	Tuning Error Signal Enable	0x0
27	Reserved			
28	BOOT_ACK_ERR_SIG_EN	R/W	Boot Ack Error Signal Enable	0x0

continues on next page

Table 21.119 – continued from previous page

Bits	Name	Access	Description	Reset
31:29	Reserved			

21.4.4.16 AUTO_CMD_ERR_AND_HOST_CTL2

Auto CMD Error Status Register and Host Control 2 register

Table 21.120: AUTO_CMD_ERR_AND_HOST_CTL2, Offset Address: 0x03c

Bits	Name	Access	Description	Reset
0	AUTO_CMD12_NO_EXE	RO	Auto CMD12 Not Executed	
1	AUTO_CMD_TOUT_ERR	RO	Auto CMD Timeout Error	
2	AUTO_CMD_CRC_ERR	RO	Auto CMD CRC Error	
3	AUTO_CMD_ENDBIT_ERR	RO	Auto CMD End Bit Error	
4	AUTO_CMD_IDX_ERR	RO	Auto CMD Index Error	
6:5	Reserved			
7	CMD_NOT_ISSUE_BY_CMD12	RO	Command Not Issued By Auto CMD12 Error	
15:8	Reserved			
18:16	UHS_MODE_SEL	R/W	USH Speed Mode Select (for SD) 0x0 : SDR12 0x1 : SDR25 0x2 : SDR50 0x3 : SDR104 0x4 : DDR50 0x5 : Reserved 0x6 : Reserved 0x7 : Reserved eMMC Speed Mode Select (for eMMC) 0x0 : Default speed 0x1 : High speed 0x2 : Reserved 0x3 : HS200 0x4 : DDR52 0x5 : Reserved 0x6 : Reserved 0x7 : Reserved	0x0
19	EN_18_SIG	R/W	1.8V Signaling Enable	0x0
21:20	DRV_SEL	R/W	Driver Strength Select 0x0 : Driver Type B 0x1 : Driver Type A 0x2 : Driver Type C 0x3 : Driver Type D	0x0
22	EXECUTE_TUNE	R/W	Execute Tuning 1 : Execute Tuning 0 : Not Tuned or Tuning Completed	0x0
23	SAMPLE_CLK_SEL	R/W	Sampling Clock Select 1 : Tuned clock is used to sample data 0 : Fixed clock is used to sample data	0x0
29:24	Reserved			
30	ASYNC_INT_EN	R/W	Asynchronous Interrupt Enable. 1 : Enable 0 : Disable	0x0
31	PRESET_VAL_ENABLE	R/W	Preset Value Enable 1 : Automatic Selection by Preset Value are Enabled 0 : SDLCK and Driver Strength are controlled by Host Driver	0x0

21.4.4.17 CAPABILITIES1

Capabilities 1 Register

Table 21.121: CAPABILITIES1, Offset Address: 0x040

Bits	Name	Access	Description	Reset
5:0	TOUT_CLK_FREQ	RO	Timeout Clock Frequency Not 0 : 1KHz~ 63KHz or 1Mhz~ 63Mhz	
6	Reserved			
7	TOUT_CLK_UNIT	RO	Timeout Clock Unit 1 : 1MHz 0 : 1KHz	
15:8	BASE_CLK_FREQ	RO	Base Clock Frequency for SD clock 0x0 : Get information through another method 0x1 : 1MHz 0x2 : 2MHz ..... 0xFF : 255Mhz	
17:16	MAX_BLK_LEN	RO	Max Block Length 0x0 : 512 (byte) 0x1 : 1024 0x2 : 2048 0x3 : Reserverd	
18	EMBEDDED_8BIT	RO	8-bit Support for Embedded Device	
19	ADMA2_SUPPORT	RO	ADMA2 Support	
20	Reserved			
21	HS_SUPPORT	RO	High Speed Support	
22	SDMA_SUPPORT	RO	SDMA Support	
23	SUSP_RES_SUPPORT	RO	Suspend/Resume Support	
24	V33_SUPPORT	RO	3.3V Support	
25	V30_SUPPORT	RO	3.0V Support	
26	V18_SUPPORT	RO	1.8V Support	
27	Reserved			
28	BUS64_SUPPORT	RO	64-bit System Bus Support	
29	ASYNC_INT_SUPPORT	RO	Asynchronous Interrupt Support	
31:30	SLOT_TYPE	RO	Slot Type 0x0 : Removable Card 0x1 : Embedded Slot 0x2 : Shared Bus Slot	

21.4.4.18 CAPABILITIES2

Capabilities 2 Register

Table 21.122: CAPABILITIES2, Offset Address: 0x044

Bits	Name	Access	Description	Reset
0	SDR50_SUPPORT	RO	SDR50 Support	
1	SDR104_SUPPORT	RO	SDR104 Support	
2	DDR50_SUPPORT	RO	DDR50 Support	
3	Reserved			
4	DRV_A_SUPPORT	RO	Driver Type A Support	
5	DRV_C_SUPPORT	RO	Driver Type C Support	
6	DRV_D_SUPPORT	RO	Driver Type D Support	
7	Reserved			
11:8	RETUNE_TIMER	RO	Timer Count for Re-Tuning 0x0 : Disable n : 2^(n-1) seconds 0xB : 1024 seconds 0xC ~ 0xE : Reserved 0xF : Get Information from other source	
12	Reserved			
13	TUNE_SDR50	RO	Use Tuning for SDR50	
15:14	RETUNE_MODE	RO	Re-Tuning Modes	
23:16	CLK_MULTIPLIER	RO	Clock Multiplier	
31:24	Reserved			

21.4.4.19 FORCE_EVENT_ERR

Force Event Register for Auto CMD Error Status

Table 21.123: CAPABILITIES2, Offset Address: 0x050

Bits	Name	Access	Description	Reset
0	FORCE_AUTO_CMD12_NOT_EXE	R/W	Force Event for Auto CMD12 Not Executed	0x0
1	FORCE_AUTO_CMD_TOUT_ERR	R/W	Force Event for Auto CMD Timeout Error	0x0
2	FORCE_AUTO_CMD_CRC_ERR	R/W	Force Event for Auto CMD CRC Error	0x0
3	FORCE_AUTO_CMD_EBIT_ERR	R/W	Force Event for Auto CMD End Bit Error	0x0
4	FORCE_AUTO_CMD_IDX_ERR	R/W	Force Event for Auto CMD Index Error	0x0
6:5	Reserved			
7	FORCE_AUTO_CMD_NOT_ISSUE	R/W	Force Event for Command Not Issued By Auto CMD12 Error	0x0
15:8	Reserved			
16	FORCE_CMD_TOUT_ERR	R/W	Force Event for Auto CMD12 Not Executed	0x0
17	FORCE_CMD_CRC_ERR	R/W	Force Event for CMD Timeout Error	0x0
18	FORCE_CMD_EBIT_ERR	R/W	Force Event for CMD End Bit Error	0x0
19	FORCE_CMD_IDX_ERR	R/W	Force Event for CMD Index Error	0x0
20	FORCE_DAT_TOUT_ERR	R/W	Force Event for DATA Timeout Error	0x0
21	FORCE_DAT_CRC_ERR	R/W	Force Event for DATA End Bit Error	0x0
22	FORCE_DAT_EBIT_ERR	R/W	Force Event for DATA Index Error	0x0
23	FORCE_CURR_LIMIT_ERR	R/W	Force Event for current limit error	0x0
24	FORCE_AUTO_CMD_ERR	R/W	Force Event for Auto CMD Error	0x0
25	FORCE_ADMA_ERR	R/W	Force Event for ADMA Error	0x0
26	FORCE_TUNING_ERR	R/W	Force Event for Tuning Error	0x0
27	Reserved			
28	FORCE_BOOT_ACK_ERR	R/W	Force Event for Response Error	0x0
31:29	Reserved			

21.4.4.20 ADMA_ERR_STS

ADMA Error Status Register

Table 21.124: ADMA_ERR_STS, Offset Address: 0x054

Bits	Name	Access	Description	Reset
1:0	ADMA_ERR_STS	RO	ADMA Error Status 0x0 : ST_STOP (Stop DMA) 0x1 : ST_FDS (Fetch Descriptor) 0x2 : Never set this state 0x3 : ST_TFR (transfer data)	
2	ADMA_LEN_MISMATCH	RO	ADMA Length Mismatch Error	
31:3	Reserved			

21.4.4.21 ADMA_SADDR_L

ADMA System Address Register for low 32-bit

Table 21.125: ADMA_SADDR_L, Offset Address: 0x058

Bits	Name	Access	Description	Reset
31:0	ADMA_SA_L	R/W	ADMA System Address for low 32-bit	0x0

21.4.4.22 ADMA_SADDR_H

ADMA System Address Register for high 32-bit

Table 21.126: ADMA_SADDR_H, Offset Address: 0x05c

Bits	Name	Access	Description	Reset
31:0	ADMA_SA_H	R/W	ADMA System Address for high 32-bit	0x0

21.4.4.23 PRESENT_VUL_INIT_DS

Present Value Register for Initialization and Default Speed

Table 21.127: PRESENT_VUL_INIT_DS, Offset Address: 0x060

Bits	Name	Access	Description	Reset
31:0	PRESENT_VUL_INIT_DS	RO	Present Value Register for Initialization and Default Speed	

21.4.4.24 PRESENT_VUL_HS_SDR12

Present Value Register for High-speed and SDR12

Table 21.128: PRESENT_VUL_HS_SDR12, Offset Address: 0x064

Bits	Name	Access	Description	Reset
31:0	PRESENT_VUL_HS_SDR12	RO	Present Value Register for High-speed and SDR12	

21.4.4.25 PRESENT_VUL_SDR25_SDR50

Present Value Register for SDR25 and SDR50

Table 21.129: PRESENT_VUL_SDR25_SDR50, Offset Address: 0x068

Bits	Name	Access	Description	Reset
31:0	PRESENT_VUL_SDR25_SDR50	RO	Present Value Register for SDR25 and SDR50	

21.4.4.26 PRESENT_VUL_SDR104_DDR50

Present Value Register for SDR104 and DDR50

Table 21.130: PRESENT_VUL_SDR104_DDR50, Offset Address: 0x06c

Bits	Name	Access	Description	Reset
31:0	PRESENT_VUL_SDR104_DDR50	RO	Present Value Register for SDR104 and DDR50	

21.4.4.27 SLOT_INT_AND_HOST_VER

Slot Interrupt Status and Host Controller Version Register

Table 21.131: SLOT_INT_AND_HOST_VER, Offset Address: 0x0fc

Bits	Name	Access	Description	Reset
7:0	INT_SLOT	RO	Interrupt Signal for Each Slot	
15:8	Reserved			
23:16	SPEC_VER	RO	Specification Version Number 00h : SD Host 1.00 01h : SD Host 2.00 02h : SD Host 3.00 03h : SD Host 4.00 04h : SD Host 4.10 05h : SD Host 4.20	
31:24	VENDOR_VER	RO	Verdor Version Number	

21.4.4.28 EMMC_CTRL

MSHC Control register

Table 21.132: EMMC_CTRL, Offset Address: 0x200

Bits	Name	Access	Description	Reset
0	EMMC_FUNC_EN	R/W	eMMC Card present	0x0
1	LATANCY_1T	R/W	Latancy 1t for cmd in	0x1
2	CLK_FREE_EN	R/W	Internal clock gating disable control	0x0
3	DISABLE_DATA_CRC_CHK	R/W	Disable Data CRC Check	0x0
7:4	Reserved			
8	EMMC_RSTN	R/W	EMMC Device Reset Signal control	0x1
9	EMMC_RSTN_OEN	R/W	Output Enable control for EMMC Device Reset Signal PAD	0x1
11:10	Reserved			
12	CQE_ALGO_SEL	R/W	Scheduler algorithm selected for execution 1 : First come First serve (FCFS_ONLY) 0 : Priority based reordering with FCFS (PRI_REORDER_PLUS_FCFS)	0x0
13	CQE_PREFETCH_DISABLE	R/W	Enable or Disable CQE's PREFETCH Feature 1 : Disable 0 : Enable	0x0
15:14	Reserved			
16	timer_clk_sel	R/W	timer clock source selection 1 : 32K 0 : 100K	0x0
31:17	Reserved			

21.4.4.29 EMMC_BOOT_CTL

eMMC Boot Control Register

Table 21.133: EMMC_BOOT_CTL, Offset Address: 0x204

Bits	Name	Access	Description	Reset
0	BOOT_MODE_ENABLE	R/W	Mandatory Boot Enable	0x0
1	BOOT_ACK_ENABLE	R/W	Boot Ack Enable	0x0
3:2	Reserved			
7:4	BOOT_TOUT_CNT	R/W	Boot Ack Timeout Counter Value	0x0
8	VALIDATE_BOOT	W	Validate Mandatory Boot Enable Bit	
31:9	Reserved			

21.4.4.30 CDET_TOUT_CTL

Card Detect Control Register

Table 21.134: CDET_TOUT_CTL, Offset Address: 0x208

Bits	Name	Access	Description	Reset
15:0	CDET_DEBUUNCE_CNT	R/W	card detect debounce counter	0x 000F
31:16	Reserved			

21.4.4.31 MBIU_CTRL

MBIU Control register

Table 21.135: MBIU_CTRL, Offset Address: 0x20c

Bits	Name	Access	Description	Reset
0	UNDEFL_INCR_EN	R/W	Undefined INCR Burst	0x1
1	BURST_INCR4_EN	R/W	INCR4 Burst	0x1
2	BURST_INCR8_EN	R/W	INCR8 Burst	0x1
3	BURST_INCR16_EN	R/W	INCR16 Burst	0x1
31:4	Reserved			

21.4.4.32 PHY_TX_RX_DLY

MBIU Control register

Table 21.136: PHY_TX_RX_DLY, Offset Address: 0x240

Bits	Name	Access	Description	Reset
6:0	PHY_TX_DLY	R/W	PHY tx delay line phase selection	0x0
7	Reserved			
9:8	PHY_TX_SRC	R/W	PHY tx delay line clock source selection 2'b00 : clk_tx 2'b01 : inverse of clk_tx 2'b1x : reserved	0x0
10	PHY_TX_EVEN_ODD	R/W	PHY tx delay line clock source selection	0x0
15:11	Reserved			
22:16	PHY_RX_DLY	R/W	PHY rx delay line phase selection 2'b00 : clk_tx 2'b01 : inverse of clk_tx 2'b1x : reserved	0x0
23	Reserved			
25:24	PHY_RX_SRC	R/W	PHY rx delay line clock source selection	0x0
26	PHY_RX_EVEN_ODD	R/W	PHY rx delay line clock source selection	0x0
31:27	Reserved			

21.4.4.33 PHY_DS_DLY

PHY DS delay line register

Table 21.137: PHY_DS_DLY, Offset Address: 0x244

Bits	Name	Access	Description	Reset
6:0	PHY_DS_DLY	R/W	PHY DS delay line phase selection	0x0
7	Reserved			
9:8	PHY_DS_SRC	R/W	PHY DS delay line clock source selection	0x0
10	PHY_DS_EVEN_ODD	R/W	PHY DS delay line clock source selection	0x0
31:11	Reserved			

21.4.4.34 PHY_DLY_STS

PHY delay line status register

Table 21.138: PHY_DLY_STS, Offset Address: 0x248

Bits	Name	Access	Description	Reset
0	PHY_TX_LEAD_LAG	RO	PHY tx delay line lead or lag flag	
1	PHY_RX_LEAD_LAG	RO	PHY rx delay line lead or lag flag	
2	PHY_DS_LEAD_LAG	RO	PHY ds delay line lead or lag flag	
31:3	Reserved			

21.4.4.35 PHY_CONFIG

PHY Configuration register

Table 21.139: PHY_CONFIG, Offset Address: 0x24c

Bits	Name	Access	Description	Reset
0	PHY_TX_BPS	R/W	PHY tx data path bypass enable 0 : Pipe enable 1 : Bypass	0x1
1	ADJ_TIMING_EN	R/W	Adjust bus timing enable	0x0
7:2	Reserved			
9:8	ADJ_NCR	R/W	Adjust NCR counter	0x0
11:10	ADJ_NCRC	R/W	Adjust NCRC counter	0x0
31:12	Reserved			

21.5 GPIO

The system includes 4 groups of GPIO (General Purpose Input/Output) under Active Domain, which are GPIO0 ~ GPIO3, and 1 group of GPIO under No-die Domain, RTCSYS_GPIO. Each group of GPIO provides 32 programmable input and output pins.

Note: In this manual, GPIOA is often used instead of GPIO0, GPIOB instead of GPIO1, GPIOC instead of GPIO2, and GPIOD instead of GPIO3.

The direction of each pin can be arbitrarily set as input or output, used to generate output signals for specific applications or collect input signals for specific applications. When set as an input pin, the GPIO can be used as an interrupt source; when set as an output pin, each GPIO can independently output 0 or 1.

GPIO can generate maskable interrupts based on the level or transition value of the input signal. The GPIOx_INTR_FLAG (x = 0 ~ 3) signal gives the interrupt controller an indication that an interrupt has occurred.

21.5.1 Features

The direction of each pin can be arbitrarily set as input or output.

- When set as an input pin, GPIO can be used as an interrupt source.
- When set as an output pin, each GPIO can output 0 or 1 independently.

21.5.2 Way of Working

21.5.2.1 Interface Reset

When the chip is powered on or the system is reset, the four GPIO modules will be reset at the same time, and the GPIO pins will be in the input state by default after reset.

21.5.2.2 General Purpose Input and Output

Each pin can be set as input or output arbitrarily. The steps are as follows:

- Step 1: Configure the register GPIO_SWPORTA_DDR, set whether the GPIO is used as input or output.
- Step 2: When configured as an input pin, read the GPIO_EXT_PORTA register to view the input signal value; when configured as an output pin, write the output value to the GPIO_SWPORTA_DR register to control the GPIO output level.

21.5.2.3 Interrupt Operation

Each GPIO can be used as an interrupt source, controlled through 9 registers such as GPIO_INTEN. Through these registers the user can select the interrupt source, interrupt level polarity and edge triggering characteristics.

When multiple GPIO interrupts occur at the same time, they will be aggregated into one interrupt for reporting (each of the 4 groups of GPIOs will have a collective interrupt flag reported).

The characteristics of the interrupt source and the interrupt trigger category are determined by the five registers GPIO_INTTYPE_LEVEL, GPIO_INT_POLARITY, GPIO_INTMASK, GPIO_DEBOUNCE, and GPIO_LS_SYNC.

The original status and masked status of the interrupt are read through GPIO_RAW_INTSTATUS and GPIO_INTSTATUS. The clearing of interrupt status can be controlled by setting GPIO_PORTA_EOI.

Each GPIO can support interrupts. The setting steps are as follows:

- Step 1: Configure the register GPIO_INTTYPE_LEVEL, select level trigger or edge trigger.
- Step 2: Configure the register GPIO_INT_POLARITY, select low level/high level trigger and falling edge/rising edge trigger.
- Step 3: Write 0xFFFFFFFF to the register GPIO_PORTA_EOI to clear the interrupt.
- Step 4: Configure the GPIO_INTEN register; enable the GPIO pin interrupt function.

21.5.3 GPIO Register Overview

The chip includes 4 groups of GPIO under Active Domain and 1 group of GPIO under No-die Domain. The base address is shown in table *Base addresses of GPIO modules*.

Table 21.140: Base addresses of GPIO modules

GPIO Module	Base Address
GPIO0	0x03020000
GPIO1	0x03021000
GPIO2	0x03022000
GPIO3	0x03023000
RTCSYS_GPIO	0x05021000

Table *GPIO Registers Overview* is the offset address and definition of the registers of the GPIO module (GPIO0) in group 1. GPIOs in other groups have the same register definitions.

Table 21.141: GPIO Registers Overview

Name	Address Offset	Description
GPIO_SWPORTA_DR	0x000	Port A data register
GPIO_SWPORTA_DDR	0x004	Port A data direction register
GPIO_INTEN	0x030	Interrupt enable register
GPIO_INTMASK	0x034	Interrupt mask register
GPIO_INTTYPE_LEVEL	0x038	Interrupt level register
GPIO_INT_POLARITY	0x03c	Interrupt polarity register
GPIO_INTSTATUS	0x040	Interrupt status of Port A
GPIO_RAW_INTSTATUS	0x044	Raw interrupt status of Port A (pre-masking)
GPIO_DEBOUNCE	0x048	Debounce enable register
GPIO_PORTA_EOI	0x04c	Port A clear interrupt register
GPIO_EXT_PORTA	0x050	Port A external port register
GPIO_LS_SYNC	0x060	Level-sensitive synchronization enable register

21.5.4 GPIO Register Description

21.5.4.1 GPIO_SWPORTA_DR

Table 21.142: GPIO_SWPORTA_DR, Offset Address: 0x000

Bits	Name	Access	Description	Reset
31:0	GPIO_SWPORTA_DR	R/W	Values written to this register are output on the I/O signals for Port A if the corresponding data direction bits for Port A are set to Output mode and the corresponding control bit for Port A is set to Software mode. The value read back is equal to the last value written to this register.	0x0

21.5.4.2 GPIO_SWPORTA_DDR

Table 21.143: GPIO_SWPORTA_DDR, Offset Address: 0x004

Bits	Name	Access	Description	Reset
31:0	GPIO_SWPORTA_DDR	R/W	Values written to this register independently control the direction of the corresponding data bit in Port A. The default direction can be configured as input or output after system reset through the GPIO_DFLT_DIR_A parameter. 0 - Input (default) 1 - Output	0x0

21.5.4.3 GPIO_INTEN

Table 21.144: GPIO_INTEN, Offset Address: 0x030

Bits	Name	Access	Description	Reset
31:0	GPIO_INTEN	R/W	Allows each bit of Port A to be configured for interrupts. By default the generation of interrupts is disabled. Whenever a 1 is written to a bit of this register, it configures the corresponding bit on Port A to become an interrupt; otherwise, Port A operates as a normal GPIO signal. Interrupts are disabled on the corresponding bits of Port A if the corresponding data direction register is set to Output or if Port A mode is set to Hardware. 0 - Configure Port A bit as normal GPIO signal (default) 1 - Configure Port A bit as interrupt	0x0

21.5.4.4 GPIO_INTMASK

Table 21.145: GPIO_INTMASK, Offset Address: 0x034

Bits	Name	Access	Description	Reset
31:0	GPIO_INTMASK	R/W	Controls whether an interrupt on Port A can create an interrupt for the interrupt controller by not masking it. By default, all interrupts bits are unmasked. Whenever a 1 is written to a bit in this register, it masks the interrupt generation capability for this signal; otherwise interrupts are allowed through. The unmasked status can be read as well as the resultant status after masking. 0 - Interrupt bits are unmasked (default) 1 - Mask interrupt	0x0

21.5.4.5 GPIO_INTTYPE_LEVEL

Table 21.146: GPIO_INTTYPE_LEVEL, Offset Address: 0x038

Bits	Name	Access	Description	Reset
31:0	GPIO_INTTYPE_LEVEL	R/W	Controls the type of interrupt that can occur on Port A. Whenever a 0 is written to a bit of this register, it configures the interrupt type to be level-sensitive; otherwise, it is edge-sensitive. 0 - Level-sensitive (default) 1 - Edge-sensitive	0x0

21.5.4.6 GPIO_INT_POLARITY

Table 21.147: GPIO_INT_POLARITY, Offset Address: 0x03c

Bits	Name	Access	Description	Reset
31:0	GPIO_INT_POLARITY	R/W	Controls the polarity of edge or level sensitivity that can occur on input of Port A. Whenever a 0 is written to a bit of this register, it configures the interrupt type to falling-edge or active-low sensitive; otherwise, it is rising-edge or active-high sensitive. 0 - Active-low (default) 1 - Active-high	0x0

21.5.4.7 GPIO_INTSTATUS

Table 21.148: GPIO_INTSTATUS, Offset Address: 0x040

Bits	Name	Access	Description	Reset
31:0	GPIO_INTSTATUS	RO	Interrupt status of Port A	

21.5.4.8 GPIO_RAW_INTSTATUS

Table 21.149: GPIO_RAW_INTSTATUS, Offset Address: 0x044

Bits	Name	Access	Description	Reset
31:0	GPIO_RAW_INTSTATUS	RO	Raw interrupt of status of Port A (premasking bits)	

21.5.4.9 GPIO_DEBOUNCE

Table 21.150: GPIO_DEBOUNCE, Offset Address: 0x048

Bits	Name	Access	Description	Reset
31:0	GPIO_DEBOUNCE	R/W	Controls whether an external signal that is the source of an interrupt needs to be debounced to remove any spurious glitches. Writing a 1 to a bit in this register enables the debouncing circuitry. A signal must be valid for two periods of an external clock before it is internally processed. 0 - No debounce (default) 1 - Enable debounce	0x0

21.5.4.10 GPIO_PORTA_EOI

Table 21.151: GPIO_PORTA_EOI, Offset Address: 0x04c

Bits	Name	Access	Description	Reset
31:0	GPIO_PORTA_EOI	R/W	Controls the clearing of edge type interrupts from Port A. When a 1 is written into a corresponding bit of this register, the interrupt is cleared. All interrupts are cleared when Port A is not configured for interrupts. 0 - No interrupt clear (default) 1 - Clear interrupt	0x0

21.5.4.11 GPIO_EXT_PORTA

Table 21.152: GPIO_EXT_PORTA, Offset Address: 0x050

Bits	Name	Access	Description	Reset
31:0	GPIO_EXT_PORTA	RO	When Port A is configured as Input, then reading this location reads the values on the signal. When the data direction of Port A is set as Output, reading this location reads the data register for Port A.	

21.5.4.12 GPIO_LS_SYNC

Table 21.153: GPIO_LS_SYNC, Offset Address: 0x060

Bits	Name	Access	Description	Reset
0	GPIO_LS_SYNC	R/W	[0] Synchronization level Writing a 1 to this register results in all level-sensitive interrupts being synchronized to pclk_intr. 0 - No synchronization to pclk_intr (default) 1 - Synchronize to pclk_intr	0x0
31:1	Reserved			

21.6 USB DRD

21.6.1 Overview

The function of USB DRD is to play the role of Host or Device respectively, which can be changed through software settings. The transmission protocol complies with the USB 2.0 specification, and the maximum transfer rate can reach more than 40MB/s; the main operating modes of Host/Device are Scatter gather DMA transfer (scatter gather DMA), details will be described in the Host and Device chapters respectively; the functions of USB DRD are briefly listed as follows:

- Control Transfer
- Bulk Transfer
- Isochronous Transfer
- Host can connect to USB Hub and supports Interrupt Transfer
- Passed the USB Electrical Characteristics Test (USBET), with good signal quality and compatibility

21.6.2 Function Description

21.6.2.1 System Block Diagram

The picture below shows the system block diagram inside the USB DRD:

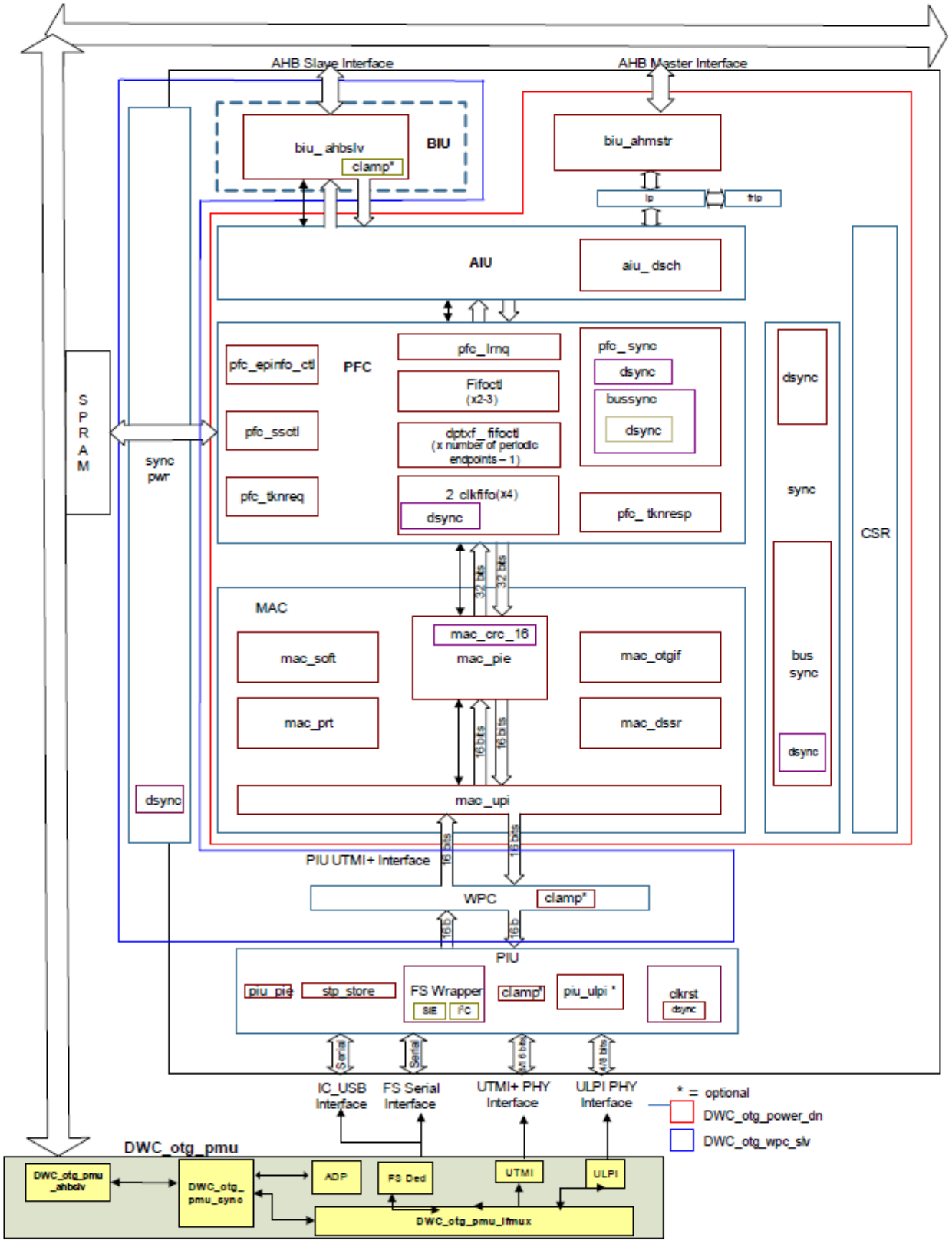


Diagram 21.42: USB DRD internal system block diagram

21.6.2.2 Functional Features

The functions of USB DRD are briefly listed as follows:

- Comply with USB2.0 transmission protocol specification.
- Backwards compatible with USB1.1 transmission protocol specification.
- Supports HS/FS/LS three speed modes.
- Support Host or Device function.
- Supports four transmission types specified by the USB transmission protocol: control transmission, batch transmission, real-time transmission, and interrupt transmission.
- Can be connected to a USB Hub to expand a single interface into multiple USB interfaces.
- Connect up to 127 Devices via USB Hub expansion.
- Support USB2.0 sleep/resume (suspend/resume) power saving mode.
- Support HID devices such as keyboard and mouse.
- Device mode is mainly used for downloading and updating internal software, and can also be used for other functions, such as data transmission.
- The maximum transfer rate can reach more than 40MB/s.

21.6.3 USBC Function and Register Description

21.6.3.1 USBC Functional Description

USB DRD can switch between Host or Device functions. You can choose to use either one, but it cannot work at the same time. Its function selection and management are controlled by the USBC block; in addition, Host and device are triggered by some events and interrupts on the Serial Bus. , the register will also be placed in this block.

21.6.3.2 USBC Register Overview

Table 21.154: USBC Registers Overview

Name	Address Offset	Description
GOTGCTL	0x000	Control and Status Register
GOTGINT	0x004	Interrupt Register
GAHBCFG	0x008	AHB Configuration Register
GUSBCFG	0x00c	USB Configuration Register
GRSTCTL	0x010	Reset Register
GINTSTS	0x014	Interrupt Status Register
GINTMSK	0x018	Interrupt Mask Register
GUID	0x03c	User ID Register
GLPMCFG	0x054	Core LPM Configuration Register
GPWRDN	0x058	Power Down Register

21.6.3.3 USBC Register Description

The memory addressing location of USBC is 0x0434_0000, which is represented by USBC_BASE_ADDR in this article. If it is to be read and written, the real addressing location in the memory space will be represented by USBC_BASE_ADDR + Offset; each register has Its corresponding relative addressing (Offset), the details are described below.

GOTGCTL

Control and Status Register

Table 21.155: GOTGCTL, Offset Address: 0x000

Bits	Name	Access	Description	Reset
0	SesReqScs	RO	Mode: Device only Session Request Success (SesReqScs) The core sets this bit when a session request initiation is successful. • 1'b0: Session request failure • 1'b1: Session request success Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	
1	SesReq	R/W	Mode: SRP-capable device Session Request (SesReq) The application sets this bit to initiate a session request on the USB. The application can clear this bit by writing a 0 when the Host Negotiation Success Status Change bit in the OTG Interrupt register (GOTGINT.HstNegSucStsChng) is SET. The core clears this bit when the HstNegSucStsChng bit is cleared. If you use the USB 1.1 Full-Speed Serial Transceiver interface to initiate the session request, the application must wait until the VBUS discharges to 0.2 V, after the B-Session Valid bit in this register (GOTGCTL.BSesVld) is cleared. This discharge time varies between different PHYs and can be obtained from the PHY vendor. • 1'b0: No session request • 1'b1: Session request Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	0x0
2	VbvalidOvEn	R/W	Mode: Host only VBUS Valid Override Enable (VbvalidOvEn) This bit is used to enable/disable the software to override the Bvalid signal using the GOTGCTL.VbvalidOvVal. • 1'b1: Internally Bvalid received from the PHY is overridden with GOTGCTL.VbvalidOvVal. • 1'b0: Override is disabled and bvalid signal from the respective PHY selected is used internally by the core. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0

To be continued ……

Table 21.156: GOTGCTL, Offset Address: 0x000 (continued)

Bits	Name	Access	Description	Reset
3	VbvalidOvVal	R/W	Mode: Host only VBUS Valid Override Value (VbvalidOvVal) This bit is used to set Override value for vbus-valid signal when GOTGCTL.VbvalidOvEn is set. • 1'b0: vbusvalid value is 1'b0 when GOT-GCTL.VbvalidOvEn =1 • 1'b1: vbusvalid value is 1'b1 when GOT-GCTL.VbvalidOvEn =1	0x0
4	AvalidOvEn	R/W	Mode: Host only A-Peripheral Session Valid Override Enable (AvalidOvEn) This bit is used to enable/disable the software to override the Avalid signal using the GOTGCTL.AvalidOvVal. • 1'b1: Internally Avalid received from the PHY is overridden with GOT-GCTL.AvalidOvVal. • 1'b0: Override is disabled and avalid signal from the respective PHY selected is used internally by the core.	0x0
5	AvalidOvVal	R/W	Mode: Host only A-Peripheral Session Valid Override Value (AvalidOvVal) This bit is used to set Override value for Avalid signal when GOTGCTL.AvalidOvEn is set. • 1'b0: Avalid value is 1'b0 when GOT-GCTL.AvalidOvEn =1 • 1'b1: Avalid value is 1'b1 when GOT-GCTL.AvalidOvEn =1	0x0
6	BvalidOvEn	R/W	Mode: Device only B-Peripheral Session Valid Override Enable (BvalidOvEn) This bit is used to enable/disable the software to override the Bvalid signal using the GOT-GCTL.BvalidOvVal. • 1'b1: Internally Bvalid receiv ed from the PHY is overridden with GOT-GCTL.BvalidOvVal. • 1'b0: Override is disabled and bvalid signal from the respective PHY selected is used internally by the force	0x0

To be continued ……

Table 21.157: GOTGCTL, Offset Address: 0x000 (continued)

Bits	Name	Access	Description	Reset
7	BvalidOvVal	R/W	Mode: Device only B-Peripheral Session Valid Override Value (BvalidOvVal) This bit is used to set Override value for Bvalid signal when GOTGCTL.BvalidOvEn is set. • 1'b0: Bvalid value is 1'b0 when GOTGCTL.BvalidOvEn =1 • 1'b1: Bvalid value is 1'b1 when GOTGCTL.BvalidOvEn =1	0x0
8	HstNegScs	RO	Mode: HNP-capable device Host Negotiation Success (HstNegScs) The core sets this bit when host negotiation is successful. The core clears this bit when the HNP Request (HNPReq) bit in this register is set. • 1'b0: Host negotiation failure • 1'b1: Host negotiation success	
9	HNPReq	R/W	Mode: HNP Capable OTG Device HNP Request (HNPReq) The application sets this bit to initiate an HNP request to the connected USB host. The application can clear this bit by writing a 0 when the Host Negotiation Success Status Change bit in the OTG Interrupt register (GOTGINT.HstNegSucStsChng) is set. The core clears this bit when the HstNegSucStsChng bit is cleared. • 1'b0: No HNP request • 1'b1: HNP request	0x0
10	HstSetHNPEn	R/W	Mode: HNP Capable OTG Host Host Set HNP Enable (HstSetHNPEn) The application sets this bit when it has successfully enabled HNP (using the SetFeature.SetHNPEnable command) on the connected device. • 1'b0: Host Set HNP is not enab led • 1'b1: Host Set HNP is enabled	0x0

To be continued ……

Table 21.158: GOTGCTL, Offset Address: 0x000 (continued)

Bits	Name	Access	Description	Reset
11	DevHNPEn	R/W	Mode: HNP Capable OTG Device Device HNP Enabled (DevHNPEn) The application sets this bit when it successfully receives aSetFeature.SetHNPEnable command from the connected USB host. • 1'b0: HNP is not enabled in the application • 1'b1: HNP is enabled in the application	0x0
12	EHEn	R/W	Embedded Host Enable It is used to select between OTG A Device state Machine andEmbedded Host state machine. • 1'b1: Embedded Host State Mach ine is selected • 1'b0: OTG A Device state machi ne is selected Note: This field is valid only in SRP-Capable OTG Mode (OTG_MODE=0,1)	0x0
14:13	Reserved_00_14_13	RO	Reserved for future use.	
15	DbnceFltrBypass	R/W	Mode: Host and Device Debounce Filter Bypass Bypass Debounce filters for avalid, bvalid, vbusvalid, sessend, and iddig signals when enabled. • 1'b0: Disabled • 1'b1: Enabled	0x0
16	ConIDSts	RO	Mode: Host and Device Connector ID Status (ConIDSts) Indicates the connector ID status on a connect event. • 1'b0: The DWC_otg core is in A-Device mode • 1'b1: The DWC_otg core is in B-Device mode	
17	DbncTime	RO	Mode: Host only Long/Short Debounce Time (DbncTime) Indicates the debounce time of a detected connection. • 1'b0: Long debounce time, used for physical connections (100 ms + 2.5 us) • 1'b1: Short debounce time, use d for soft connections (2.5 us)	

To be continued ……

Table 21.159: GOTGCTL, Offset Address: 0x000 (continued)

Bits	Name	Access	Description	Reset
18	ASesVld	RO	Mode: Host only A-Session Valid (ASesVld) Indicates the Host mode transceiver status. • 1'b0: A-session is not valid • 1'b1: A-session is valid	
19	BSesVld	RO	Mode: Device only B-Session Valid (BSesVld) Indicates the Device mode transceiver status. • 1'b0: B-session is not valid. • 1'b1: B-session is valid. In OTG mode, you can use this bit to determine if the device is connected or disconnected.	
20	OTGVer	R/W	OTG Version (OTGVer) Indicates the OTG revision. • 1'b0: OTG Version 1.3. In this version the core supports Data line pulsing and VBus pulsing for SRP. • 1'b1: OTG Version 2.0. In this version the core supports only Data line pulsing for SRP.	0x0
21	CurMod_operation	RO	Current Mode of Operation (CurMod) Mode: Host and Device Indicates the current mode. • 1'b0: Device mode • 1'b1: Host mode	
26:22	Mult-ValIdBC_operation	RO	Multi Valued ID pin (MultValIdBC) Mode: Host and Device Battery Charger ACA inputs in the following order: • Bit 26 - rid_float • Bit 25 - rid_gnd • Bit 24 - rid_a • Bit 23 - rid_b • Bit 22 - rid_c	
27	ChirpEn	RO	Chirp On Enable (ChirpEn) Mode: Device Only This bit when programmed to 1'b1 results in the core asserting chirp_on before sending an actual Chirp "K" signal on USB. This bit is present only if OTG_BC_SUPPORT = 1. If OTG_BC_SUPPORT!=1, this bit is a reserved bit.	
31:28	Reserved_00_31_28	RO	Reserved for future use.	

GOTGINT

Interrupt Register

Table 21.160: GOTGINT, Offset Address: 0x004

Bits	Name	Access	Description	Reset
1:0	Reserved_04_1_0	RO	Reserved for future use. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	
2	SesEndDet	RWC	Write Behavior: One to clear Mode: Host and Device Session End Detected (SesEndDet) The core sets this bit when the utmiotg_bvalid signal is deasserted.This bit can be set only by the core and the application should write 1 to clear it.	
7:3	Reserved_04_7_3	RO	Reserved for future use.	
8	SesReqSucStsChng	RWC	Write Behavior: One to clear Mode: Host and Device Session Request Success Status Change (SesReqSucStsChng) The core sets this bit on the success or failure of a session request. The application must read the Session Request Success bit in the OTG Control and Status register (GOTGCTL.SesReqScs) to check for success or failure.This bit can be set only by the core and the application should write 1 to clear it.	
9	HstNegSucStsChng	RWC	Write Behavior: One to clear Mode: Host and Device Host Negotiation Success Status Change (HstNegSucStsChng) The core sets this bit on the success or failure of a USB host negotiation request. The application must read the Host Negotiation Success bit of the OTG Control and Status register (GOTGCTL.HstNegScs) to check for success or failure.This bit can be set only by the core and the application should write 1 to clear it.	
16:10	Reserved_04_16_10	RO	Reserved for future use.	

To be continued ……

Table 21.161: GOTGINT, Offset Address: 0x004 (continued)

Bits	Name	Access	Description	Reset
17	HstNegDet	RWC	Write Behavior: One to clear Mode: Host and Device Host Negotiation Detected (HstNegDet) The core sets this bit when it detects a host negotiation request on the USB.This bit can be set only by the core and the application should write 1 to clear it.	
18	ADevTOUTChg	RWC	Write Behavior: One to clear Mode: Host and Device A-Device Timeout Change (ADevTOUTChg) The core sets this bit to indicate that the A-device has timed out while waiting for the B-device to connect.This bit can be set only by the core and the application should write 1 to clear it.	
19	DbnceDone	RWC	Write Behavior: One to clear Mode: Host only Debounce Done (DbnceDone) The core sets this bit when the debounce is completed after the device connect. The application can start driving USB reset after seeing this interrupt. This bit is only valid when the HNP Capable or SRP Capable bit is set in the Core USB Configuration register (GUSBCFG.HNPCap or GUSBCFG.SRPCap, respectively).This bit can be set only by the core and the application should write 1 to clear it.	
20	MultValIpChng	RWC	Write Behavior: One to clear This bit when set indicates that there is a change in the value of at least one ACA pin value. This bit is present only if OTG_BC_SUPPORT=1, otherwise it is reserved.	
31:21	Reserved_04_31_21	RO	Reserved for future use. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	

GAHBCFG

AHB Configuration Register

Table 21.162: GAHBCFG, Offset Address: 0x008

Bits	Name	Access	Description	Reset
0	GlblIntrMsk	R/W	Mode: Host and device Global Interrupt Mask (GlblIntrMsk) The application uses this bit to mask or unmask the interrupt line assertion to itself. Irrespective of this bit's setting, the interrupt status registers are updated by the core. • 1'b0: Mask the interrupt assertion to the application. • 1'b1: Unmask the interrupt assertion to the application. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	0x0
4:1	HBstLen	R/W	Mode: Host and device Burst Length/Type (HBstLen) This field is used in both External and Internal DMA modes. In External DMA mode, these bits appear on dma_burst[3:0] ports, which can be used by an external wrapper to interface the External DMA Controller interface to Synopsys DW_ahb_dmac or ARM PrimeCell. External DMA Mode defines the DMA burst length in terms of 32-bit words: • 4'b0000: 1 word • 4'b0001: 4 words • 4'b0010: 8 words • 4'b0011: 16 words • 4'b0100: 32 words • 4'b0101: 64 words • 4'b0110: 128 words • 4'b0111: 256 words • Others: Reserved Internal DMA Mode-AHB Master burst type: • 4'b0000 Single • 4'b0001 INCR 4'b0011 • INCR4 4'b0101 • INCR8 4'b0111 • INCR16 • Others: Reserved	0x0

To be continued ……

Table 21.163: GAHBCFG, Offset Address: 0x008 (continued)

Bits	Name	Access	Description	Reset
5	DMAEn	R/W	Mode: Host and device DMA Enable (DMAEn) • 1'b0: Core operates in Slave mode • 1'b1: Core operates in a DMA mode This bit is always 0 when Slave-Only mode has been selected.	0x0
6	Reserved_08_6	RO	Reserved for future use. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	
7	NPTxFEmpLvl	R/W	Mode: Host and device Non-Periodic TxFIFO Empty Level (NPTxFEmpLvl) This bit is used only in Slave mode. In host mode and with Shared FIFO with device mode, this bit indicates when the Non-Periodic TxFIFO Empty Interrupt bit in the Core Interrupt register (GINTSTS.NPTxFEmp) is triggered. With dedicated FIFO in device mode, this bit indicates when IN endpoint Transmit FIFO empty interrupt (DIEPINTn.TxFEmp) is triggered. Host mode and with Shared FIFO with device mode: • 1'b0: GINTSTS.NPTxFEmp interrupt indicates that the Non- Periodic TxFIFO is half empty • 1'b1: GINTSTS.NPTxFEmp interrupt indicates that the Non- Periodic TxFIFO is completely empty Dedicated FIFO in device mode: • 1'b0: DIEPINTn.TxFEmp interrupt indicates that the IN Endpoint TxFIFO is half empty • 1'b1: DIEPINTn.TxFEmp interrupt indicates that the IN Endpoint TxFIFO is completely empty	0x0

To be continued ……

Table 21.164: GAHBCFG, Offset Address: 0x008 (continued)

Bits	Name	Access	Description	Reset
8	PTxFEmpLvl	R/W	Mode: Host only Periodic TxFIFO Empty Level (PTxFEmpLvl) Indicates when the Periodic TxFIFO Empty Interrupt bit in the Core Interrupt register (GINTSTS.PTxFEmp) is triggered. This bit is used only in Slave mode. • 1'b0: GINTSTS.PTxFEmp interrupt indicates that the Periodic TxFIFO is half empty • 1'b1: GINTSTS.PTxFEmp interrupt indicates that the Periodic TxFIFO is completely empty	0x0
20:9	Reserved_08_20_9	RO	Reserved for future use. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	
21	RemMemSupp	R/W	Mode: Host and Device Remote Memory Support (RemMemSupp) This bit is programmed to enable the functionality to wait for the system DMA Done Signal for the DMA Write Transfers. • GAHBCFG.RemMemSupp=1 The int_dma_req output signal is asserted when HSOTG DMA starts write transfer to the external memory. When the core is done with the Transfers it asserts int_dma_done signal to flag the completion of DMA writes from HSOTG. The core then waits for sys_dma_done signal from the system to proceed further and complete the Data Transfer corresponding to a particular Channel/Endpoint. • GAHBCFG.RemMemSupp=0 The int_dma_req and int_dma_done signals are not asserted and the core proceeds with the assertion of the XferComp interrupt as soon as the DMA write transfer is done at the HSOTG Core Boundary and it doesn't wait for the sys_dma_done signal to complete the DATA transfers.	0x0

To be continued ……

Table 21.165: GAHBCFG, Offset Address: 0x008 (continued)

Bits	Name	Access	Description	Reset
22	NotiAllDmaWrit	R/W	Mode: Host and Device Notify all DMA Write Transactions (NotiAllDmaWrit) This bit is programmed to enable the System DMA Done functionality for all the DMA write Transactions corresponding to the Channel/Endpoint. This bit is valid only when GAHBCFG.RemMemSupp is set to 1. • GAHBCFG.NotiAllDmaWrit = 1 DWC_otg core asserts int_dma_req for all the DMA write transactions on the AHB interface along with int_dma_done, chep_last_transact and chep_number signal informations. The core waits for sys_dma_done signal for all the DMA write transactions in order to complete the transfer of a particular Channel/Endpoint. • GAHBCFG.NotiAllDmaWrit = 0 DWC_otg core asserts int_dma_req signal only for the last transaction of DMA write transfer corresponding to a particular Channel/Endpoint. Similarly, the core waits for sys_dma_done signal only for that transaction of DMA write to complete the transfer of a particular Channel/Endpoint.	0x0
23	AHBSingle	R/W	Mode: Host and Device AHBSingleSupport (AHBSingle) This bit when programmed supports Single transfers for the remaining data in a transfer when the DWC_otg core is operating in DMA mode. • 1'b0: This is the default mode . When this bit is set to 1'b0, the remaining data in the transfer is sent using INCR burst size. • 1'b1: When set to 1'b1, the remaining data in a transfer is sent using Single burst size. Note: If this feature is enabled, the AHB RETRY and SPLIT transfers still have INCR burst type. Enable this feature when the AHB Slave connected to the DWC_otg core does not support INCR burst (and when Split, and Retry transactions are not being used in the bus.)	0x0

To be continued ……

Table 21.166: GAHBCFG, Offset Address: 0x008 (continued)

Bits	Name	Access	Description	Reset
24	InvDescEndianness	R/W	Mode: Host and Device Inverse Descriptor Endianness • 1'b0: Descriptor endianness is similar to the AHB Master endianness • 1'b1: – If the AHB Master endianness is Big Endian, the Descriptor Endianness is Little Endian. – If the AHB Master endianness is Little Endian, the Descriptor Endianness is Big Endian.	0x0
31:25	Reserved_08_31_25	RO	Reserved for future use. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	

GUSBCFG

USB Configuration Register

Table 21.167: GUSBCFG, Offset Address: 0x00c

Bits	Name	Access	Description	Reset
2:0	TOutCal	R/W	Mode: Host and Device HS/FS Timeout Calibration (TOutCal) The number of PHY clocks that the application programs in this field is added to the high-speed/full-speed interpacket timeout duration in the core to account for any additional delays introduced by the PHY. This can be required, because the delay introduced by the PHY in generating the linestate condition can vary from one PHY to another. The USB standard timeout value for high-speed operation is 736 to 816 (inclusive) bit times. The USB standard timeout value for fullspeed operation is 16 to 18 (inclusive) bit times. The application must program this field based on the speed of enumeration. The number of bit times added per PHY clock are: High-speed operation: • One 30-MHz PHY clock = 16 bit times • One 60-MHz PHY clock = 8 bit times Full-speed operation: • One 30-MHz PHY clock = 0.4 bit times • One 60-MHz PHY clock = 0.2 bit times • One 48-MHz PHY clock = 0.25 bit times Using the HS as an example, if you set ToutCal to '001' you add one 30MHz PHY clock or 16 bit times. If you set ToutCal to '010' you add two 30MHz PHY clocks or 32 bit times, and so on. The 3 bits allow you to add up to 7 PHY clocks, and the number of bit times depend on the speed, and the PHY clock you are using. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0

To be continued ……

Table 21.168: GUSBCFG, Offset Address: 0x00c (continued)

Bits	Name	Access	Description	Reset
3	PHYIf	RO	Mode: Host and Device PHY Interface (PHYIf) The application uses this bit to configure the core to support a UTMI+ PHY with an 8- or 16-bit interface. When a ULPI PHY is chosen, this must be set to 8-bit mode. • 1'b0: 8 bits • 1'b1: 16 bits This bit is writable only If UTMI+ and ULPI were selected. Otherwise, this bit returns the value for the power-on interface selected during configuration. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	
4	ULPI_UTMI_Sel	R/W	Mode: Host and Device ULPI or UTMI+ Select (ULPI_UTMI_Sel) The application uses this bit to select either a UTMI+ interface or ULPI Interface. • 1'b0: UTMI+ Interface • 1'b1: ULPI Interface This bit is writable only If UTMI+ and ULPI was specified for High- Speed PHY Interface(s). Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0
5	FSIntf	R/W	Mode: Host and Device Full-Speed Serial Interface Select (FSIntf) The application uses this bit to select either a unidirectional or bidirectional USB 1.1 full-speed serial transceiver interface. • 1'b0: 6-pin unidirectional full-speed serial interface • 1'b1: 3-pin bidirectional full-speed serial interface If a USB 1.1 Full-Speed Serial Transceiver interface was not selected, this bit is always 0, with Read Only access. If a USB 1.1 FS interface was selected, then the application can set this bit to select between the 3- and 6-pin interfaces, and access is Read and Write.	0x0

To be continued ……

Table 21.169: GUSBCFG, Offset Address: 0x00c (continued)

Bits	Name	Access	Description	Reset
6	PHYSel	R/W	Mode: Host and Device USB 2.0 High-Speed PHY or USB 1.1 Full-Speed Serial Transceiver Select (PHYSel) The application uses this bit to select either a high-speed UTMI+ or ULPI PHY, or a full-speed transceiver. • 1'b0: USB 2.0 high-speed UTMI+ or ULPI PHY • 1'b1: USB 1.1 full-speed seria l transceiver If a USB 1.1 Full-Speed Serial Transceiver interface was not selected, this bit is always 0, with Read Only access. If a high-speed PHY interface was not selected, this bit is always 1, with Read Only access. If both interface types were selected (parameters have non-zero values), the application uses this bit to select which interface is active, and access is Read and Write.	0x0
7	DDRSel	R/W	Mode: Host and Device ULPI DDR Select (DDRSel) The application uses this bit to select a Single Data Rate (SDR) or Double Data Rate (DDR) or ULPI interface. • 1'b0: Single Data Rate ULPI Interface, with 8-bit-wide data bus • 1'b1: Double Data Rate ULPI Interface, with 4-bit-wide data bus This bit is valid only when OTG_HSPHY_INTERFACE = 2 or 3.	0x0
8	SRPCap	R/W	Mode: Host and Device SRP-Capable (SRPCap) The application uses this bit to control the DWC_otg core SRP capabilities. If the core operates as a non-SRP-capable B-device, it cannot request the connected A-device (host) to activate VBUS and start a session. • 1'b0: SRP capability is not enabled. • 1'b1: SRP capability is enabled. This bit is writable only if an SRP mode was specified for Mode of Operation in coreConsultant (parameter OTG_MODE). Otherwise, reads return 0. If SRP functionality is disabled by the software, the OTG signals on the PHY domain must be tied to the appropriate values.	0x0

To be continued ……

Table 21.170: GUSBCFG, Offset Address: 0x00c (continued)

Bits	Name	Access	Description	Reset
9	HNPCap	R/W	Mode: Host and Device HNP-Capable (HNPCap) The application uses this bit to control the DWC_otg core's HNP capabilities. • 1'b0: HNP capability is not enabled. • 1'b1: HNP capability is enabled. This bit is writable only if an HNP mode was specified for Mode of Operation in coreConsultant (parameter OTG_MODE). Otherwise, reads return 0. If HNP functionality is disabled by the software, the OTG signals on the PHY domain must be tied to the appropriate values.	0x0
13:10	USBTrdTim	R/W	Mode: Device only USB Turnaround Time (USBTrdTim) Sets the turnaround time in PHY clocks. Specifies the response time for a MAC request to the Packet FIFO Controller (PFC) to fetch data from the DFIFO (SPRAM). This must be programmed to • 4'h5: When the MAC interface is 16-bit UTMI+. • 4'h9: When the MAC interface is 8-bit UTMI+. Note: The values above are calculated for the minimum AHB frequency of 30 MHz. USB turnaround time is critical for certification where long cables and 5-Hubs are used, so If you need the AHB to run at less than 30 MHz, and If USB turnaround time is not critical, these bits can be programmed to a larger value.	0x5
14	Reserved_0C_14	RO	Reserved for future use.	

To be continued ……

Table 21.171: GUSBCFG, Offset Address: 0x00c (continued)

Bits	Name	Access	Description	Reset
15	PhyLPwrClkSel	R/W	Mode: Host and Device PHY Low-Power Clock Select (PhyLPwrClkSel) Selects either 480-MHz or 48-MHz (low-power) PHY mode. In FS and LS modes, the PHY can usually operate on a 48-MHz clock to save power. • 1'b0: 480-MHz Internal PLL clock • 1'b1: 48-MHz External Clock In 480 MHz mode, the UTMI interface operates at either 60 or 30- MHz, depending upon whether 8- or 16-bit data width is selected. In 48-MHz mode, the UTMI interface operates at 48 MHz in FS mode and at either 48 or 6 MHz in LS mode (depending on the PHY vendor). This bit drives the utmi_fsls_low_power core output signal, and is valid only For UTMI+ PHYs.	0x0
16	OtgI2CSel	R/W	Mode: Host and Device UTMIFS or I2C Interface Select (OtgI2CSel) The application uses this bit to select the I2C interface. • 1'b0: UTMI USB 1.1 Full-Speed interface for OTG signals • 1'b1: I2C interface for OTG signals This bit is writable only if I2C and UTMIFS were specified for Enable I2C Interface? in coreConsultant (parameter OTG_I2C_INTERFACE = 2). Otherwise, reads return 0.	0x0
17	ULPIFsLs	R/W	Mode: Host and Device ULPI FS/LS Select (ULPIFsLs) The application uses this bit to select the FS/LS serial interface for the ULPI PHY. This bit is valid only when the FS serial transceiver is selected on the ULPI PHY. • 1'b0: ULPI interface • 1'b1: ULPI FS/LS serial interface (Valid only when RTL parameters OTG_HSPHY_INTERFACE = 2 or 3 and OTG_FSPHY_INTERFACE = 1, 2, or 3) Before setting this bit, the application needs to ensure that GUSBCFG.ULPI_UTMI_SEL = 1'b1.	0x0

To be continued ……

Table 21.172: GUSBCFG, Offset Address: 0x00c (continued)

Bits	Name	Access	Description	Reset
18	ULP IAutoRes	R/W	Mode: Host and Device ULPI Auto Resume (ULPIAutoRes) This bit sets the AutoResume bit in the Interface Control register on the ULPI PHY. • 1'b0: PHY does not use AutoResume feature. • 1'b1: PHY uses AutoResume feature. (Valid only when RTL parameter OTG_HSPHY_INTERFACE = 2 or 3)	0x0
19	ULPIClkSusM	R/W	Mode: Host and Device ULPI Clock SuspendM (ULPIClkSusM) This bit sets the ClockSuspendM bit in the Interface Control register on the ULPI PHY. This bit applies only in serial or carkit modes. • 1'b0: PHY powers down internal clock during suspend. • 1'b1: PHY does not power down internal clock. (Valid only when RTL parameter OTG_HSPHY_INTERFACE = 2 or 3)	0x0
20	ULPIExtVbusDrv	R/W	Mode: Host only ULPI External VBUS Drive (ULPIExtVbusDrv) This bit selects between internal or external supply to drive 5V on VBUS, in ULPI PHY. • 1'b0: PHY drives VBUS using internal charge pump (Default). • 1'b1: PHY drives VBUS using external supply. (Valid only when RTL parameter OTG_HSPHY_INTERFACE = 2 or 3)	0x0

To be continued ……

Table 21.173: GUSBCFG, Offset Address: 0x00c (continued)

Bits	Name	Access	Description	Reset
21	ULPIExtVbusIndicator	R/W	Mode: Host only ULPI External VBUS Indicator (ULPIExtVbusIndicator) This bit indicates to the ULPI PHY to use an external VBUS overcurrent indicator. • 1'b0: PHY uses internal VBUS valid comparator. • 1'b1: PHY uses external VBUS valid comparator. (Valid only when RTL parameter OTG_HSPHY_INTERFACE = 2 or 3)	0x0
22	TermSelDLPulse	R/W	Mode: Device only TermSel DLine Pulsing Selection (TermSelDLPulse) This bit selects utmi_termselect to drive data line pulse during SRP. • 1'b0: Data line pulsing using utmi_txvalid (Default). • 1'b1: Data line pulsing using utmi_termsel.	0x0
23	Complement	R/W	Mode: Host only Indicator Complement Controls the PHY to invert the ExternalVbusIndicator input signal, generating the Complement Output. For more information, refer to the ULPI Specification. • 1'b0: PHY does not invert ExternalVbusIndicator signal • 1'b1: PHY does invert ExternalVbusIndicator signal This bit is reserved and read-only when OTG_HSPHY_INTERFACE is set to 0 or 1.	0x0
24	Indicator	R/W	Mode: Host only Indicator Pass Through Controls whether the Complement Output is qualified with the Internal Vbus Valid comparator before being used in the Vbus State in the RX CMD. For more information, refer to the ULPI Specification. • 1'b0: Complement Output signal is qualified with the Internal VbusValid comparator. • 1'b1: Complement Output signal is not qualified with the Internal VbusValid comparator. This bit is reserved and read-only when OTG_HSPHY_INTERFACE is set to 0 or 1.	0x0

To be continued ……

Table 21.174: GUSBCFG, Offset Address: 0x00c (continued)

Bits	Name	Access	Description	Reset
25	ULPI	R/W	Mode: Host only ULPI Interface Protect Disable Controls circuitry built into the PHY For protecting the ULPI interface when the link tri-states STP and data. Any pull-ups or pull-downs employed by this feature can be disabled. For more information, refer to the ULPI Specification. • 1'b0: Enables the interface protect circuit • 1'b1: Disables the interface protect circuit This bit is reserved and read-only when OTG_HSPHY_INTERFACE is set to 0 or 1.	0x0
26	IC_USBCap	RO	Mode: Host and Device IC_USB-Capable (IC_USBCap) The application uses this bit to control the DWC_otg core's IC_USB capabilities. • 1'b0: IC_USB PHY Interface is not selected. • 1'b1: IC_USB PHY Interface is selected. This bit is writable only if OTG_ENABLE_IC_USB=1 and OTG_FSPHY_INTERFACE!=0. The reset value depends on the configuration parameter OTG_SELECT_IC_USB when OTG_ENABLE_IC_USB = 1. In all other cases, this bit is set to 1'b0 and the bit is read only.	
27	IC_USBTrafCtl	R/W	Mode: Device only IC_USB TrafficPullRemove Control (IC_USBTrafCtl) When this bit is set, pullup/pulldown resistors are detached from the USB during traffic signaling, per section 6.3.4 of the IC_USB specification. This bit is valid only when configuration parameter OTG_ENABLE_IC_USB = 1 and register field USBCFG.IC_USBCap is set to 1.	0x0
28	TxEndDelay	R/W	Mode: Device only Tx End Delay (TxEndDelay) Writing 1'b1 to this bit enables the core to follow the TxEndDelay timings as per UTMI+ specification 1.05 section 4.1.5 for opmode signal during remote wakeup. • 1'b0: Normal Mode. • 1'b1: Tx End delay.	0x0

To be continued ……

Table 21.175: GUSBCFG, Offset Address: 0x00c (continued)

Bits	Name	Access	Description	Reset
29	ForceHstMode	R/W	Mode: Host and device Force Host Mode (ForceHstMode) Writing a 1 to this bit forces the core to host mode irrespective of utmiotg_iddig input pin. • 1'b0: Normal Mode. • 1'b1: Force Host Mode. After setting the force bit, the application must wait at least 25 ms before the change to take effect. When the simulation is in scale down mode, waiting for 500 us is sufficient. This bit is valid only when OTG_MODE = 0, 1 or 2. In all other cases, this bit reads 0.	0x0
30	ForceDevMode	R/W	Mode: Host and device Force Device Mode (ForceDevMode) Writing a 1 to this bit forces the core to device mode irrespective of utmiotg_iddig input pin. • 1'b0: Normal Mode. • 1'b1: Force Device Mode. After setting the force bit, the application must wait at least 25 ms before the change to take effect. When the simulation is in scale down mode, waiting for 500 us is sufficient. This bit is valid only when OTG_MODE = 0, 1 or 2. In all other cases, this bit reads 0.	0x0
31	CorruptTxPkt	R/W	Mode: Host and device Corrupt Tx packet (CorruptTxPkt) This bit is for debug purposes only. Never set this bit to 1.The application should always write 1'b0 to this bit.	0x0

GRSTCTL

Reset Register

Table 21.176: GRSTCTL, Offset Address: 0x010

Bits	Name	Access	Description	Reset
0	CSftRst	RO	Write Behavior: One to set Mode: Host and Device Core Soft Reset (CSftRst) Resets the hclk and phy_clock domains as follows: • Clears the interrupts and all the CSR registers except the following register bits: – PCGCCTL.RstPdwnModule – PCGCCTL.GateHclk – PCGCCTL.PwrClmp – PCGCCTL.StopPPhyLPwrClkSelclk – GUSBCFG.PhyLPwrClkSel – GUSBCFG.DDRSel – GUSBCFG.PHYSel – GUSBCFG.FSIntf – GUSBCFG.ULPI_UTMI_Sel – GUSBCFG.PHYIf – GUSBCFG.TxEndDelay – GUSBCFG.TermSelDLPulse – GUSBCFG.ULPIClkSusM – GUSBCFG.ULPIAutoRes – GUSBCFG.ULPIFsLs – GGPIO – GPWRDN – GADPCTL – HCFG.FSLSPclkSel – DCFG.DevSpd – DCTL.SftDiscon • All module state machines (except the AHB Slave Unit) are reset to the IDLE state, and all the transmit FIFOs and the receive FIFO are flushed. • Any transactions on the AHB Master are terminated as soon as possible, after gracefully completing the last data phase of an AHB transfer. Any transactions on the USB are terminated immediately. • When Hibernation or ADP feature is enabled, the PMU module is not reset by the Core Soft Reset. The application can write to this bit any time it wants to reset the core. This is a self-clearing bit and the core clears this bit after all the necessary logic is reset in the core, which can take several clocks, depending on the current state of the core. After this bit is cleared, the application must wait at least 3 PHY clocks before doing any access to the PHY domain (synchronization delay). The application must also must check that bit 31 of this register is 1 (AHB Master is IDLE) before starting any operation.	

To be continued ……

Table 21.177: GRSTCTL, Offset Address: 0x010 (continued)

Bits	Name	Access	Description	Reset
			Typically, software reset is used during software development and also when you dynamically change the PHY selection bits in the USB configuration registers listed above. When you change the PHY, the corresponding clock for the PHY is selected and used in the PHY domain. After a new clock is selected, the PHY domain has to be reset for proper operation. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	
1	PIUFSSftRst	RO	Write Behavior: One to set Mode: Host and Device PIU FS Dedicated Controller Soft Reset (PIUFSSftRst) Resets the PIU FS Dedicated Controller All module state machines in FS Dedicated Controller of PIU are reset to the IDLE state. Used to reset the FS Dedicated controller in PIU in case of any PHY Errors like Loss of activity or Babble Error resulting in the PHY remaining in RX state for more than one frame boundary. This is a self clearing bit and core clears this bit after all the necessary logic is reset in the core. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	
2	FrmCntrRst	RO	Write Behavior: One to set Mode: Host only Host Frame Counter Reset (FrmCntrRst) The application writes this bit to reset the (micro)frame number counter inside the core. When the (micro)frame counter is reset, the subsequent SOF sent out by the core has a (micro)frame number of 0. If the application writes 1 to the bit, it may not be able to read back the value as it gets cleared by the core in a few clock cycles. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	
3	INTknQFlsh	RWS	Mode: Device only IN Token Sequence Learning Queue Flush (INTknQFlsh) This bit is valid only if OTG_EN_DED_TX_FIFO = 0. The application writes this bit to flush the IN Token Sequence Learning Queue.	

To be continued ……

Table 21.178: GRSTCTL, Offset Address: 0x010 (continued)

Bits	Name	Access	Description	Reset
4	RxFFlsh	RO	Write Behavior: One to set Mode: Host and Device RxFIFO Flush (RxFFlsh) The application can flush the entire RxFIFO using this bit, but must first ensure that the core is not in the middle of a transaction. The application must only write to this bit after checking that the core is neither reading from the RxFIFO nor writing to the RxFIFO. The application must wait until the bit is cleared before performing any other operations. This bit requires 8 clocks (slowest of PHY or AHB clock) to clear.	
5	TxFFlsh	RWS	Write Behavior: One to set Mode: Host and Device TxFIFO Flush (TxFFlsh) This bit selectively flushes a single or all transmit FIFOs, but cannot do so if the core is in the midst of a transaction. The application must write this bit only after checking that the core is neither writing to the TxFIFO nor reading from the TxFIFO. Verify using these registers: • Read - NAK Effective Interrupt ensures the core is not reading from the FIFO • Write - GRSTCTL.AHBIdle ensures the core is not writing anything to the FIFO Flushing is normally recommended when FIFOs are reconfigured or when switching between Shared FIFO and Dedicated Transmit FIFO operation. FIFO flushing is also recommended during device endpoint disable. The application must wait until the core clears this bit before performing any operations. This bit takes eight clocks to clear, using the slower clock of phy_clk or hclk.	

To be continued ……

Table 21.179: GRSTCTL, Offset Address: 0x010 (continued)

Bits	Name	Access	Description	Reset
10:6	TxFNum	R/W	Mode: Host and Device TxFIFO Number (TxFNum) This is the FIFO number that must be flushed using the TxFIFO Flush bit. This field must not be changed until the core clears the TxFIFO Flush bit. • 5'h0: – Non-periodic TxFIFO flush in Host mode – Non-periodic TxFIFO flush in device mode when in shared FIFO operation – Tx FIFO 0 flush in device mode when in dedicated FIFO mode • 5'h1: – Periodic TxFIFO flush in Host mode – Periodic TxFIFO 1 flush in Device mode when in shared FIFO operation – TXFIFO 1 flush in device mode when in dedicated FIFO mode • 5'h2: – Periodic TxFIFO 2 flush in Device mode when in shared FIFO operation – TXFIFO 2 flush in device mode when in dedicated FIFO mode ... • 5'hF: – Periodic TxFIFO 15 flush in Device mode when in shared FIFO operation – TXFIFO 15 flush in device mode when in dedicated FIFO mode • 5'h10: – Flush all the transmit FIFOs in device or host mode.	0x0
29:11	Reserved_10_29_11	RO	Reserved for future use.	
30	DMAReq	RO	Mode: Host and Device DMA Request Signal (DMAReq) Indicates that the DMA request is in progress. Used for debug.	
31	AHBIdle	RO	Mode: Host and Device AHB Master Idle (AHBIdle) Indicates that the AHB Master State Machine is in the IDLE condition.	

GINTSTS

Interrupt Status Register

Table 21.180: GINTSTS, Offset Address: 0x014

Bits	Name	Access	Description	Reset
0	CurMod	RO	Mode: Host and Device Current Mode of Operation (CurMod) Indicates the current mode. • 1'b0: Device mode • 1'b1: Host mode Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	
1	ModeMis	RWC	Write Behavior: One to clear Mode: Host and Device Mode Mismatch Interrupt (ModeMis) The core sets this bit when the application is trying to access: • A Host mode register, when the core is operating in Device mode • A Device mode register, when the core is operating in Host mode The register access is completed on the AHB with an OKAY response, but is ignored by the core internally and does not affect the operation of the core. This bit can be set only by the core and the application should write 1 to clear it. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	
2	OTGInt	RO	Mode: Host and Device OTG Interrupt (OTGInt) The core sets this bit to indicate an OTG protocol event. The application must read the OTG Interrupt Status (GOTGINT) register to determine the exact event that caused this interrupt. The application must clear the appropriate status bit in the GOTGINT register to clear this bit. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	

continues on next page

Table 21.180 – continued from previous page

Bits	Name	Access	Description	Reset
3	Sof	RWC	Write Behavior: One to clear Mode: Host and Device Start of (micro)Frame (Sof) In Host mode, the core sets this bit to indicate that an SOF (FS), micro-SOF (HS), or Keep-Alive (LS) is transmitted on the USB. The application must write a 1 to this bit to clear the interrupt. In Device mode, in the core sets this bit to indicate that an SOF token has been received on the USB. The application can read the Device Status register to get the current (micro)Frame number. This interrupt is seen only when the core is operating at either HS or FS.This bit can be set only by the core and the application must write 1 to clear it. Note: The register may return 1'b1 if read immediately after power on reset. If the register bit reads 1'b1 immediately after power on reset, it does not indicate that an SOF has been sent (in host mode), or SOF has been received (in device mode). The read value of this interrupt is valid only after a valid connection between host and device is established. If the bit is set after power on reset, the application can clear the bit.	
4	RxFLvl	RO	Mode: Host and Device RxFIFO Non-Empty (RxFLvl) Indicates that there is at least one packet pending to be read from the RxFIFO.	
5	NPTxFEmp	RO	Mode: Host and Device Non-periodic TxFIFO Empty (NPTxFEmp) This interrupt is asserted when the Non-periodic TxFIFO is either half or completely empty, and there is space for at least one entry to be written to the Non-periodic Transmit Request Queue. The half or completely empty status is determined by the Non-periodic TxFIFO Empty Level bit in the Core AHB Configuration register (GAHBCFG.NPTxFEmpLvl). In host mode, the application can use GINTSTS.NPTxFEmp with the OTG_EN_DED_TX_FIFO parameter set to either 1 or 0. In device mode, the application uses GINTSTS.NPTxFEmp when OTG_EN_DED_TX_FIFO=0. When OTG_EN_DED_TX_FIFO=1, the application uses DIEPINTn.TxFEmp.	

continues on next page

Table 21.180 – continued from previous page

Bits	Name	Access	Description	Reset
6	GINNakEff	RO	Mode: Device only Global IN Non-periodic NAK Effective (GINNakEff) Indicates that the Set Global Non-periodic IN NAK bit in the Device Control register (DCTL.SGNPInNak) set by the application has taken effect in the core. That is, the core has sampled the Global IN NAK bit set by the application. This bit can be cleared by clearing the Clear Global Non-periodic IN NAK bit in the Device Control register (DCTL.CGNPInNak). This interrupt does not necessarily mean that a NAK handshake is sent out on the USB. The STALL bit takes precedence over the NAK bit.	
7	GOUTNakEff	RO	Mode: Device only Global OUT NAK Effective (GOUTNakEff) Indicates that the Set Global OUT NAK bit in the Device Control register (DCTL.SGOUTNak) set by the application has taken effect in the core. This bit can be cleared by writing the Clear Global OUT NAK bit in the Device Control register (DCTL.CGOUTNak).	
8	ULPICKINT _I2CCKINT	RWC	Write Behavior: One to clear Mode: Host and Device ULPI Carkit Interrupt (ULPICKINT) The core sets this interrupt when a ULPI Carkit interrupt is received. The core's PHY sets ULPI Carkit interrupt in UART or Audio mode. This field is used only if the Carkit interface was enabled in coreConsultant (parameter OTG_ULPI_CARKIT = 1). Otherwise, reads return 0. I2C Carkit Interrupt (I2CCKINT) The core sets this interrupt when a Carkit interrupt is received. The core's PHY sets the I2C Carkit interrupt in Audio mode. This field is used only if the I2C interface was enabled in coreConsultant (parameter OTG_I2C_INTERFACE = 1). Otherwise, reads return 0.	
9	I2CINT	RWC	Write Behavior: One to clear Mode: Host and Device I2C Interrupt (I2CINT) The core sets this interrupt when I2C access is completed on the I2C interface. This field is used only if the I2C interface was enabled in coreConsultant (parameter OTG_I2C_INTERFACE = 1). Otherwise, reads return 0.	

continues on next page

Table 21.180 – continued from previous page

Bits	Name	Access	Description	Reset
10	ErlySusp	RWC	Write Behavior: One to clear Mode: Device only Early Suspend (ErlySusp) The core sets this bit to indicate that an Idle state has been detected on the USB For 3 ms.	
11	USBSusp	RWC	Write Behavior: One to clear Mode: Device only USB Suspend (USBSusp) The core sets this bit to indicate that a suspend was detected on the USB. The core enters the Suspend state when there is no activity on the linestate signal for an extended period of time.	
12	USBRst	RWC	Write Behavior: One to clear Mode: Device only USB Reset (USBRst) The core sets this bit to indicate that a reset is detected on the USB.	
13	EnumDone	RWC	Write Behavior: One to clear Mode: Device only Enumeration Done (EnumDone) The core sets this bit to indicate that speed enumeration is complete. The application must read the Device Status (DSTS) register to obtain the enumerated speed.	
14	ISOOutDrop	RWC	Write Behavior: One to clear Mode: Device only Isochronous OUT Packet Dropped Interrupt (ISOOutDrop) The core sets this bit when it fails to write an isochronous OUT packet into the RxFIFO because the RxFIFO does not have enough space to accommodate a maximum packet size packet for the isochronous OUT endpoint.	
15	EOPF	RWC	Write Behavior: One to clear Mode: Device only End of Periodic Frame Interrupt (EOPF) Indicates that the period specified in the Periodic Frame Interval field of the Device Configuration register (DCFG.PerFrInt) has been reached in the current microframe.	
16	RstrDoneInt	RWC	Mode: Host and Device Restore Done Interrupt (RstrDoneInt) The core sets this bit to indicate that the Restore command after Hibernation was completed by the core. The core continues from Suspend state into the mode dictated by the PCGCCTL.RestoreMode field. This bit is valid only when Hibernation feature is enabled (OTG_EN_PWRPOPT=2).	

continues on next page

Table 21.180 – continued from previous page

Bits	Name	Access	Description	Reset
17	EPMis	RO	Write Behavior: One to clear Mode: Device only Endpoint Mismatch Interrupt (EPMis) Note: This interrupt is valid only in shared FIFO operation. Indicates that an IN token has been received for a non-periodic endpoint, but the data for another endpoint is present in the top of the Non-periodic Transmit FIFO and the IN endpoint mismatch count programmed by the application has expired.	
18	IEPInt	RO	Mode: Device only IN Endpoints Interrupt (IEPInt) The core sets this bit to indicate that an interrupt is pending on one of the IN endpoints of the core (in Device mode). The application must read the Device All Endpoints Interrupt (DAINT) register to determine the exact number of the IN endpoint on Device IN Endpoint-n Interrupt (DIEPINTn) register to determine the exact cause of the interrupt. The application must clear the appropriate status bit in the corresponding DIEPINTn register to clear this bit.	
19	OEPInt	RO	Mode: Device only OUT Endpoints Interrupt (OEPInt) The core sets this bit to indicate that an interrupt is pending on one of the OUT endpoints of the core (in Device mode). The application must read the Device All Endpoints Interrupt (DAINT) register to determine the exact number of the OUT endpoint on which the interrupt occurred, and Then read the corresponding Device OUT Endpoint-n Interrupt (DOEPINTn) register to determine the exact cause of the interrupt. The application must clear the appropriate status bit in the corresponding DOEPINTn register to clear this bit.	
20	incompISOIN	RWC	Write Behavior: One to clear Mode: Device only Incomplete Isochronous IN Transfer (incompISOIN) The core sets this interrupt to indicate that there is at least one isochronous IN endpoint on which the transfer is not completed in the current microframe. This interrupt is asserted along with the End of Periodic Frame Interrupt (EOPF) bit in this register. Note: This interrupt is not asserted in Scatter/Gather DMA mode.	

continues on next page

Table 21.180 – continued from previous page

Bits	Name	Access	Description	Reset
21	incomplP_incompISOOUT	RWC	Write Behavior: One to clear Incomplete Periodic Transfer (incomplP) Mode: Host only In Host mode, the core sets this interrupt bit when there are incomplete periodic transactions still pending which are scheduled for the current microframe. Incomplete Isochronous OUT Transfer (incompISOOUT) Mode: Device only In Device mode, the core sets this interrupt to indicate that there is at least one isochronous OUT endpoint on which the transfer is not completed in the current microframe. This interrupt is asserted along with the End of Periodic Frame Interrupt (EOPF) bit in this register.	
22	FetSusp	RWC	Write Behavior: One to clear Mode: Device only Data Fetch Suspended (FetSusp) This interrupt is valid only in DMA mode. This interrupt indicates that the core has stopped fetching data For IN endpoints due to the unavailability of TxFIFO space or Request Queue space. This interrupt is used by the application For an endpoint mismatch algorithm. For example, after detecting an endpoint mismatch, the application: • Sets a Global non-periodic IN NAK handshake • Disables In endpoints • Flushes the FIFO • Determines the token sequence from the IN Token Sequence Learning Queue • Re-enables the endpoints • Clears the Global non-periodic IN NAK handshake If the Global non-periodic IN NAK is cleared, the core has not yet fetched data for the IN endpoint, and the IN token is received. The core generates an 'IN token received when FIFO empty' interrupt. DWC_otg then sends the host a NAK response. To avoid this scenario, the application can check the GINTSTS.FetSusp interrupt, which ensures that the FIFO is full before clearing a Global NAK handshake. Alternatively, the application can mask the "IN token received when FIFO empty" interrupt when clearing a Global IN NAK handshake.	

continues on next page

Table 21.180 – continued from previous page

Bits	Name	Access	Description	Reset
23	ResetDet	RWC	Write Behavior: One to clear Mode: Device only Reset detected Interrupt (ResetDet) In Device mode, this interrupt is asserted when a reset is detected on the USB in partial power-down mode when the device is in Suspend. In Host mode, this interrupt is not asserted.	
24	PrtInt	RO	Mode: Host only Host Port Interrupt (PrtInt) The core sets this bit to indicate a change in port status of one of the DWC_otg core ports in Host mode. The application must read the Host Port Control and Status (HPRT) register to determine the exact event that caused this interrupt. The application must clear the appropriate status bit in the Host Port Control and Status register to clear this bit.	
25	HChInt	RO	Mode: Host only Host Channels Interrupt (HChInt) The core sets this bit to indicate that an interrupt is pending on one of the channels of the core (in Host mode). The application must read the Host All Channels Interrupt (HAINT) register to determine the exact number of the channel on which the interrupt occurred and then read the corresponding Host Channel-n Interrupt (HCINTn) register to determine the exact cause of the interrupt. The application must clear the appropriate status bit in the HCINTn register to clear this bit.	
26	PTxFEmp	RO	Mode: Host only Periodic TxFIFO Empty (PTxFEmp) This interrupt is asserted when the Periodic Transmit FIFO is either half or completely empty and there is space for at least one entry to be written in the Periodic Request Queue. The half or completely empty status is determined by the Periodic TxFIFO Empty Level bit in the Core AHB Configuration register (GAHBCFG.PTxFEmpLvl).	

continues on next page

Table 21.180 – continued from previous page

Bits	Name	Access	Description	Reset
27	LPM_Int	RWC	Write Behavior: One to clear Mode: Host and Device LPM Transaction Received Interrupt (LPM_Int) • Device Mode - This interrupt is asserted when the device receives an LPM transaction and responds with a non-ERROREd response. • Host Mode - This interrupt is asserted when the device responds to an LPM transaction with a non-ERROREd response or when the host core has completed LPM transactions for the programmed number of times (GLPMCFG.RetryCnt).	
28	ConIDStsChng	RWC	Write Behavior: One to clear Mode: Host and Device Connector ID Status Change (ConIDStsChng) The core sets this bit when there is a change in connector ID status.	
29	DisconnInt	RWC	Write Behavior: One to clear Mode: Host only Disconnect Detected Interrupt (DisconnInt) Asserted when a device disconnect is detected.	
30	SessReqInt	RWC	Write Behavior: One to clear Mode: Host and Device Session Request/New Session Detected Interrupt (SessReqInt) In Host mode, this interrupt is asserted when a session request is detected from the device. In Device mode, this interrupt is asserted when the utmisrp_bvalid signal goes high.	

continues on next page

Table 21.180 – continued from previous page

Bits	Name	Access	Description	Reset
31	WkUpInt	RWC	Write Behavior: One to clear Mode: Host and Device Resume/Remote Wakeup Detected Interrupt (WkUpInt) Wakeup Interrupt during Suspend(L2) or LPM(L1) state. • During Suspend (L2): – Device Mode - This interrupt is asserted only when Host Initiated Resume is detected on USB. – Host Mode - This interrupt is asserted only when Device Initiated Remote Wakeup is detected on USB. • During LPM (L1): – Device Mode - This interrupt is asserted for either Host Initiated Resume or Device Initiated Remote Wakeup on USB. – Host Mode - This interrupt is asserted for either Host Initiated Resume or Device Initiated Remote Wakeup on USB.	

GINTMSK

Interrupt Mask Register

Table 21.181: GINTMSK, Offset Address: 0x018

Bits	Name	Access	Description	Reset
0	Reserved_18_0	RO	Reserved for future use. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	
1	ModeMisMsk	R/W	Mode: Host and Device Mode Mismatch Interrupt Mask (ModeMisMsk) Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	0x0
2	OTGIntMsk	R/W	Mode: Host and Device OTG Interrupt Mask (OTGIntMsk) Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0
3	SofMsk	R/W	Mode: Host and Device Start of (micro)Frame Mask (SofMsk)	0x0

continues on next page

Table 21.181 – continued from previous page

Bits	Name	Access	Description	Reset
4	RxFLvlMsk	R/W	Mode: Host and Device Receive FIFO Non-Empty Mask (RxFLvlMsk)	0x0
5	NPTxFEmpMsk	R/W	Mode: Host and Device Non-periodic TxFIFO Empty Mask (NPTxFEmpMsk)	0x0
6	GINNakEffMsk	R/W	Mode: Device only Global Non-periodic IN NAK Effective Mask (GINNakEffMsk)	0x0
7	GOUTNakEffMsk	R/W	Mode: Device only Global OUT NAK Effective Mask (GOUT-NakEffMsk)	0x0
8	ULPICK-INTMsk_I2CCKINTMsk	R/W	ULPI Carkit Interrupt Mask (ULPICKINTMsk) Mode: Host and Device I2C Carkit Interrupt Mask (I2CCKINTMsk) Mode: Host and Device	0x0
9	I2CIntMsk	R/W	Mode: Host and Device I2C Interrupt Mask (I2CIntMsk)	0x0
10	ErlySuspMsk	R/W	Mode: Device only Early Suspend Mask (ErlySuspMsk)	0x0
11	USBSuspMsk	R/W	Mode: Device only USB Suspend Mask (USBSuspMsk)	0x0
12	USBRstMsk	R/W	Mode: Device only USB Reset Mask (USBRstMsk)	0x0
13	EnumDoneMsk	R/W	Mode: Device only Enumeration Done Mask (EnumDoneMsk)	0x0
14	ISOOutDropMsk	R/W	Mode: Device only Isochronous OUT Packet Dropped Interrupt Mask (ISOOutDropMsk)	0x0
15	EOPFMsk	R/W	Mode: Device only End of Periodic Frame Interrupt Mask (EOPFMsk)	0x0
16	RstrDoneIntMsk	R/W	Mode: Host and Device Restore Done Interrupt Mask (RstrDoneIntMsk) This field is valid only when Hibernation feature is enabled (OTG_EN_PWROPT=2).	0x0
17	EPMisMsk	R/W	Mode: Device only Endpoint Mismatch Interrupt Mask (EPMisMsk)	0x0
18	IEPIntMsk	R/W	Mode: Device only IN Endpoints Interrupt Mask (IEPIntMsk)	0x0
19	OEPIntMsk	R/W	Mode: Device only OUT Endpoints Interrupt Mask (OEPIntMsk)	0x0
20	incompISOINMsk	R/W	Mode: Device only Incomplete Isochronous IN Transfer Mask (in-compISOINMsk) This bit is enabled only when device periodic endpoints are enabled in Dedicated TxFIFO mode.	0x0

continues on next page

Table 21.181 – continued from previous page

Bits	Name	Access	Description	Reset
21	incomplPMsk_incompISOOUTMsk	R/W	Incomplete Periodic Transfer Mask (incomplPMsk) Mode: Host only Incomplete Isochronous OUT Transfer Mask (incompISOOUTMsk) Mode: Device only	0x0
22	FetSuspMsk	R/W	Mode: Device only Data Fetch Suspended Mask (FetSuspMsk)	0x0
23	ResetDetMsk	R/W	Mode: Device only Reset detected Interrupt Mask (ResetDetMsk)	0x0
24	PrtIntMsk	R/W	Mode: Host only Host Port Interrupt Mask (PrtIntMsk)	0x0
25	HChIntMsk	R/W	Mode: Host only Host Channels Interrupt Mask (HChIntMsk)	0x0
26	PTxFEmpMsk	R/W	Mode: Host only Periodic TxFIFO Empty Mask (PTxFEmpMsk)	0x0
27	LPM_IntMsk	R/W	Mode: Host and Device LPM Transaction received interrupt Mask (LPM_IntMsk)	0x0
28	ConIDStsChngMsk	R/W	Mode: Host and Device Connector ID Status Change Mask (ConIDStsChngMsk)	0x0
29	DisconnIntMsk	R/W	Mode: Host and Device Disconnect Detected Interrupt Mask (DisconnIntMsk)	0x0
30	SessReqIntMsk	R/W	Mode: Host and Device Session Request/New Session Detected Interrupt Mask (SessReqIntMsk)	0x0
31	WkUpIntMsk	R/W	Mode: Host and Device Resume/Remote Wakeup Detected Interrupt Mask (WkUpIntMsk) The WakeUp bit is used for LPM state wake up in a way similar to that of wake up in suspend state.	0x0

GUID

User ID Register

Table 21.182: GUID, Offset Address: 0x03c

Bits	Name	Access | Description		Reset
31:0	UserID	R/W	User ID (UserID) Application-programmable ID field. Reset: Configurable	0x0

GLPMCFG

Core LPM Configuration Register

Table 21.183: GLPMCFG, Offset Address: 0x054

Bits	Name	Access	Description	Reset
0	LPMCap	R/W	Mode: Host and Device LPM-Capable (LPMCap) The application uses this bit to control the DWC_otg core LPM capabilities. If the core operates as a non-LPM-capable host, it cannot request the connected device or hub to activate LPM mode. If the core operates as a non-LPM-capable device, it cannot respond to any LPM transactions. • 1b0: LPM capability is not enabled • 1b1: LPM capability is enabled This bit is writable only if an LPM mode was specified for Mode of Operation in coreConsultant (parameter OTG_ENABLE_LPM). Otherwise, reads return 0. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0
1	AppL1Res	R/W	Mode: Device only LPM response programmed by application (AppL1Res) Handshake response to LPM token preprogrammed by device application software. The response depends on GLPMCFG.LPMCap. If GLPMCFG.LPMCap is 1'b0, then the core always responds with NYET response. If GLPMCFG.LPMCap is 1'b1, the core response is as follows. • 1: ACK Even though ACK is preprogrammed, the core Device responds with ACK only on successful LPM transaction. The LPM transaction is successful if: – No PID/CRC5 Errors in either EXT token or LPM token (else ERROR) – Valid bLinkState = 0001B (L1) received in LPM transaction (else STALL) – No data pending in transmit queue (else NYET). • 0: NYET The pre-programmed software bit is overridden for response to LPM token when: – The received bLinkState is not L1 (STALL response), or – An error is detected in either of the LPM token packets because of corruption (ERROR response). Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	0x0

To be continued ……

Table 21.184: GLPMCFG, Offset Address: 0x054 (continued)

Bits	Name	Access	Description	Reset
5:2	HIRD	R/W	Mode: Host and Device • EnBESL = 1'b0 Host-Initiated Resume Duration (HIRD) Host Mode: The value of HIRD to be sent in an LPM transaction. This value is also used to initiate resume for a duration TL1HubDrvResume1 for host initiated resume. Device Mode (Read-Only): This field is updated with the Received LPM Token HIRD bmAttribute when an ACK, NYET, or STALL response is sent to an LPM transaction. Sl. No HIRD[3:0] THIRD (us) 1 4'b0000 50 2 4'b0001 125 3 4'b0010 200 4 4'b0011 275 5 4'b0100 350 6 4'b0101 425 7 4'b0110 500 8 4'b0111 575 9 4'b1000 650 10 4'b1001 725 11 4'b1010 800 12 4'b1011 875 13 4'b1100 950 14 4'b1101 1025 15 4'b1110 1100 16 4'b1111 1175 Reset: 4'b0000 • EnBESL = 1'b1 Best Effort Service Latency (BESL) Host Mode: The value of BESL to be sent in an LPM transaction. This value is also used to initiate resume for a duration TL1HubDrvResume1 for host initiated resume. Device Mode (Read-Only): This field is updated with the Received LPM Token BESL bmAttribute when an ACK, NYET, or STALL response is sent to an LPM transaction. Sl. No BESL[3:0] TBESL (us) 1 4'b0000 125 2 4'b0001 150 3 4'b0010 200 4 4'b0011 300 5 4'b0100 400 6 4'b0101 500 7 4'b0110 1000 8 4'b0111 2000	0x0

To be continued ……

Table 21.185: GLPMCFG, Offset Address: 0x054 (continued)

Bits	Name	Access	Description	Reset
			9 4'b1000 3000 10 4'b1001 4000 11 4'b1010 5000 12 4'b1011 6000 13 4'b1100 7000 14 4'b1101 8000 15 4'b1110 9000 16 4'b1111 10000 Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	
6	bRemoteWake	R/W	Mode: Host and Device RemoteWakeEnable (bRemoteWake) Host Mode: The value of remote wake up to be sent in the wIndex field of LPM transaction. Device Mode (Read-Only): This field is updated with the Received LPM Token bRemoteWake bmAttribute when an ACK, NYET, or STALL response is sent to an LPM transaction.	0x0
7	EnblSlpM	R/W	Mode: Host and Device Enable utmi_sleep_n (EnblSlpM) ULPI Interface: The application uses this bit to control the utmi_sleep_n assertion to the PHY when in L1 state. For the host, this bit is valid only in "local device" mode. • 1b0: utmi_sleep_n assertion from the core is not transferred to the external PHY. • 1b1: utmi_sleep_n assertion from the core is transferred to the external PHY. Note: When a ULPI interface is configured, enabling this bit results in a write to Bit 7 of the ULPI Function Control register. The Synopsys ULPI PHY supports writing to this bit, and in the L1 state asserts SleepM when utmi_l1_suspend_n cannot be asserted. Other Interfaces: The application uses this bit to control utmi_sleep_n assertion to the PHY in the L1 state. For the host, this bit is valid only in Local Device mode. • 1'b0: utmi_sleep_n assertion from the core is not transferred to the external PHY. • 1'b1: utmi_sleep_n assertion from the core is transferred to the external PHY when utmi_l1_suspend_n cannot be asserted.	0x0

To be continued ……

Table 21.186: GLPMCFG, Offset Address: 0x054 (continued)

Bits	Name	Access	Description	Reset
12:8	HIRD_Thres	R/W	Mode: Host and Device BESL or HIRD Threshold (HIRD_Thres) Device Mode: • EnBESL = 1'b0: The core puts the PHY into deep low power mode in L1 (by core asserting L1SuspendM) when HIRD value is greater than or equal to the value defined in this field HIRD_Thres[3:0] and HIRD_Thres[4] is set to 1'b1. • EnBESL = 1'b1: The core puts the PHY into deep low power mode in L1 (by core asserting L1SuspendM) when BESL value is greater than or equal to the value defined in this field BESL_Thres[3:0] and BESL_Thres[4] is set to 1'b1. • DCTL.DeepSleepBESLReject = 1'b1: In device initiated resume, the core expects the Host to resume service to the device within the BESL value corresponding to L1 exit time specified in HIRD_Thresh[3:0]. The Device sends a NYET response when the received HIRD in LPM token is greater than HIRD threshold. • Note: To differentiate between Deep Sleep and Shallow sleep HIRD greater than or equal to HIRD threshold comparison is done. For differentiating between NYET or ACK response for LPM token HIRD greater than HIRD Threshold comparison is used. Host Mode: The core puts the PHY into deep low power mode in L1 (by core asserting L1SuspendM) when HIRD_Thres[4] is set to 1b1. HIRD_Thres[3:0] specifies the time for which resume signaling is to be reflected by host (TL1HubDrvResume2) on the USB bus when it detects device initiated resume. HIRD_Thres must not be programmed with a value greater than 4'b1100 in Host mode, because this exceeds maximum TL1HubDrvResume2. Sl. No —-Thres[3:0] — -Host Mode Resume Signaling Time (us) — - — - — - — - ——— - — - EnBESL = 1'b0 - -EnBESL = 1'b1 1 4'b0000 60 75 2 4'b0001 135 100 3 4'b0010 -210 150 4 4'b0011 285 250 5 4'b0100 360 350 6 4'b0101 435 450 7 4'b0110 510 950 8 4'b0111 585 Invalid 9 4'b1000 660 Invalid 10 4'b1001 735 Invalid 11 4'b1010 810 Invalid 12 4'b1011 885 Invalid 13 4'b1100 960 Invalid 14 4'b1101 Invalid Invalid 15 4'b1110 Invalid Invalid 16 4'b1111 Invalid Invalid	0x0

To be continued ……

Table 21.187: GLPMCFG, Offset Address: 0x054 (continued)

Bits	Name	Access	Description	Reset
			The following truth table explains the difference in behavior between the UTMI and ULPI interface in different modes of operation: Bit 7 –Bit 6 –sleep_n —l1_suspend_n –suspend_n —Mode of Operation 0 — —–1- — —-1— — –1— — ——- — –1— — ——- –Normal Operation 0 — —–0- — —-1— — –1— — ——- — –0— — ——- –L2 Suspend 1 — —–0- — —-1— — –0— — ——- — –1— — ——- –L1 Deep Sleep 1 — —–1- — —-0— — –1— — ——- — –1— — ——- –L1 Shallow Sleep	
14:13	CoreL1Res	RO	Mode: Host and Device LPM Response (CoreL1Res) Device Mode: The response of the core to LPM transaction received is reflected in these two bits. Host Mode: Handshake response received from local device for LPM transaction • 11 - ACK • 10 - NYET • 01 - STALL • 00 - ERROR (No handshake response)	
15	SlpSts	RO	Mode: Device only Port Sleep Status (SlpSts) This bit is set as long as a Sleep condition is present on the USB bus. The core enters the Sleep state when an ACK response is sent to an LPM transaction and the TL1TokenRetry timer has expired. To stop the PHY clock, the application must set the Port Clock Stop bit, which asserts the PHY Suspend input signal. The application must rely on SlpSts and not ACK in CoreL1Res to confirm transition into sleep. The core comes out of sleep: • When there is any activity on the USB linestate • When the application writes to the Remote Wakeup Signaling bit in the Device Control register (DCTL.RmtWkUpSig) or when the application resets or softdisconnects the device.	

To be continued ……

Table 21.188: GLPMCFG, Offset Address: 0x054 (continued)

Bits	Name	Access	Description	Reset
			Host Mode: The host transitions to Sleep (L1) state as a side-effect of a successful LPM transaction by the core to the local port with ACK response from the device. The read value of this bit reflects the current Sleep status of the port. The core clears this bit after: • The core detects a remote L1 Wakeup signal, • The application sets the Port Reset bit or the Port L1Resume bit in the HPRT register, or • The application sets the L1Resume/ Remote Wakeup Detected Interrupt bit or Disconnect Detected Interrupt bit in the Core Interrupt register (GINTSTS.L1WkUpInt or GINTSTS.DisconnInt, respectively). Values: • 1b0: Core not in L1 • 1b1: Core in L1	
16	L1ResumeOK	RO	Mode: Host and device Sleep State Resume OK (L1ResumeOK) Indicates that the application or host can start resume from Sleep state. This bit is valid in LPM sleep (L1) state. It is set in sleep mode after a delay of 50 us (TL1Residency). This bit is reset when SlpSts is 0. • 1b1: The application or core can start Resume from Sleep state • 1b0: The application or core cannot start Resume from Sleep state	
20:17	LPM_Chnl_Indx	R/W	Mode: Host only LPM Channel Index (LPM_Chnl_Indx) The channel number on which the LPM transaction has to be applied while sending an LPM transaction to the local device. Based on the LPM channel index, the core automatically inserts the device address and end point number programmed in the corresponding channel into the LPM transaction.	0x0
23:21	LPM_Retry_Cnt	R/W	Mode: Host only LPM Retry Count (LPM_Retry_Cnt) When the device gives an ERROR response, this is the number of additional LPM retries that the host performs until a valid device response (STALL, NYET, or ACK) is received.	0x0

To be continued ……

Table 21.189: GLPMCFG, Offset Address: 0x054 (continued)

Bits	Name	Access	Description	Reset
24	SndLPM	RWS	Write Behavior: One to set Mode: Host only Send LPM Transaction (SndLPM) When the application software sets this bit, an LPM transaction containing two tokens, EXT and LPM is sent. The hardware clears this bit once a valid response (STALL, NYET, or ACK) is received from the Device or the core has finished transmitting the programmed number of LPM retries. Note: This bit must be set only when the host is connected to a local port.	
27:25	LPM_RetryCnt_Sts	RO	Mode: Host only LPM Retry Count Status (LPM_RetryCnt_Sts) Number of LPM Host Retries still remaining to be transmitted for the current LPM sequence.	
28	EnBESL	R/W	Mode: Host and device Enable Best Effort Service Latency (BESL) This bit enables the BESL feature as defined in the LPM errata: • 11'b0: The core works as descri bed in the following document: USB 2.0 Link Power Management Addendum Engineering Change Notice to the USB 2.0 specification, July 16, 2007 • 1'b1: The core works as per the LPM Errata	0x0
29	RstrSlpSts	R/W	Mode: Device only Restore SlpSts (RstrSlpSts) When the application power-gates the core (partial power-down or hibernation), the application needs to program this bit to restore the LPM status in the core. Based on the BESL value received from the Host, the application needs to program this bit during restore process. The application should program this bit depending on whether it decided to put the core in Shallow Sleep (Clock Gating Only) or Deep Sleep (Power Gating) mode: • 1'b0: The application puts the core in Shallow Sleep mode based on the BESL value from the Host. • 1'b1: The application puts the core in Deep Sleep mode based on the BESL value from the Host.	0x0

To be continued ……

Table 21.190: GLPMCFG, Offset Address: 0x054 (continued)

Bits	Name	Access	Description	Reset
30	HSICCon	R/W	Mode: Host and device HSIC-Connect (HSICCon) The application must use this bit The application must use this bit to initiate the HSIC Attach sequence. Host Mode: Once this bit is set, the Host Core configures to drive HSIC Idle state (STROBE=1&DATA=0) on the bus. It then waits for device to initiate the HSIC Connect sequence. Device Mode: Once this bit is set, the Device Core waits for HSIC Idle linestate on the bus. After receiving the Idle linestate it then initiates the HSIC Connect. This bit is valid only if OTG_ENABLE_HSIC = 1, if_sel_hsic = 1, and InvSelHSIC = 0. Otherwise, it is read-only.	0x0
31	InvSelHsic	R/W	Mode: Host and device HSIC-Invert Select HSIC (InvSelHsic) The application uses this bit to control the DWC_otg core HSIC enable/disable. This bit overrides and functionally inverts the if_sel_hsic input port signal. If the core is non-HSIC-capable, it can connect to only PHYs that are not HSIC capable. If the core is HSIC-capable, it can connect only to PHYs that are HSIC capable. • If if_sel_hsic input signal is 1: – InvSelHsic = 1b1: HSIC capability is not enabled – InvSelHsic = 1b0: HSIC capability is enabled • If if_sel_hsic input signal is 0: – InvSelHsic = 1b1: HSIC capability is enabled – InvSelHsic = 1b0: HSIC capability is not enabled This bit is writable only if HSIC mode is specified for Mode of Operation in coreConsultant (parameter OTG_ENABLE_HSIC). This bit is valid only if OTG_ENABLE_HSIC is enabled. Otherwise, reads return 0.	0x0

GPWRDN

Power Down Register

Table 21.191: GPWRDN, Offset Address: 0x058

Bits	Name	Access	Description	Reset
0	PMUIntSel	R/W	Mode: Host and Device PMU Interrupt Select (PMUIntSel) When the hibernation functionality is selected (OTG_EN_PWROPT = 2), a write to this bit with 1'b1 enables the PMU to generate interrupts to the application. During this state, all interrupts from the DWC_otg_core module are blocked to the application. Note: This bit must be set to 1'b1 before the core is put into hibernation • 1'b0: Internal DWC_otg_core interrupt is selected • 1'b1: External DWC_otg_pmu interrupt is selected Note: This bit must not be written to during normal mode of operation. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0
1	PMUActv	R/W	Mode: Host and Device PMU Active (PMUActv) This bit enables or disables the PMU logic. • 1'b0: Disable PMU module • 1'b1: Enable PMU module Note: This bit must not be written to during normal mode of operation. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	0x0
2	Restore	R/W	Mode: Host and Device Restore The application must program this bit to enable or disable restore mode from the PMU module. • 1'b0: DWC_otg in normal mode of operation • 1'b1: DWC_otg in restore mode Note: This bit must not be written to during normal mode of operation. This bit is valid only when OTG_EN_PWROPT = 2. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0

To be continued ……

Table 21.192: GPWRDN, Offset Address: 0x058 (continued)

Bits	Name	Access	Description	Reset
3	PwrDnClmp	R/W	Mode: Host and Device Power Down Clamp (PwrDnClmp) The application must program this bit to enable or disable the clamps to all the outputs of the DWC_otg core module to prevent the corruption of other active logic. • 1'b0: Disable PMU power clamp • 1'b1: Enable PMU power clamp	0x0
4	PwrDnRst_n	R/W	Mode: Host and Device Power Down ResetN (PwrDnRst_n) The application must program this bit to reset the DWC_otg core during the Hibernation exit process or during ADP when powering up the core (if the DWC_otg core was powered off during ADP process). • 1'b1: DWC_otg is in normal operation • 1'b0: Reset DWC_otg Note: This bit must not be written to during normal mode of operation	0x1
5	PwrDnSwtch	R/W	Mode: Host and Device Power Down Switch (PwrDnSwtch) This bit indicates to the DWC_otg core whether the VDD switch is in ON or OFF state. • 1'b0: DWC_otg is in ON state • 1'b1: DWC_otg is in OFF state Note: This bit must not be written to during normal mode of operation.	0x0
6	DisableVBUS	R/W	Mode: Host and Device DisableVBUS Host Mode: The application must program this bit if HPRT0.PrtPwr was programmed to 0 before switching off the Core. This indicates to the PMU whether session was ended before entering Hibernation. • 1'b0: HPRT0.PrtPwr was not programed to 0. • 1'b1: HPRT0.PrtPwr was programmed to 0. Device Mode: The application must program this bit to inform the PMU whether the bvalid valid signal is high (session valid) or low (session end) whenever the core is switched off. • 1'b0: bvalid signal is High (Session Valid) • 1'b1: bvalid signal is Low (Session End) This bit is valid only when GPWRDN.PMUActv is 1.	0x0

To be continued ……

Table 21.193: GPWRDN, Offset Address: 0x058 (continued)

Bits	Name	Access	Description	Reset
7	LnStsChng	RWC	Write Behavior: One to clear Mode: Host and Device Line State Change (LnStsChng) This interrupt is asserted when there is a linestate change detected by the PMU. The application must read GPWRDN.Linestate to determine the current linestate on USB. • 1'b0: No LineState change on USB • 1'b1: LineState change on USB This bit is valid only when GPWRDN.PMUActv is 1 and OTG_EN_PWROPT = 2.	
8	LineStageChangeMsk	R/W	Mode: Host and Device Mask For LineStateChange interrupt (LineStageChangeMsk) This bit is valid only when OTG_EN_PWROPT = 2.	0x0
9	ResetDetected	RWC	Write Behavior: One to clear Mode: Device only ResetDetected This field indicates that Reset has been detected by the PMU module. This field generates an interrupt. • 1'b0: Reset Not Detected • 1'b1: Reset Detected This bit is valid only when OTG_EN_PWROPT = 2.	
10	Mask_ResetDetMsk	R/W	Mode: Device only Mask For ResetDetected interrupt (ResetDetMsk) This bit is valid only when OTG_EN_PWROPT = 2.	0x0

To be continued ……

Table 21.194: GPWRDN, Offset Address: 0x058 (continued)

Bits	Name	Access	Description	Reset
11	DisconnectDetect	RWC	Write Behavior: One to clear Mode: Host only DisconnectDetect This field indicates that Disconnect has been detected by the PMU. This field generates an interrupt. After detecting disconnect during hibernation the application must not restore the core, but instead start the initialization process. • 1'b0: Disconnect not detected • 1'b1: Disconnect detected This bit is valid only when OTG_EN_PWROPT = 2.	
12	Disconnect DetectMsk	R/W	Mode: Host only Mask For DisconnectDetect Interrupt (DisconnectDetectMsk) This bit is valid only when OTG_EN_PWROPT = 2.	0x0
13	ConnectDet	RO	Mode: Host and Device Write Behavior: One to clear ConnectDet This field indicates that a new connect has been detected • 1'b0: Connect not detected • 1'b1: Connect detected This bit is valid only when OTG_EN_PWROPT = 2.	
14	ConnDetMsk	R/W	Mode: Host and Device ConnDetMsk Mask for ConnectDet interrupt This bit is valid only when OTG_EN_PWROPT = 2.	0x0
15	SRPDetect	RWC	Mode: Host only SRPDetect This field indicates that SRP has been detected by the PMU. This field generates an interrupt. After detecting SRP during hibernation the application must not restore the core. The application must get into the initialization process. • 1'b0: SRP not detected • 1'b1: SRP detected	

To be continued ……

Table 21.195: GPWRDN, Offset Address: 0x058 (continued)

Bits	Name	Access	Description	Reset
16	SRPDetectMsk	R/W	Mode: Host only Mask For SRPDetect Interrupt (SRPDetectMsk)	0x0
17	StsChngInt	RWC	Write Behavior: One to clear Status Change Interrupt (StsChngInt) This field indicates a status change in either the IDDIG or BSessVld signal. • 1'b0: No Status change • 1'b1: status change detected After receiving this interrupt the application must read the GPWRDN register and interpret the change in IDDIG or BSesVld with respect to the previous value stored by the application. Note: When Battery Charger is Enabled and the ULPI interface is used, if StsChngInt is received and the application reads GPWRDN register and determines that it is due to a change in the value of IDDIG, then StsChngInt may be generated once again within the next few clock cycles. This occurs because of an ambiguity in the implementation of Battery Charger Support over the ULPI interface. After receiving the StsChngInt for the second time the application can once again read the GPWRDN register. However, this time the value IDDIG (or BSesVld) will not have changed. The application then processes the second interrupt but no further action will be required as a result.	
18	StsChngIntMsk	R/W	Mode: Host and Device Mask For StsChng Interrupt (StsChngIntMsk)	0x0

To be continued ……

Table 21.196: GPWRDN, Offset Address: 0x058 (continued)

Bits	Name	Access	Description	Reset
20:19	LineState	RO	Mode: Host and Device LineState This field indicates the current linestate on USB as seen by the PMU module. • 2'b00: DM = 0, DP = 0 • 2'b01: DM = 0, DP = 1 • 2'b10: DM = 1, DP = 0 • 2'b11: Not-defined This bit is valid only when GPWRDN.PMUActv is 1.	
21	IDDIG	RO	Mode: Host and Device IDDIG This bit indicates the status of the IDDIG signal. The application must read this bit after receiving GPWRDN.StsChngInt and decode based on the previous value stored by the application. Indicates the current mode. • 1'b1: Device mode • 1'b0: Host mode This bit is valid only when GPWRDN.PMUActv is 1.	
22	BSessVld	RO	Mode: Device only B Session Valid (BSessVld) This field reflects the B session valid status signal from the PHY. • 1'b0: B-Valid is 0 • 1'b1: B-Valid is 1 This bit is valid only when GPWRDN.PMUActv is 1.	
23	ADPInt	RWC	Write Behavior: One to clear Mode: Host and Device ADP Interrupt (ADPInt) This bit is set whenever there is a ADP event	
28:24	MultValIdBC	RO	Mode: Host and Device MultValIdBC (MultValIdBC) Battery Charger ACA inputs in the following order: • Bit 28 - rid_float • Bit 27 - rid_gnd • Bit 26 - rid_a • Bit 25 - rid_b • Bit 24 - rid_c These bits are present only if BC_SUPPORT = 1. Otherwise, these bits are reserved and will read 5'h0. Reset: As per ACA input	
31:29	Reserved_58_31_29	RO	Reserved for future use.	

21.6.4 Host Initialization Program Description

After completing the [Frequency Startup Procedure] and [Mode Switching and Initialization Procedure], you need to execute the XHCI initialization procedure, as listed below. Then you can start the four standard types of transmission according to your needs. The details of the method of starting standard transmission can be Refer to the XHCI specification book, so I won't go into details here:

Set the GINTMSK.PrtInt register to unmask state.

Set HCFG register to FS device or HS device.

Set the HPRT.PrtPwr register to 1. This setting will turn on V BUS on the USB bus.

Wait for the HPRT0.PrtConnDet interrupt to occur, which means there is a device connected to the USB downstream port.

Set the HPRT.PrtRst register to 1 and start the USB port reset.

Wait at least 10ms to allow the USB port reset enough time to complete the handshake.

Set HPRT.PrtRst to 0 to complete the USB port reset procedure.

Wait for the HPRT.PrtEnChng interrupt to occur.

Read the HPRT.PrtSpd register to obtain the enumeration speed value.

Set the HFIR register to configure the corresponding PHY Clock.

Set the RXFSIZE register to configure the RXFIFO size.

Set the GNPTXFSIZ register to configure the size of the aperiodic transmission TXFIFO.

Set the HPTXFSIZ register to configure the size of the TXFIFO for periodic transmission.

21.6.5 Host Registers

The base-address of the Host register in the entire memory space is 0x0434_0000. In this article, this base address will be represented by HOST_BASE_ADDR. Therefore, the actual addressing of each register of the Host controller in the memory space will be [HOST_BASE_ADDR+relative address].

21.6.5.1 Register Overview

Table 21.197: Host Registers Overview

Name	Address Offset	Description
HCFG	0x400	Host Configuration Register
HFIR	0x404	Host Frame Interval Register
HFNUM	0x408	Host Frame Number/Frame Time Remaining Register
HPTXSTS	0x410	Host Periodic Transmit FIFO/Queue Status Register
HAINT	0x414	Host All Channels Interrupt Register
HAINTMSK	0x418	Host All Channels Interrupt Mask Register
HFLBAddr	0x41c	Host Frame List Base Address Register
HCCHARn	0x500	Host Channel-n Characteristics Register
HCDMAn	0x514	Host Channel-n DMA Address Register
HCDMABn	0x51c	Host Channel-n DMA Buffer Address Register

21.6.5.2 Register Description

HCFG

Host Configuration Register

Table 21.198: HCFG, Offset Address: 0x400

Bits	Name	Access	Description	Reset
1:0	FSLSPclkSel	R/W	FS/LS PHY Clock Select (FSLSPclkSel) When the core is in FS Host mode: • 2'b00: PHY clock is running at 30/60 MHz • 2'b01: PHY clock is running at 48 MHz • Others: Reserved When the core is in LS Host mode: • 2'b00: PHY clock is running at 30/60 MHz. When the UTMI+/ULPI PHY Low Power mode is not selected, use 30/60 MHz. • 2'b01: PHY clock is running at 48 MHz. When the UTMI+ PHY Low Power mode is selected, use 48MHz If the PHY supplies a 48 MHz clock during LS mode. • 2'b10: PHY clock is running at 6 MHz. In USB 1.1 FS mode, use 6 MHz when the UTMI+ PHY Low Power mode is selected and the PHY supplies a 6 MHz clock during LS mode. If you select a 6 MHz clock during LS mode, you must do a soft reset. • 2'b11: Reserved Notes: • When Core in FS mode, the internal and external clocks have the same frequency. • When Core in LS mode, – If FSLSPclkSel = 2'b00: Internal and external clocks have the same frequency – If FSLSPclkSel = 2'b10: Internal clock is divided by eight version of external 48 MHz clock (utmifs_clk). Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0

To be continued ……

Table 21.199: HCFG, Offset Address: 0x400 (continued)

Bits	Name	Access	Description	Reset
2	FSLSSupp	R/W	FS- and LS-Only Support (FSLSSupp) The application uses this bit to control the core's enumeration speed. Using this bit, the application can make the core enumerate as a FS host, even If the connected device supports HS traffic. Do not make changes to this field after initial programming. • 1'b0: HS/FS/LS, based on the maximum speed supported by the connected device • 1'b1: FS/LS-only, even If the connected device can support HS Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	0x0
6:3	Reserved_400_6_3	RO	Reserved for future use. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	
7	Ena32KHzS	R/W	Enable 32 KHz Suspend mode (Ena32KHzS) This bit can be set only if FS PHY interface is selected. Otherwise, this bit needs to be set to zero. When FS PHY interface is chosen and this bit is set, the core expects that the PHY clock is switched from 48 MHz to 32 KHz during Suspend.	0x0
15:8	ResValid	R/W	Resume Validation Period (ResValid) This field is effective only when HCFG.Ena32KHzS is set. It controls the Resume period when the core resumes from Suspend. The core counts the ResValid number of clock cycles to detect a valid resume when this is set.	0x2
22:16	Reserved_400_22_16	RO	Reserved for future use.	

To be continued ……

Table 21.200: HCFG, Offset Address: 0x400 (continued)

Bits	Name	Access	Description	Reset
23	DescDMA	R/W	Enable Scatter/gather DMA in Host mode (DescDMA) When the Scatter/Gather DMA option is selected during configuration of the RTL, the application can set this bit during initialization to enable the Scatter/Gather DMA operation. Note: This bit must be modified only once after a reset. The following combinations are available for programming: • GAHBCFG.DMAEn=0,HCFG.DescDMA=0 => Slave mode • GAHBCFG.DMAEn=0,HCFG.DescDMA=1 => Invalid • GAHBCFG.DMAEn=1,HCFG.DescDMA=0 => Buffered DMA mode • GAHBCFG.DMAEn=1,HCFG.DescDMA=1 => Scatter/Gather DMA mode In non-Scatter/Gather DMA mode, this bit is reserved.	0x0
25:24	FrListEn	R/W	Frame List Entries (FrListEn) The value in the register specifies the number of entries in the Frame list. This field is valid only in Scatter/Gather DMA mode. • 2'b00: Reserved • 2'b01: 8 Entries • 2'b10: 16 Entries • 2'b11: 32 Entries In non-Scatter/Gather DMA mode, these bits are reserved.	0x0
26	PerSchedEna	R/W	Enable Periodic Scheduling (PerSchedEna) Applicable in Host Scatter/Gather DMA mode only. Enables periodic scheduling within the core. Initially, the bit is res and the core does not process any periodic channels. As soon as this bit is set, the core gets ready to start scheduling periodic channels and sets HCFG.PerSchedStat. The setting of HCFG.PerSchedStat indicates the core has enabled periodic scheduling. Once HCFG.PerSchedEna is set, the application is not supposed to reset the bit unless HCFG.PerSchedStat is set. As soon as this bit is reset, the core gets ready to stop scheduling periodic channels and resets HCFG.PerSchedStat. In non-Scatter/Gather DMA mode, this bit is reserved.	0x0
30:27	Reserved_400_30_27	RO	Reserved for future use.	

To be continued ……

Table 21.201: HCFG, Offset Address: 0x400 (continued)

Bits	Name	Access	Description	Reset
31	ModeChTimEn	R/W	Mode Change Ready Timer Enable (ModeChTimEn) This bit is used to enable/disable the Host core to wait 200 PHY clock cycles at the end of Resume to change the opmode signal to the PHY to 00 after Suspend or LPM. • 1'b0: The Host core waits for either 200 PHY clock cycles or a linestate of SE0 at the end of resume to the change the opmode from 2'b10 to 2'b00 • 1'b1: The Host core waits only for a linestate of SE0 at the end of resume to change the opmode from 2'b10 to 2'b00.	0x0

HFIR

Host Frame Interval Register

Table 21.202: HFIR, Offset Address: 0x404

Bits	Name	Acc ess	Description	R eset
15:0	FrInt	R/W	Frame Interval (FrInt) The value that the application programs to this field specifies the interval between two consecutive SOFs (FS) or micro- SOFs (HS) or Keep-Alive tokens (HS). This field contains the number of PHY clocks that constitute the required frame interval. The default value set in this field for an FS operation when the PHY clock frequency is 60 MHz. The application can write a value to this register only after the Port Enable bit of the Host Port Control and Status register (HPRT.PrtEnaPort) has been set. If no value is programmed, the core calculates the value based on the PHY clock specified in the FS/LS PHY Clock Select field of the Host Configuration register (HCFG.FSLSPclkSel). Do not change the value of this field after the initial configuration. • 125 us * (PHY clock frequency for HS) • 1 ms * (PHY clock frequency for FS/LS) Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	0xEA60
16	HFIRRldCtrl	R/W	Reload Control (HFIRRldCtrl) This bit allows dynamic reloading of the HFIR register during run time. • 1'b0: The HFIR cannot be reloaded dynamically • 1'b1: The HFIR can be dynamically reloaded during runtime. This bit needs to be programmed during initial configuration and its value must not be changed during runtime.	0x0
31:17	Reserved_404_31_17	RO	Reserved for future use. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	

HFNUM

Host Frame Number/Frame Time Remaining Register

Table 21.203: HFNUM, Offset Address: 0x408

Bits	Name	Access	Description	Reset
15:0	FrNum	RO	Frame Number (FrNum) This field increments when a new SOF is transmitted on the USB, and is reset to 0 when it reaches 16'h3FFF. This field is writable only if Remove Optional Features? was not selected in coreConsultant (OTG_RM_OTG_FEATURES = 0). Otherwise, reads return the frame number value. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	
31:16	FrRem	RO	Frame Time Remaining (FrRem) Indicates the amount of time remaining in the current microframe (HS) or Frame (FS/LS), in terms of PHY clocks. This field decrements on each PHY clock. When it reaches zero, this field is reloaded with the value in the Frame Interval register and a new SOF is transmitted on the USB. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	

HPTXSTS

Host Periodic Transmit FIFO/Queue Status Register

Table 21.204: HPTXSTS, Offset Address: 0x410

Bits	Name	Access	Description	Reset
15:0	PTxFSpcAvail	RO	Periodic Transmit Data FIFO Space Available (PTxFSpcAvail) Indicates the number of free locations available to be written to in the Periodic TxFIFO. Values are in terms of 32-bit words • 16'h0: Periodic TxFIFO is full • 16'h1: 1 word available • 16'h2: 2 words available • 16'hn: n words available (n: 0 ~ 32,768) • 16'h8000: 32,768 words available • Others: Reserved Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	
23:16	PTxQSpcAvail	RO	Periodic Transmit Request Queue Space Available (PTxQSpcAvail) Indicates the number of free locations available to be written in the Periodic Transmit Request Queue. This queue holds both IN and OUT requests. • 8'h0: Periodic Transmit Request ueue is full • 8'h1: 1 location available • 8'h2: 2 locations available • n: n locations available (n: 0~16) • Others: Reserved Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	
31:24	PTxQTop	RO	Top of the Periodic Transmit Request Queue (PTxQTop) This indicates the entry in the Periodic Tx Request Queue that is currently being processes by the MAC. This register is used for debugging. • Bit [31]: Odd/Even (micro)Frame – 1'b0: send in even (micro)Frame – 1'b1: send in odd (micro)Frame • Bits [30:27]: Channel/endpoint number • Bits [26:25]: Type – 2'b00: IN/OUT – 2'b01: Zero-length packet – 2'b10: CSPLIT – 2'b11: Disable channel command • Bit [24]: Terminate (last entry for the selected channel or endpoint)	

HAINT

Host All Channels Interrupt Register

Table 21.205: HAINT, Offset Address: 0x414

Bits	Name	Access	Description	Reset
15:0	HAINT	RO	Channel Interrupts (HAINT) One bit per channel: Bit 0 for Channel 0, bit 15 for Channel 15 Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	
31:16	Reserved_414_31_16	RO	Reserved for future use	

HAINTMSK

Host All Channels Interrupt Mask Register

Table 21.206: HAINTMSK, Offset Address: 0x418

Bits	Name	Access	Description	Reset
15:0	HAINTMsk	R/W	Channel Interrupt Mask (HAINTMsk) One bit per channel: Bit 0 for channel 0, bit 15 for channel 15 Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0
31:16	Reserved_418_31_16	RO	Reserved for future use	

HFLBAddr

Host Frame List Base Address Register

Table 21.207: HFLBAddr, Offset Address: 0x41c

Bits	Name	Access	Description	Reset
31:0	HFLBAddr	R/W	The starting address of the Frame list. This register is used only for Isochronous and Interrupt Channels. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0

HCCHARn

Host Channel-n Characteristics Register

Table 21.208: HCCHARn, Offset Address: 0x500

Bits	Name	Access	Description	Reset
10:0	MPS	R/W	Maximum Packet Size (MPS) Indicates the maximum packet size of the associated endpoint. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0
14:11	EPNum	R/W	Endpoint Number (EPNum) Indicates the endpoint number on the device serving as the data source or sink. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	0x0
15	EPDir	R/W	Endpoint Direction (EPDir) Indicates whether the transaction is IN or OUT. • 1'b0: OUT • 1'b1: IN Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0
16	Reserved_500_16	RO	Reserved for future use.	
17	LSpdDev	R/W	Low-Speed Device (LSpdDev) This field is set by the application to indicate that this channel is communicating to a low-speed device. The application must program this bit when a low speed device is connected to the host through an FS HUB. The DWC_otg Host core uses this field to drive the XCVR_SELECT signal to 2'b11 while communicating to the LS Device through the FS hub. Note: In a peer to peer setup, the DWC_otg Host core ignores this bit even if it is set by the application software.	0x0

To be continued ……

Table 21.209: HCCHARn, Offset Address: 0x500 (continued)

Bits	Name	Access	Description	Reset
19:18	EPType	R/W	Endpoint Type (EPType) Indicates the transfer type selected. • 2'b00: Control • 2'b01: Isochronous • 2'b10: Bulk • 2'b11: Interrupt	0x0
21:20	EC	R/W	Multi Count (MC) / Error Count (EC) When the Split Enable bit of the Host Channel-n Split Control register (HCSPLTn.SpltEna) is reset (1'b0), this field indicates to the host the number of transactions that must be executed per microframe for this periodic endpoint. For non-periodic transfers, this field is used only in DMA mode, and specifies the number packets to be fetched for this channel before the internal DMA engine changes arbitration. • 2'b00: Reserved. This field yields undefined results. • 2'b01: 1 transaction • 2'b10: 2 transactions to be issued for this endpoint per microframe • 2'b11: 3 transactions to be issued for this endpoint per microframe When HCSPLTn.SpltEna is set (1'b1), this field indicates the number of immediate retries to be performed for a periodic split transaction on transaction errors. This field must be set to at least 2'b01.	0x0
28:22	DevAddr	R/W	Device Address (DevAddr) This field selects the specific device serving as the data source or sink.	0x0

To be continued ……

Table 21.210: HCCHARn, Offset Address: 0x500 (continued)

Bits	Name	Access	Description	Reset
29	OddFrm	R/W	Odd Frame (OddFrm) This field is set (reset) by the application to indicate that the OTG host must perform a transfer in an odd (micro)frame. This field is applicable for only periodic (isochronous and interrupt) transactions. • 1'b0: Even (micro)frame • 1'b1: Odd (micro)frame This field is not applicable for Scatter/Gather DMA mode and need not be programmed by the application and is ignored by the core.	0x0
30	ChDis	RWS	Write Behavior: One to set Channel Disable (ChDis) The application sets this bit to stop transmitting/receiving data on a channel, even before the transfer for that channel is complete. The application must wait for the Channel Disabled interrupt before treating the channel as disabled.	
31	ChEna	RWS	Write Behavior: One to set Channel Enable (ChEna) When Scatter/Gather mode is enabled: • 1'b0: Indicates that the descriptor structure is not yet ready. • 1'b1: Indicates that the descriptor structure and data buffer with data is setup and this channel can access the descriptor. When Scatter/Gather mode is disabled: This field is set by the application and cleared by the OTG host. • 1'b0: Channel disabled • 1'b1: Channel enabled	

HCDMAn

Host Channel-n DMA Address Register

Table 21.211: HCDMAn, Offset Address: 0x514

Bits	Name	Access	Description	Reset
31:0	DMAAddr	R/W	DMA Address (DMAAddr) This field holds the start address in the external memory from which the data for the endpoint must be fetched or to which it must be stored. This register is incremented on every AHB transaction. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0

HCDMABn

Host Channel-n DMA Buffer Address Register

Table 21.212: HCDMABn, Offset Address: 0x51c

Bits	Name	Access	Description	Reset
31:0	DMABufAddr	R/W	DMA Address (DMAAddr) Holds the current buffer address. This register is updated as and when the data transfer for the corresponding end point is in progress. This register is present only in Scatter/Gather DMA mode. Otherwise this field is reserved. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0

21.6.6 Device Initialization

Follow these steps:

Set the following bits in the DCFG register:

DescDMA is 1, start descriptor DMA mode

Device Speed is HS or FS

Status bits of non-zero transfers

Periodically transmitted Interval value

Set the FIFO Threshold size in DMA transfers

Clear the DCTL.SftDiscon bit to allow the device to start the Connection action to the Host

Clear GINTMSK following bits

USB Port Reset mask

Enumeration done mask

Early suspend mask

USB suspend mask

SOF mask

Wait for the GINTSTS.USBReset interrupt to occur and start the USB Reset initialization process

Wait for the GINTSTS.EnumerationDone interrupt to occur, which means the USB Reset program has been completed, and read the DSTS register, obtain the enumeration speed, and perform the Enumeration initialization program

21.6.7 Device Registers

The basic-address of the Device register in the entire memory space is **0x0434_0000**. In this article, this basic addressing will be represented by DEV_BASE_ADDR, so the actual addressing of each register in the memory space is It will be [DEV_BASE_ADDR + relative address].

21.6.7.1 Register Overview

Table 21.213: Device Registers Overview

Name	Address Offset	Description
DCFG	0x800	Device Configuration Register
DCTL	0x804	Device Control Register
DSTS	0x808	Device Status Register
DIEPMSK	0x810	Device IN Endpoint Common Interrupt Mask Register
DOEPMSK	0x814	Device OUT Endpoint Common Interrupt Mask Register
DAINT	0x818	Device All Endpoints Interrupt Register
DAINTMSK	0x81c	Device Endpoints Interrupt Mask Register
DIEPEMPMSK	0x834	Device IN Endpoint FIFO Empty Interrupt Mask Register
DEACHINT	0x838	Device Each Endpoint Interrupt Register
DEACHINTMSK	0x83c	Device Each Endpoint Interrupt Register Mask

21.6.7.2 Register Description

DCFG

Device Configuration Register

Table 21.214: DCFG, Offset Address: 0x800

Bits	Name	Access	Description	Reset
1:0	DevSpd	R/W	Device Speed (DevSpd) Indicates the speed at which the application requires the core to enumerate, or the maximum speed the application can support. However, the actual bus speed is determined only after the chirp sequence is completed, and is based on the speed of the USB host to which the core is connected. • 2'b00: High speed (USB 2.0 PHY clock is 30 MHz or 60 MHz) • 2'b01: Full speed (USB 2.0 PHY clock is 30 MHz or 60 MHz) • 2'b10: Low speed (USB 1.1 transceiver clock is 6 MHz). If you select 6MHz LS mode, you must do a soft reset. • 2'b11: Full speed (USB 1.1 transceiver clock is 48 MHz) Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0
2	NZStsOUTHShk	R/W	Non-Zero-Length Status OUT Handshake (NZStsOUTHShk) The application can use this field to select the handshake the core sends on receiving a non zero-length data packet during the OUT transaction of a control transfer's Status stage. • 1'b1: Send a STALL handshake ona non zero-length status OUT transaction and do not send the received OUT packet to the application. • 1'b0: Send the received OUT packet to the application (zero-length or non zero-length) and send a handshake based on the NAK and STALL bits for the endpoint in the Device Endpoint Control register. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	0x0

To be continued ……

Table 21.215: DCFG, Offset Address: 0x800 (continued)

Bits	Name	Access	Description	Reset
3	Ena32KHzSusp	R/W	Enable 32 KHz Suspend mode (Ena32KHzSusp) This bit can be set only if FS PHY interface is selected. Otherwise, this bit needs to be set to zero. If FS PHY interface is chosen and this bit is set, the PHY clock during Suspend must be switched from 48 MHz to 32 KHz. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0
10:4	DevAddr	R/W	Device Address (DevAddr) The application must program this field after every SetAddress control command.	0x0
12:11	PerFrInt	R/W	Periodic Frame Interval (PerFrInt) Indicates the time within a (micro)frame at which the application must be notified using the End Of Periodic Frame Interrupt. This can be used to determine if all the isochronous traffic for that (micro)frame is complete. • 2'b00: 80% of the (micro)frame interval • 2'b01: 85% • 2'b10: 90% • 2'b11: 95%	0x0
13	EnDevOutNak	R/W	Enable Device OUT NAK (EnDevOutNak) This bit enables setting NAK for Bulk OUT endpoints after the transfer is completed for Device mode Descriptor DMA mode. • 1'b0: The core does not set NAK after Bulk OUT transfer complete • 1'b1: The core sets NAK after Bulk OUT transfer complete This is a one time programmable bit after reset like any other DCFG register bits. This bit is valid only when OTG_EN_DESC_DMA == 1'b1.	0x0
14	XCVRDLY	R/W	Enables or disables delay between xcvr_sel and txvalid during device chirp • 1'b1: Enable delay between xcvr_sel and txvalid during Device chirp • 1'b0: No delay between xcvr_sel and txvalid during Device chirp	0x0

To be continued ……

Table 21.216: DCFG, Offset Address: 0x800 (continued)

Bits	Name	Access	Description	Reset
15	Errat icIntMsk	R/W	Mode: Device Erratic Error Interrupt Mask • 1'b1: Mask early suspend interrupt on erratic error • 1'b0: Early suspend interrupt is generated on erratic error	0x0
17:16	Reserved_8 00_17_16	RO	Reserved for future use.	
22:18	EPMisCnt	R/W	IN Endpoint Mismatch Count (EPMisCnt) This field is valid only in shared FIFO operation. The application programs this field with a count that determines when the core generates an Endpoint Mismatch interrupt (GINTSTS.EPMis). The core loads this value into an internal counter and decrements it. The counter is reloaded whenever there is a match or when the counter expires. The width of this counter depends on the depth of the Token Queue.	0x8
23	DescDMA	R/W	Enable Scatter/Gather DMA in Device mode (DescDMA). When the Scatter/Gather DMA option is selected during configuration of the RTL, the application can set this bit during initialization to enable the Scatter/Gather DMA operation. Note: This bit must be modified only once after a reset. The following combinations are available for programming: • GAHBCFG.DMAEn=0,DCFG.DescDMA=0 => Slave mode • GAHBCFG.DMAEn=0,DCFG.DescDMA=1 => Invalid • GAHBCFG.DMAEn=1,DCFG.DescDMA=0 => Buffer DMA mode • GAHBCFG.DMAEn=1,DCFG.DescDMA=1 => Scatter/Gather DMA mode	0x0

To be continued ……

Table 21.217: DCFG, Offset Address: 0x800 (continued)

Bits	Name	Access	Description	Reset
25:24	PerSchIntvl	R/W	Periodic Scheduling Interval (PerSchIntvl) PerSchIntvl must be programmed only for Scatter/Gather DMA mode. This field specifies the amount of time the Internal DMA engine must allocate For fetching periodic IN endpoint data. Based on the number of periodic endpoints, this value must be specified as 25,50 or 75% of (micro)frame. When any periodic endpoints are active, the internal DMA engine allocates the specified amount of time in fetching periodic IN endpoint data. When no periodic endpoints are active, the internal DMA engine services non-periodic endpoints, ignoring this field. After the specified time within a (micro)frame, the DMA switches to fetching for non-periodic endpoints. • 2'b00: 25% of (micro)frame. • 2'b01: 50% of (micro)frame. • 2'b10: 75% of (micro)frame. • 2'b11: Reserved.	0x0
31:26	ResValid	R/W	Resume Validation Period (ResValid) This field is effective only when DCFG.Ena32KHzSusp is set. It controls the resume period when the core resumes from suspend. The core counts for "ResValid" number of clock cycles to detect a valid resume when this bit is set.	0x2

DCTL

Device Control Register

Table 21.218: DCTL, Offset Address: 0x804

Bits	Name	Access	Description	Reset
0	RmtWkUpSig	R/W	Remote Wakeup Signaling (RmtWkUpSig) When the application sets this bit, the core initiates remote signaling to wake the USB host. The application must set this bit to instruct the core to exit the Suspend state. As specified in the USB 2.0 specification, the application must clear this bit 1- 15 ms after setting it.If LPM is enabled and the core is in the L1 (Sleep) state, when the application sets this bit, the core initiates L1 remote signaling to wake up the USB host. The application must set this bit to instruct the core to exit the Sleep state. As specified in the LPM specification, the hardware automatically clears this bit 50 us (TL1DevDrvResume) after being set by the application. The application must not set this bit when GLPMCFG bRemoteWake from the previous LPM transaction is zero. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0
1	SftDiscon	R/W	Soft Disconnect (SftDiscon) The application uses this bit to signal the DWC_otg core to do a soft disconnect. As long as this bit is set, the host does not see that the device is connected, and the device does not receive signals on the USB. The core stays in the disconnected state until the application clears this bit. • 1'b0: Normal operation. When this bit is cleared after a soft disconnect, the core drives the phy_opmode_o signal on the UTMI+ to 2'b00, which generates a device connect event to the USB host. When the device is reconnected, the USB host restarts device enumeration. • 1'b1: The core drives the phy_opmode_o signal on the UTMI+ to 2'b01, which generates a device disconnect event to the USB host. Note: This bit can be also used for ULPI/FS Serial interfaces. Note: This bit is not impacted by a soft reset. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	0x1

To be continued ……

Table 21.219: DCTL, Offset Address: 0x804 (continued)

Bits	Name	Access	Description	Reset
2	GNPINNakSts	RO	Global Non-periodic IN NAK Status (GNPINNakSts) • 1'b0: A handshake is sent out based on the data availability in the transmit FIFO. • 1'b1: A NAK handshake is sent out on all non-periodic IN endpoints, irrespective of the data availability in the transmit FIFO. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	
3	GOUTNakSts	RO	Global OUT NAK Status (GOUTNakSts) • 1'b0: A handshake is sent based on the FIFO Status and the NAK and STALL bit settings. • 1'b1: No data is written to the RxFIFO, irrespective of space availability. Sends a NAK handshake on all packets, except on SETUP transactions. All isochronous OUT packets are dropped.	
6:4	TstCtl	R/W	Test Control (TstCtl) • 3'b000: Test mode disabled • 3'b001: Test_J mode • 3'b010: Test_K mode • 3'b011: Test_SE0_NAK mode • 3'b100: Test_Packet mode • 3'b101: Test_Force_Enable • Others: Reserved	0x0
7	SGNPInNak	RWC	Set Global Non-periodic IN NAK (SGNPInNak) A write to this field sets the Global Non-periodic IN NAK.The application uses this bit to send a NAK handshake on all nonperiodic IN endpoints. The core can also set this bit when a timeout condition is detected on a non-periodic endpoint in shared FIFO operation. The application must set this bit only after making sure that the Global IN NAK Effective bit in the Core Interrupt Register (GINTSTS.GINNakEff) is cleared.	
8	CGNPInNak	RWC	Clear Global Non-periodic IN NAK (CGNPInNak) A write to this field clears the Global Non-periodic IN NAK.	

To be continued ……

Table 21.220: DCTL, Offset Address: 0x804 (continued)

Bits	Name	Access	Description	Reset
9	SGOUTNak	RWC	Set Global OUT NAK (SGOUTNak) A write to this field sets the Global OUT NAK. The application uses this bit to send a NAK handshake on all OUT endpoints. The application must set the this bit only after making sure that the Global OUT NAK Effective bit in the Core Interrupt Register (GINTSTS.GOUTNakEff) is cleared.	
10	CGOUTNak	RWC	Clear Global OUT NAK (CGOUTNak) A write to this field clears the Global OUT NAK.	
11	PWROnPrgDone	R/W	Power-On Programming Done (PWROnPrgDone) The application uses this bit to indicate that register programming is complete after a wake-up from Power Down mode.	0x0
12	Reserved_804_12	RO	Reserved for future use.	
14:13	GMC	R/W	Global Multi Count (GMC) GMC must be programmed only once after initialization. Applicable only for Scatter/Gather DMA mode. This indicates the number of packets to be serviced for that end point before moving to the next end point. It is only for non-periodic end points. • 2'b00: Invalid. • 2'b01: 1 packet. • 2'b10: 2 packets. • 2'b11: 3 packets. The value of this field automatically changes to 2'h1 when DCFG.DescDMA is set to 1. When Scatter/Gather DMA mode is disabled, this field is reserved and reads 2'b00.	0x0

To be continued ……

Table 21.221: DCTL, Offset Address: 0x804 (continued)

Bits	Name	Access	Description	Reset
15	IgnrFrmNum	R/W	Ignore frame number for isochronous endpoints (IgnrFrmNum) Slave Mode (GAHBCFG.DMAEn=0): This bit is not valid in Slave mode and should not be programmed to 1.Non-Scatter/Gather DMA mode (GAHBCFG.DMAEn=1,DCFG.DescDMA=0): This bit is not used when Threshold mode is enabled and should not be programmed to 1. In non-Scatter/Gather DMA mode, the application receives transfer complete interrupt after transfers for multiple (micro)frames are completed. • When Scatter/Gather DMA mode is disabled, this field is used by the application to enable periodic transfer interrupt. The application can program periodic endpoint transfers for multiple (micro)frames. • 0: Periodic transfer interrupt feature is disabled; the application must program transfers for periodic endpoints every (micro)frame • 1: Packets are not flushed when an ISOC IN token is received for an elapsed frame. The core ignores the frame number, sending packets as soon as the packets are ready, and the corresponding token is received. This field is also used by the application to enable periodic transfer interrupts. Scatter/Gather DMA Mode (GAHBCFG.DMAEn=1,DCFG.DescDMA=1): This bit is not applicable to high-speed, high-bandwidth transfers and should not be programmed to 1. In addition, this bit is not used when Threshold mode is enabled and should not be programmed to 1. • 0: The core transmits the packets only in the frame number in which they are intended to be transmitted. • 1: Packets are not flushed when an ISOC IN token is received for an elapsed frame. The core ignores the frame number, sending packets as soon as the packets are ready, and the corresponding token is received. When this bit is set, there must be only one packet per descriptor.	0x0
16	NakOnBble	R/W	NAK on Babble Error (NakOnBble) Set NAK automatically on babble (NakOnBble). The core sets NAK automatically for the endpoint on which babble is received.	0x0

To be continued ……

Table 21.222: DCTL, Offset Address: 0x804 (continued)

Bits	Name	Access	Description	Reset
17	EnContOnBNA	R/W	Enable Continue on BNA (EnContOnBNA) This bit enables the DWC_otg core to continue on BNA for Bulk OUT and INTR OUT endpoints. With this feature enabled, when a Bulk OUT or INTR OUT endpoint receives a BNA interrupt the core starts processing the descriptor that caused the BNA interrupt after the endpoint re-enables the endpoint. • 1'b0: After receiving BNA interrupt, the core disables the endpoint. When the endpoint is re-enabled by the application, the core starts processing from the DOEPDMA descriptor. • 1'b1: After receiving BNA interrupt, the core disables the endpoint. When the endpoint is re-enabled by the application, the core starts processing from the descriptor that received the BNA interrupt. This bit is valid only when OTG_EN_DESC_DMA == 1'b1. It is a one-time programmable after reset bit like any other DCTL register bits.	0x0
18	DeepSleepBESLReject	R/W	Deep Sleep BESL Reject Core rejects LPM request with HIRD value greater than HIRD threshold programmed. NYET response is sent for LPM tokens with HIRD value greater than HIRD threshold. By default, the Deep Sleep BESL Reject feature is disabled.	0x0
31:19	Reserved_8 04_31_19	RO	Reserved for future use.	

DSTS

Device Status Register

Table 21.223: DSTS, Offset Address: 0x808

Bits	Name	Access	Description	Reset
0	SuspSts	RO	Suspend Status (SuspSts) In Device mode, this bit is set as long as a Suspend condition is detected on the USB. The core enters the Suspend state when there is no activity on the phy_line_state_i signal for an extended period of time. The core comes out of the suspend under the following conditions: • If there is any activity on the phy_line_state_i signal • If the application writes to the Remote Wakeup Signaling bit in the Device Control register (DCTL.RmtWkUpSig). Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	
2:1	EnumSpd	RO	Enumerated Speed (EnumSpd) Indicates the speed at which the DWC_otg core has come up after speed detection through a chirp sequence. • 2'b00: High speed (PHY clock is running at 30 or 60 MHz) • 2'b01: Full speed (PHY clock is running at 30 or 60 MHz) • 2'b10: Low speed (PHY clock is running at 6 MHz) • 2'b11: Full speed (PHY clock is running at 48 MHz) Low speed is not supported for devices using a UTMI+ PHY.	
3	ErrticErr	RO	Erratic Error (ErrticErr) The core sets this bit to report any erratic errors (phy_rxvalid_i/phy_rxvldh_i or phy_rxactive_i is asserted For at least 2 ms, due to PHY error) seen on the UTMI+. Due to erratic errors, the DWC_otg core goes into Suspend state and an interrupt is generated to the application with Early Suspend bit of the Core Interrupt register (GINTSTS.ErlySusp). If the early suspend is asserted because of an erratic error, the application can only perform a soft disconnect recover.	
7:4	Reserved_808_7_4	RO	Reserved for future use.	

To be continued ……

Table 21.224: DSTS, Offset Address: 0x808 (continued)

Bits	Name	Access	Description	Reset
21:8	SOFFN	RO	Frame or Microframe Number of the Received SOF (SOFFN) When the core is operating at high speed, this field contains a microframe number. When the core is operating at full or low speed, this field contains a Frame number. Note: This register may return a non zero value if read immediately after power on reset. In case the register bit reads non zero immediately after power on reset it does not indicate that SOF has been received from the host. The read value of this interrupt is valid only after a valid connection between host and device is established.	
23:22	DevLnSts	RO	Device Line Status (DevLnSts) Indicates the current logic level USB data lines • Bit [23]: Logic level of D+ • Bit [22]: Logic level of D-	
31:24	Reserved_808_31_24	RO	Reserved for future use. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	

DIEPMSK

Device IN Endpoint Common Interrupt Mask Register

Table 21.225: DIEPMSK, Offset Address: 0x810

Bits	Name	Access	Description	Reset
0	DiXferComplMsk	R/W	Transfer Completed Interrupt Mask (XferComplMsk) Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0
1	DiEPDisbldMsk	R/W	Endpoint Disabled Interrupt Mask (EPDisbldMsk)	0x0
2	DiAHBErrMsk	R/W	AHB Error Mask (AHBErrMsk)	0x0
3	TimeOUTMsk	R/W	Timeout Condition Mask (TimeOUTMsk) (Non-isochronous endpoints)	0x0
4	INTknTXFEmpMsk	R/W	IN Token Received When TxFIFO Empty Mask (INTknTXFEmpMsk)	0x0
5	INTknEPMisMsk	R/W	IN Token received with EP Mismatch Mask (INTknEPMisMsk)	0x0
6	INEPNakEffMsk	R/W	IN Endpoint NAK Effective Mask (INEPNakEffMsk)	0x0
7	Reserved_810_7	RO	Reserved for future use.	
8	TxfifoUndrnMsk	R/W	Fifo Underrun Mask (TxfifoUndrnMsk)	0x0
9	BNAInIntrMsk	R/W	BNA Interrupt Mask (BNAInIntrMsk) This bit is valid only when Device Descriptor DMA is enabled.	0x0
12:10	Reserved_810_12_10	RO	Reserved for future use.	
13	DiNAKMsk	R/W	NAK interrupt Mask (NAKMsk) Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	0x0
31:14	Reserved_810_31_14	RO	Reserved for future use.	

DOEPMSK

Device OUT Endpoint Common Interrupt Mask Register

Table 21.226: DOEPMSK, Offset Address: 0x814

Bits	Name	Access	Description	Reset
0	XferComplMsk	R/W	Transfer Completed Interrupt Mask (XferComplMsk) Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0
1	EPDisbldMsk	R/W	Endpoint Disabled Interrupt Mask (EPDisbldMsk)	0x0
2	AHBErrMsk	R/W	AHB Error (AHBErrMsk)	0x0
3	SetUPMsk	R/W	SETUP Phase Done Mask (SetUPMsk) Applies to control endpoints only.	0x0
4	OUTTknEPdisMsk	R/W	OUT Token Received when Endpoint Disabled Mask (OUTTknEPdisMsk) Applies to control OUT endpoints only.	0x0
5	StsPhseRcvdMsk	R/W	Status Phase Received Mask (StsPhseRcvdMsk)	0x0
6	Back2BackSETupMsk	R/W	Back-to-Back SETUP Packets Received Mask (Back2BackSETupMsk) Applies to control OUT endpoints only.	0x0
7	Reserved_814_7	RO	Reserved for future use.	
8	OutPktErrMsk	R/W	OUT Packet Error Mask (OutPktErrMsk)	0x0
9	BnaOutIntrMsk	R/W	BNA interrupt Mask (BnaOutIntrMsk)	0x0
11:10	Reserved_814_11_10	RO	Reserved for future use.	
12	BbleErrMsk	R/W	Babble Error interrupt Mask (BbleErrMsk)	0x0
13	NAKMsk	R/W	NAK interrupt Mask (NAKMsk)	0x0
14	NYETMsk	R/W	NYET interrupt Mask (NYETMsk) Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel	0x0
31:15	Reserved_814_31_15	RO	Reserved for future use.	

DAINT

Device All Endpoints Interrupt Register

Table 21.227: DAINT, Offset Address: 0x818

Bits	Name	Access	Description	Reset
15:0	InEpInt	RO	OUT Endpoint Interrupt Bits (OutEPInt) One bit per OUT endpoint: Bit 16 for OUT endpoint 0, bit 31 for OUT endpoint 15 Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	
31:16	OutEPInt	RO	OUT Endpoint Interrupt Bits (OutEPInt) One bit per OUT endpoint: Bit 16 for OUT endpoint 0, bit 31 for OUT endpoint 15	

DAINTMSK

Device Endpoints Interrupt Mask Register

Table 21.228: DAINTMSK, Offset Address: 0x81c

Bits	Name	Access	Description	Reset
15:0	InEpMsk	R/W	IN EP Interrupt Mask Bits (InEpMsk) One bit per IN Endpoint: Bit 0 for IN EP 0, bit 15 for IN EP 15 The value of this field depends on the number of IN endpoints that are configured. Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0
31:16	OutEpMsk	R/W	OUT EP Interrupt Mask Bits (OutEpMsk) One per OUT endpoint: Bit 16 for OUT EP 0, bit 31 for OUT EP 15 The value of this field depends on the number of OUT endpoints that are configured.	0x0

DIEPEMPMSK

Device IN Endpoint FIFO Empty Interrupt Mask Register

Table 21.229: DIEPEMPMSK, Offset Address: 0x834

Bits	Name	Access	Description	Reset
15:0	InEpTxfEmpMsk	R/W	IN EP Tx FIFO Empty Interrupt Mask Bits (InEpTxfEmpMsk) These bits acts as mask bits for DIEPINTn. TxFEmp interrupt One bit per IN Endpoint: • Bit 0 for IN endpoint 0 • … • Bit 15 for endpoint 15 Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0
31:16	Reserved_834_31_16	RO	Reserved for future use.	

DEACHINT

Device Each Endpoint Interrupt Register

Table 21.230: DEACHINT, Offset Address: 0x838

Bits	Name	Access	Description	Reset
15:0	EchInEpInt	RO	IN Endpoint Interrupt Bits (EchInEpInt) One bit per IN Endpoint: • Bit 0 for IN endpoint 0 • ... • Bit 15 for endpoint 15 Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	
31:16	EchOutEPInt	RO	OUT Endpoint Interrupt Bits (EchOutEPInt) One bit per OUT endpoint: • Bit 16 for OUT endpoint 0 • ... • Bit 31 for OUT endpoint 15	

DEACHINTMSK

Device Each Endpoint Interrupt Register Mask

Table 21.231: DEACHINTMSK, Offset Address: 0x83c

Bits	Name	Access	Description	Reset
15:0	EchInEpMsk	R/W	IN EP Interrupt Mask Bits (EchInEpMsk) One bit per IN Endpoint: • Bit 0 for IN endpoint 0 • ... • Bit 15 for endpoint 15 Shadow: Yes Shadow Ctrl: vs_1t Shadow Read Select: shrd_sel One-Way: Enabled	0x0
31:16	EchOutEpMsk	R/W	OUT EP Interrupt Mask Bits (EchOutEpMsk) One per OUT Endpoint: • Bit 16 for IN endpoint 0 • ... • Bit 31 for endpoint 15	0x0

21.7 SARADC

21.7.1 Overview

SARADC is an analog signal to digital conversion controller. This chip has up to 2 SARADC controllers, one in the Active Domain and another one in the No-die Domain, each providing 3 independent channels.

Note: The chip does not bring out all ADC channels on the pins for the controller under Active Domain. For details, please refer to *ADC*.

21.7.2 Features

- Controller operating frequency 12.5MHz;
- The scanning frequency cannot be higher than 320K/s;
- Each controller provides 3 independent channels;
- 12bit sampling accuracy;
- Can trigger sequential scanning of three channels at one time;
- Automatically report interruption after scanning is completed;

21.7.3 Way of Working

The CPU configures the scanning channel. Each SARADC controller can configure 3 channels at the same time and start SARADC for channel scanning. After the channel scan completes all enabled channels, the system is notified of the completion of the scan through an interrupt, and the CPU can obtain the conversion results.

After the system is powered on, in order to ensure the SARADC measurement accuracy, it is recommended to calibrate the SARADC module of the chip. The calibration is performed offline. The specific method is to first set saradc_test.reg_saradc_vrefsel to external mode, and then manually adjust saradc_trim.reg_saradc_trim repeatedly until the actual read sampling value is close to the external reference voltage value after conversion to meet the accuracy requirements. After the calibration is completed, record the value of saradc_trim.reg_saradc_trim. Then each time the power is turned on, through software programming, set saradc_test.reg_saradc_vrefsel to external mode and then set the recorded value to saradc_trim.reg_saradc_trim.

21.7.4 SARADC Register Overview

1 group in Active Domain, Base address: 0x030F0000

1 group in No-die Domain (RTCSYS_SARADC), Base address: 0x0502C000

Each set of registers has the same definition, taking RTCSYS_SARADC as an example:

Table 21.232: Base address 0x0502C000

Name	Address Offset	Description
saradc_ctrl	0x004	control register
saradc_status	0x008	status register
saradc_cyc_set	0x00c	saradc waveform setting register
saradc_ch1_result	0x014	channel 1 result register
saradc_ch2_result	0x018	channel 2 result register
saradc_ch3_result	0x01c	channel 3 result register
saradc_intr_en	0x020	interrupt enable register
saradc_intr_clr	0x024	interrupt clear register
saradc_intr_sta	0x028	interrupt status register
saradc_intr_raw	0x02c	interrupt raw status register
saradc_test	0x030	test register
saradc_trim	0x034	trim register

21.7.5 SARADC Register Description

21.7.5.1 saradc_ctrl

Table 21.233: saradc_ctrl, Offset Address: 0x004

Bits	Name	Access	Description	Reset
0	reg_saradc_en	RWS	when re_saradc_en is set , saradc start to measure the channels enabled in reg_saradc_sel	
3:1	Reserved			
7:4	reg_saradc_sel	R/W	select channel (1~3)	0x0
31:8	Reserved			

21.7.5.2 saradc_status

Table 21.234: saradc_status, Offset Address: 0x008

Bits	Name	Access	Description	Reset
0	sta_saradc_busy	RO	busy rise when re_saradc_en is set	
3:1	Reserved			
7:4	sta_saradc_ch_busy	RO	per channel busy status	
15:8	Reserved			
19:16	sta_saradc_st	RO	fsm status for debug	
24:20	sta_saradc_cycle	RO	sample cycle for debug	
31:25	Reserved			

21.7.5.3 saradc_cyc_set

Table 21.235: saradc_cyc_set, Offset Address: 0x00c

Bits	Name	Access	Description	Reset
4:0	reg_saradc_cyc_settling	R/W	saradc startup cycle = 1 + reg_saradc_cyc_settling , default is 16 cycle	0xF
7:5	Reserved			
11:8	reg_saradc_cyc_samp	R/W	saradc sample window = 1 + reg_saradc_cyc_samp , default is 4 cycle	0x3
15:12	reg_saradc_cyc_clkdiv	R/W	saradc clock divider , freq = ip_clk/(1+clk_div) , default is 25M/2 = 12.5M = 80ns	0x1
19:16	reg_saradc_cyc_comp	R/W	saradc compare cycle = 1+ reg_saradc_cyc_comp , default is 12 cycle	0xB
31:20	Reserved			

21.7.5.4 saradc_ch1_result

Table 21.236: saradc_ch1_result, Offset Address: 0x014

Bits	Name	Access	Description	Reset
11:0	sta_saradc_ch1_result	RO	ch1 measure result	
14:12	Reserved			
15	sta_saradc_ch1_valid	RO	ch1 measure result is valid. The valid status will be cleared when this channel is re-triggered to remeasure,	
31:16	Reserved			

21.7.5.5 saradc_ch2_result

Table 21.237: saradc_ch2_result, Offset Address: 0x018

Bits	Name	Access	Description	Reset
11:0	sta_saradc_ch2_result	RO	ch2 measure result	
14:12	Reserved			
15	sta_saradc_ch2_valid	RO	ch2 measure result is valid. The valid status will be cleared when this channel is re-triggered to remeasure,	
31:16	Reserved			

21.7.5.6 saradc_ch3_result

Table 21.238: saradc_ch3_result, Offset Address: 0x01c

Bits	Name	Access	Description	Reset
11:0	sta_saradc_ch3_result	RO	ch3 measure result	
14:12	Reserved			
15	sta_saradc_ch3_valid	RO	ch3 measure result is valid. The valid status will be cleared when this channel is re-triggered to remeasure,	
31:16	Reserved			

21.7.5.7 saradc_intr_en

Table 21.239: saradc_intr_en, Offset Address: 0x020

Bits	Name	Access	Description	Reset
0	sta_saradc_intr_en	R/W	interrupt enable (mask)	0x0
31:1	Reserved			

21.7.5.8 saradc_intr_clr

Table 21.240: saradc_intr_clr, Offset Address: 0x024

Bits	Name	Access	Description	Reset
0	sta_saradc_intr_clr	RWC	interrupt clear	
31:1	Reserved			

21.7.5.9 saradc_intr_sta

Table 21.241: saradc_intr_sta, Offset Address: 0x028

Bits	Name	Access	Description	Reset
0	sta_saradc_intr_sta	RO	interrup masked status [0]: all channels measurenment in this time is finished	
31:1	Reserved			

21.7.5.10 saradc_intr_raw

Table 21.242: saradc_intr_raw, Offset Address: 0x02c

Bits	Name	Access	Description	Reset
0	sta_saradc_intr_raw	RO	interrupt raw status [0]: all channels measurenment in this time is finished	
31:1	Reserved			

21.7.5.11 saradc_test

Table 21.243: saradc_test, Offset Address: 0x030

Bits	Name	Access	Description	Reset
0:1	Reserved			
2	reg_saradc_vrefsel	RW	1'b0:internal 1'b1:external(from VDD18A)	0
31:3	Reserved			

21.7.5.12 saradc_trim

Table 21.244: saradc_trim, Offset Address: 0x034

Bits	Name	Access	Description	Reset
3:0	reg_saradc_trim	RW	bit[0:2]: The larger the value, the smaller the vref, and the larger the reading. bit[3]: reverse, from b1000 to b0111 vref decreases	0x0
31:4	Reserved			

21.8 Temperature Sensor

21.8.1 Overview

Because the chip junction temperature is too high, it may cause thermal run-away and cause permanent damage. Therefore, the chip needs to be temperature controlled.

The first stage is software behavior. The temperature sensor can automatically detect whether the temperature exceeds a specific temperature at regular intervals and issue an overheating interrupt. After the software receives the overheating interrupt, it can reduce power consumption and temperature by limiting the frequency or voltage of high-power modules, starting fans, etc. the goal of. If the temperature returns to a safe range, the restrictions are lifted.

The second stage is hardware behavior. If the temperature continues to rise after the software is activated, the hardware will intervene to perform thermal shut-down emergency response. However, this function is turned off by default. You need to set the relevant settings after starting the software and then enable it.

This chip has two built-in temperature sensors, and the CPU can periodically monitor the chip temperature. When the chip is overheated and unresponsive, the power management module can be triggered to shut down and reset the system to avoid the risk of overheating.

21.8.2 Way of Working

- Single measurement time configuration: reg_tempsen_accsel = 1 (1024T) configuration, the required single measurement time is (1/(25M/12)*(1024+2+64) ~ 523.2us.
- Cycle measurement time configuration: reg_tempsen_auto_prediv is defaulted to 24, making each unit of reg_tempsen_auto_cycle 1us. The period between two measurement times needs to be configured with reg_tempsen_auto_cycle greater than 524. For example, to measure once per second, configure reg_tempsen_auto_cycle = 1000000.
- Measurement channel configuration: If reg_tempsen_sel = 3 is configured, two temperature sensors will be triggered to measure simultaneously.
- Configure high and low temperature monitoring temperature: Configure the temperature tempsen_chx_temp_th that triggers high temperature alarm and low temperature recovery.
- Configuration interruption.
- Enable the temperature sensor for measurement: configure reg_tempsen_en = 1 to perform measurement and wait for interrupt.

Table 21.245: Tempsensor Interrupt Signal Description

Bit	Signal	Description
[0]	Irq_Temp0_measure	Tempsens0 measurement completed.
[1]	Irq_Temp1_measure	Tempsens1 measurement completed.
[2]	reserved	Reserved
[3]	reserved	Reserved
[4]	Irq_Temp0_over_high_level	Tempsens0 The temperature is greater than or equal to the high temperature critical value.
[5]	Irq_Temp1_over_high_level	Tempsens1 The temperature is greater than or equal to the high temperature critical value.
[6]	reserved	Reserved
[7]	reserved	Reserved
[8]	Irq_Temp0_under_low_level	Tempsens0 The temperature is less than or equal to the low temperature critical value.
[9]	Irq_Temp1_under_low_level	Tempsens1 The temperature is less than or equal to the low temperature critical value.
[10]	reserved	Reserved
[11]	reserved	Reserved
[12]	Irq_Temp0_over_high_cont	The number of times the Tempsens0 temperature is greater than or equal to the high temperature critical value has reached the value of reg_tempsen_ovhl_cnt_to_irq.
[13]	Irq_Temp1_over_high_cont	The number of times the Tempsens1 temperature is greater than or equal to the high temperature critical value has reached the value of reg_tempsen_ovhl_cnt_to_irq.
[14]	reserved	Reserved
[15]	reserved	Reserved
[16]	Irq_Temp0_under_low_cont	The number of times the temperature of Tempsens0 is less than or equal to the low temperature critical value has reached the value of reg_tempsen_udll_cnt_to_irq.
[17]	Irq_Temp1_under_low_cont	The number of times the temperature of Tempsens1 is less than or equal to the low temperature critical value has reached the value of reg_tempsen_udll_cnt_to_irq.
[18]	reserved	Reserved
[19]	reserved	Reserved
[20]	Irq_Temp0_overheat	The temperature of Tempsens0 is greater than or equal to the overheating critical value.
[21]	Irq_Temp1_overheat	The temperature of Tempsens1 is greater than or equal to the overheating critical value.
[22]	reserved	Reserved
[23]	reserved	Reserved

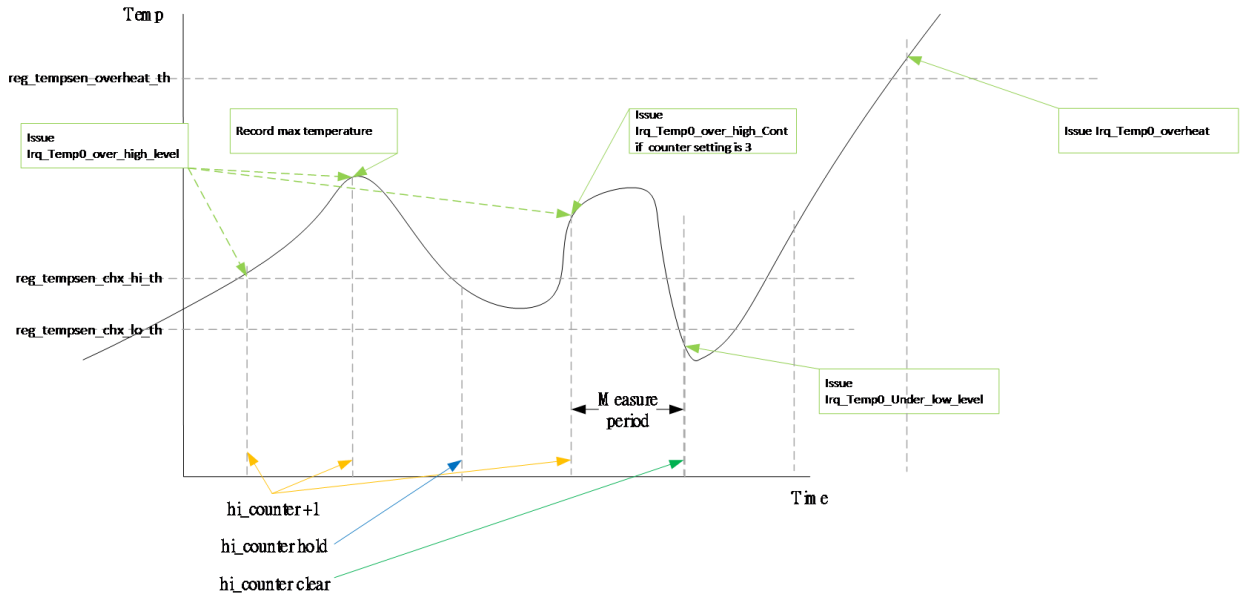


Diagram 21.43: Temperature measurement time, count and interrupt occurrence relationship diagram

- View temperature measurement results: sta_tempsen_chX_result records the previously completed temperature measurement results, sta_tempsen_chX_max_result records the maximum temperature value ever measured, tempsen_chX_temp_th_cnt records the number of consecutive high and low temperatures.
- Configure the overheat protection temperature reg_tempsen_overheat_th, overheat reset request countdown time reg_tempsen_overheat_cycle and enable reg_overheat_reset_en.

Refer to the RTC register chapter again to configure the RTC registers hw_thm_shdn_en, RTC_EN_THM_SHDN, RTC_THM_SHDN_AUTO_REBOOT to enable overheating to trigger power off or restart. When overheating occurs, the tempsensor controller will first issue an interrupt and start counting down. When sta_tempsen_overheat_countdown is equal to 1, the RTC's power-off protection request will be triggered. If the software intervenes before and performs temperature control, you can configure reg_overheat_reset_clr to clear the countdown. However, when the next measurement result is still overheated, the overheat protection interrupt will still be triggered and countdown again.

21.8.3 Temperature sensor Register Overview

Table 21.246: BassAddress: 0x030A0000

Name	Address Offset	Description
tempsen_version	0x000	ip version number
tempsen_ctrl	0x004	control register
tempsen_status	0x008	staus register
tempsen_set	0x00c	temperature sensor macro setting
tempsen_intr_en	0x010	interrupt enable
tempsen_intr_clr	0x014	interrupt clear
tempsen_intr_sta	0x018	interupt status
tempsen_intr_raw	0x01c	interrupt raw status
tempsen_ch0_result	0x020	temperature sensor channel 0 result
tempsen_ch1_result	0x024	temperature sensor channel 1 result
tempsen_ch0_temp_th	0x040	temperature sensor channel 0 threshold
tempsen_ch1_temp_th	0x044	temperature sensor channel 1 threshold
Overheat_th	0x060	overheat threshold register
tempsen_auto_period	0x064	auto sample setting register
tempsen_overheat_ctrl	0x068	overheat control register
tempsen_overheat_countdown	0x06c	overheat status register
tempsen_ch0_temp_th_cnt	0x070	counter of channel 0 over/under threshold event
tempsen_ch1_temp_th_cnt	0x074	counter of channel 1 over/under threshold event

21.8.4 Temperature sensor Register Description

21.8.4.1 tempsen_version

Table 21.247: tempsen_version, BassAddress: 0x000

Bits	Name	Access	Description	Reset
31:0	reg_ip_version	RO	verision 1.0	

21.8.4.2 tempsen_ctrl

Table 21.248: tempsen_ctrl, BassAddress: 0x004

Bits	Name	Access	Description	Reset
0	reg_tempsen_en	R/W	when re_tempsen_en is set , tempsen start to measure the channel set in reg_tempsen_sel	0x0
3:1	Reserved			
7:4	reg_tempsen_sel	R/W	temperature sense channel selection	0x0
15:8	Reserved			
23:16	reg_tempsen_ovhl_cnt_to_irq	R/W	counting threshold of high temperature	0x8
31:24	reg_tempsen_udll_cnt_to_irq	R/W	counting threshold of low temperature	0x8

21.8.4.3 tempsen_status

Table 21.249: tempsen_ctrl, BassAddress: 0x008

Bits	Name	Access	Description	Reset
0	sta_tempsen_busy	RO	busy status rise when re_tempsen_en is set	
31:1	Reserved			

21.8.4.4 tempsen_set

Table 21.250: tempsen_set, BassAddress: 0x00c

Bits	Name	Access	Description	Reset
0	reg_tempsen_bgen	R/W	sensor macro bandgap enable	0x0
1	reg_tempsen_chopen	R/W	sensor macro chopper function enable	0x1
2	reg_tempsen_choppol	R/W	sensor macro chopper polarity when CHOPEN=0	0x1
3	reg_tempsen_clkpol	R/W	sensor macro clock polarity when DA_TEMPSEN_EN=0	0x1
5:4	reg_tempsen_chopsel	R/W	sensor macro chop period, 0:128T, 1:256T, 2:512T, 3:1024T	0x2
7:6	reg_tempsen_accsel	R/W	sensor macro accumulate period, 0:512T, 1:1024T, 2:2048T, 3:4096T	0x1
15:8	reg_tempsen_cyc_clkdiv	R/W	clock divider for sensor macro, freq = ip_clk/(1+clk_div) , default is 25M/12 =2.083M , T = 0.48us	0xB
17:16	reg_tempsen_tsel	R/W	sensor macro test selection, please keep 0	0x0
18	reg_tempsen_en_bjt_test	R/W	sensor macro test selection, please keep 0	0x0
31:19	Reserved			

21.8.4.5 tempsen_intr_en

Table 21.251: tempsen_intr_en, BassAddress: 0x010

Bits	Name	Access	Description	Reset
31:0	sta_tempsen_intr_en	R/W	interrupt enable	0x0

21.8.4.6 tempsen_intr_clr

Table 21.252: tempsen_intr_clr, BassAddress: 0x014

Bits	Name	Access	Description	Reset
31:0	sta_tempsen_intr_clr	RWC	interrupt clear	

21.8.4.7 tempsen_intr_sta

Table 21.253: tempsen_intr_clr, BassAddress: 0x018

Bits	Name	Access	Description	Reset
31:0	sta_tempsen_intr_sta	RO	interrupt masked status	

21.8.4.8 tempsen_intr_raw

Table 21.254: tempsen_intr_raw, BassAddress: 0x01c

Bits	Name	Access	Description	Reset
31:0	sta_tempsen_intr_raw	RO	interrupt raw status: [3:0] ch3~ch0 measurement finish [7:4] ch3~ch0 measurement result is higher than high temperature threshold [11:8] ch3~ch0 measurement result is lower than low temperature threshold [15:12] ch3~ch0's high temperature event count is more than threshold [19:16] ch3~ch0's low temperature event count is more than threshold [23:20] ch3~ch0 measurement result is higher than overheat temperature	

21.8.4.9 tempsen_ch0_result

Table 21.255: tempsen_intr_raw, BassAddress: 0x020

Bits	Name	Access	Description	Reset
12:0	sta_tempsen_ch0_result	RO	channel 0 current temperature measurement result	
15:13	Reserved			
28:16	sta_tempsen_ch0_max_result	RO	channel 0 max temperature measurement result	
30:29	Reserved			
31	clr_tempsen_ch0_max_result	RWC	write 1 to clear channel 0 max temperature measurement result	

21.8.4.10 tempsen_ch1_result

Table 21.256: tempsen_ch1_result, BassAddress: 0x024

Bits	Name	Access	Description	Reset
12:0	sta_tempsen_ch1_result	RO	channel 1 current temperature measurement result	
15:13	Reserved			
28:16	sta_tempsen_ch1_max_result	RO	channel 1 max temperature measurement result	
30:29	Reserved			
31	clr_tempsen_ch1_max_result	RWC	write 1 to clear channel 1 max temperature measurement result	

21.8.4.11 tempsen_ch0_temp_th

Table 21.257: tempsen_ch0_temp_th, BassAddress: 0x040

Bits	Name	Access	Description	Reset
12:0	reg_tempsen_ch0_hi_th	R/W	channel 0 high temperature threshold to trigger interrupt	0x0
15:13	Reserved			
28:16	reg_tempsen_ch0_lo_th	R/W	channel 0 low temperature threshold to trigger interrupt	0x0
31:29	Reserved			

21.8.4.12 tempsen_ch1_temp_th

Table 21.258: tempsen_ch1_temp_th, BassAddress: 0x044

Bits	Name	Access	Description	Reset
12:0	reg_tempsen_ch1_hi_th	R/W	channel 1 high temperature threshold to trigger interrupt	0x0
15:13	Reserved			
28:16	reg_tempsen_ch1_lo_th	R/W	channel 1 low temperature threshold to trigger interrupt	0x0
31:29	Reserved			

21.8.4.13 Overheat_th

Table 21.259: Overheat_th, BassAddress: 0x060

Bits	Name	Access	Description	Reset
12:0	reg_tempsen_overheat_th	R/W	overheat temperature threshold	0x0
31:13	Reserved			

21.8.4.14 tempsen_auto_period

Table 21.260: tempsen_auto_period, BassAddress: 0x064

Bits	Name	Access	Description	Reset
23:0	reg_tempsen_auto_cycle	R/W	auto measure period. T_measure = reg_tempsen_auto_cycle*T_prediv	0x0
31:24	reg_tempsen_auto_prediv	R/W	a predivider setting for auto measure period. T_prediv = (25M/(reg_tempsen_auto_prediv+1))	0x18

21.8.4.15 tempsen_overheat_ctrl

Table 21.261: tempsen_overheat_ctrl, BassAddress: 0x068

Bits	Name	Access	Description	Reset
29:0	reg_tempsen_overheat_cycle	R/W	After overheat event happens, the cycle count will be load to a counter and trigger counting down. when counting down to 1, a reset signal will be issue to power control unit.	0x10 0000
30	reg_overheat_reset_clr	RWC	write 1 to stop overheat reset counting.	
31	reg_overheat_reset_en	R/W	enable overheat reset counting down.	0x0

21.8.4.16 tempsen_overheat_countdown

Table 21.262: tempsen_overheat_countdown, BassAddress: 0x06c

Bits	Name	Access	Description	Reset
29:0	sta_tempsen_overheat_countdown	RO	overheat reset counter	
30	Reserved			
31	sta_overheat_reset	RO	overheat reset signal status	

21.8.4.17 tempsen_ch0_temp_th_cnt

Table 21.263: tempsen_ch0_temp_th_cnt, BassAddress: 0x070

Bits	Name	Access	Description	Reset
7:0	sta_ch0_over_hi_temp_th_cnt	RO	channel 0 high temperature event count status	
15:8	sta_ch0_under_lo_temp_th_cnt	RO	channel 0 low temperature event count status	
16	reg_ch0_temp_th_cnt_clr	RWC	write 1 to clear channel 0 temperature event count	
31:17	Reserved			

21.8.4.18 tempsen_ch1_temp_th_cnt

Table 21.264: tempsen_ch1_temp_th_cnt, BassAddress: 0x074

Bits	Name	Access	Description	Reset
7:0	sta_ch1_over_hi_temp_th_cnt	RO	channel 1 high temperature event count status	
15:8	sta_ch1_under_lo_temp_th_cnt	RO	channel 1 low temperature event count status	
16	reg_ch1_temp_th_cnt_clr	RWC	write 1 to clear channel 1 temperature event count	
31:17	Reserved			

21.9 PWM

21.9.1 Overview

The chip provides 4 PWM controllers PWM0, PWM1, PWM2 and PWM3.

Each controller provides 4 independent PWM signal outputs. They are:

- PWM0 includes PWM[0], PWM[1], PWM[2], PWM[3].
- PWM1 includes PWM[4], PWM[5], PWM[6], PWM[7].
- PWM2 includes PWM[8], PWM[9], PWM[10], PWM[11].
- PWM3 includes PWM[12], PWM[13], PWM[14], PWM[15].

21.9.2 Features

The PWM clock source can be selected from 100MHz or 148.5MHz, and the default is 100MHz.

- There is an internal 30-bit counter, the output period and the number of high/low level beats are configurable.
- Supports up to 50MHz (100MHz/2) or 74.25MHz (148.5MHz/2) output, and the lowest is about 0.093Hz (100MHz/(2^30-1)) or 0.138Hz (148.5MHz/(2^32-1)) .
- Supports two modes: continuous output (PWMMODE = 0) and fixed pulse number output (PWMMODE = 1).
- Supports 4-channel PWM synchronous output mode (SHIFTMODE = 1), and the phase difference of the 4-channel PWM output can be controlled through the configuration register.

21.9.3 Way of Working

The basic configuration process of PWM is as follows (taking PWM[0] as an example):

1. Based on the selected clock source, calculate the square wave period and low-level beat number to be output.
2. Write the corresponding values to registers HLPERIOD0 and PERIOD0.
3. To operate in continuous output mode, configure PWMMODE to 0, set PWMSTART[0] to 1, and PWM[0] starts to output until PWMSTART[0] is set to 0 to end the output.
4. If you want to operate in the fixed pulse number output mode, configure PWMMODE as 1, and write the number of square waves to be output into the register PCOUNT0. After setting PWMSTART[0] to 1, PWM[0] starts to output and ends automatically after reaching the set square wave number, and the status register PWMDONE changes from 0 to 1.

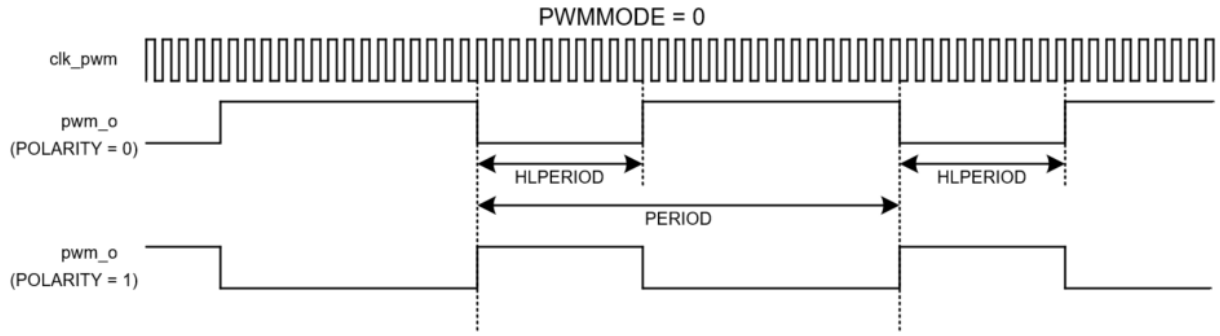


Diagram 21.44: PWM Continuous mode

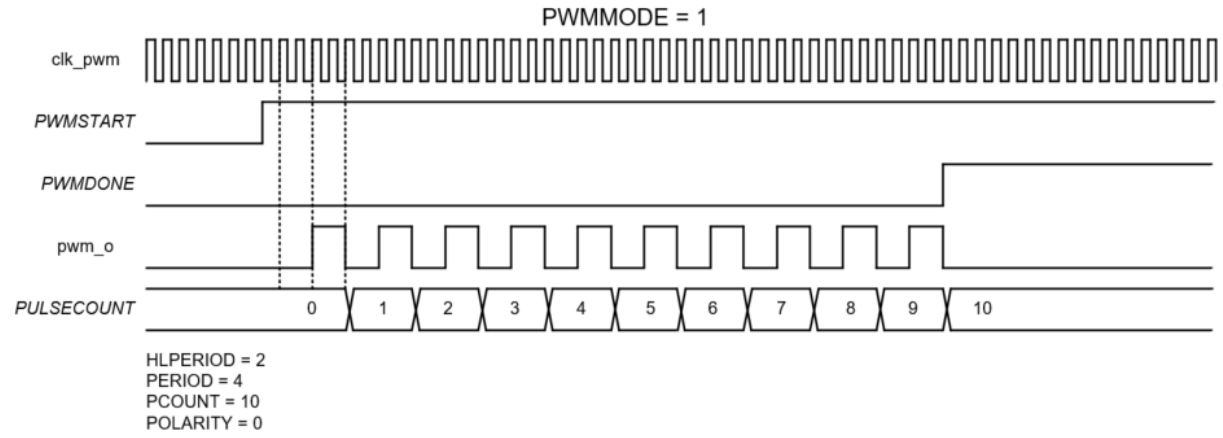


Diagram 21.45: PWM Pulse count mode

For example: To output a waveform with a frequency of 1MHz, a low level ratio of 75%, and a pulse number of 16

1. Using the default 100MHz clock source, the period number (PERIOD0) is configured as 100MHz / 1MHz = 100, and the low level number (HLPERIOD0) is configured as 100 x 75% = 75. The number of pulses (PCOUNT0) is configured as 16.
2. Write 1 to PWMSTART[0] to start outputting the waveform.
3. Read the register PWMDONE[0] until the value is 1, indicating that the output is completed.
4. The register PULSECOUNT0 can be read to confirm the output pulse number status value.

To enable PWM again, you need to write 0 and then 1 to PWMSTART[0] to reset the counter and status register.

When the 4-channel PWM is to operate in synchronous output mode, first configure SHIFTMODE to 1. The process is as follows:

1. Configure HLPERIOD0/PERIOD0, HLPERIOD1/PERIOD1, HLPERIOD2/PERIOD2, HLPERIOD3/PERIOD3 to the same value.
2. According to the phase difference that the four square wave waveforms need to be staggered, configure appropriate values into the registers SHIFTCOUNT0, SHIFTCOUNT1, SHIFTCOUNT2, and SHIFTCOUNT3.
3. Configure PWMSTART[3:0] to 4'hF, and set SHIFTSTART to 1. The 4-channel counter will start counting at the same time, and output the n-th PWM waveform when the counter value is equal to SHIFTCOUNTn.

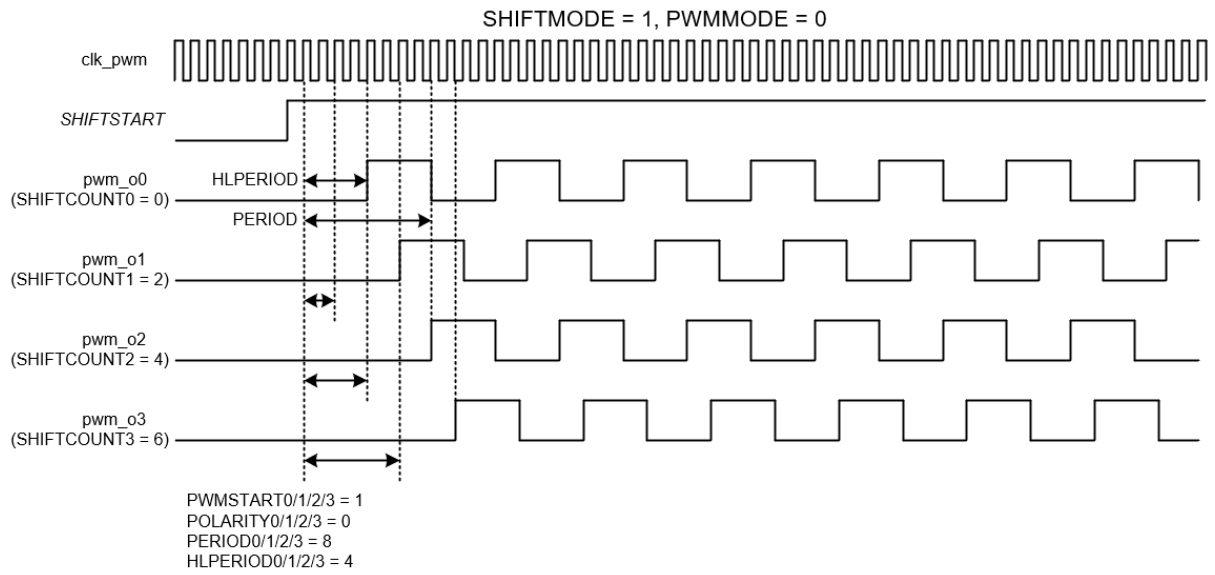


Diagram 21.46: PWM Continuous Shift Mode

For example: To output 4 channels of square waves with a frequency of 1KHz and a low level of 75% at the same time, the waveforms of each channel should be staggered by 1/4 cycle.

1. Using the default 100MHz clock source, the cycle number is configured as 100MHz / 1KHz = 100,000, and the low level number is configured as 100,000 x 75% = 75,000.
2. Configure SHIFTCOUNT0 = 0, SHIFTCOUNT1 = 100,000 x 1/4 = 25,000, SHIFTCOUNT2 = 100,000 x 2/4 = 50,000, SHIFTCOUNT3 = 100,000 x 3/4 = 75,000.
3. Set PWMSTART[3:0] to 4'hF, and set SHIFTSTART to 1, and the 4 PWM channels will output the first pulse in sequence.

Set SHIFTSTART to 0 to end the output, and read the register PWMDONE[3:0] until the value is 4'hF, which means that all 4 channels are output.

21.9.4 PWM Register Overview

An overview of the PWM registers is shown in table *PWM Registers Overview*. Here take PWM0 controller as an example, the other 3 controllers are similar.

Table 21.265: PWM Registers Overview

Name	Address Offset	Description
HLPERIOD0	0x000	PWM[0] low -level shooting number
PERIOD0	0x004	PWM[0] square wave cycle shooting
HLPERIOD1	0x008	PWM[1] low -level shooting number
PERIOD1	0x00c	PWM[1] square wave cycle shooting
HLPERIOD2	0x010	PWM[2] low -level shooting number
PERIOD2	0x014	PWM[2] square wave cycle shooting
HLPERIOD3	0x018	PWM[3] low -level shooting number
PERIOD3	0x01c	PWM[3] square wave cycle shooting
POLARITY	0x040	PWM mode settings
PWMSTART	0x044	Make PWM output
PWMDONE	0x048	PWM Ending State
PWMUPDATE	0x04c	Dynamic loading PWM cycle parameter
PCOUNT0	0x050	Set PWM[0] pulse number
PCOUNT1	0x054	Set PWM[1] pulse number
PCOUNT2	0x058	Set PWM[2] pulse number
PCOUNT3	0x05c	Set PWM[3] pulse number
PULSECOUNT0	0x060	PWM[0] output pulse meter status
PULSECOUNT1	0x064	PWM[1] output pulse meter status
PULSECOUNT2	0x068	PWM[2] output pulse meter status
PULSECOUNT3	0x06c	PWM[3] output pulse meter status
SHIFTCOUNT0	0x080	Synchronous mode PWM[0] initial difference
SHIFTCOUNT1	0x084	Synchronous mode PWM[1] initial difference
SHIFTCOUNT2	0x088	Synchronous mode PWM[2] initial difference
SHIFTCOUNT3	0x08c	Synchronous mode PWM[3] initial difference
SHIFTSTART	0x090	PWM synchronization mode enable
PWM_OE	0x0d0	PWM IO output enable

21.9.5 PWM Register Description

Here take PWM0 controller as an example, the other 3 controllers are similar.

21.9.5.1 HLPERIOD0

Table 21.266: HLPERIOD0, Offset Address: 0x000

Bits	Name	Access	Description	Reset
29:0	HLPERIOD0	R/W	PWM[0] Number of low level taps (in clk_pwm) When POLARITY[0] is 0, this value is the number of low level beats, when POLARITY[0] is 1, this value is the number of high level beats.	0x1

21.9.5.2 PERIOD0

Table 21.267: PERIOD0, Offset Address: 0x004

Bits	Name	Access	Description	Reset
29:0	PERIOD0	R/W	PWM[0] Square wave period beats (in clk_pwm) Note The PERIOD value must be greater than the HLPERIOD value.	0x2

21.9.5.3 HLPERIOD1

Table 21.268: PERIOD0, Offset Address: 0x008

Bits	Name	Access	Description	Reset
29:0	HLPERIOD1	R/W	PWM[1] Number of low level taps (in clk_pwm) When POLARITY[1] is 0, this value is the number of low level beats, when POLARITY[1] is 1, this value is the number of high level beats.	0x1

21.9.5.4 PERIOD1

Table 21.269: PERIOD1, Offset Address: 0x00c

Bits	Name	Access	Description	Reset
29:0	PERIOD1	R/W	PWM[1] Square wave period beats (in clk_pwm) Note The PERIOD value must be greater than the HLPERIOD value.	0x2

21.9.5.5 HLPERIOD2

Table 21.270: PERIOD1, Offset Address: 0x010

Bits	Name	Access	Description	Reset
29:0	HLPERIOD2	R/W	PWM[2] Number of low level taps (in clk_pwm) When POLARITY[2] is 0, this value is the number of low level beats, when POLARITY[2] is 1, this value is the number of high level beats.	0x1

21.9.5.6 PERIOD2

Table 21.271: PERIOD2, Offset Address: 0x014

Bits	Name	Access	Description	Reset
29:0	PERIOD2	R/W	PWM[2] Square wave period beats (in clk_pwm) Note The PERIOD value must be greater than the HLPERIOD value.	0x2

21.9.5.7 HLPERIOD3

Table 21.272: HLPERIOD3, Offset Address: 0x018

Bits	Name	Access	Description	Reset
29:0	HLPERIOD3	R/W	PWM[3] Number of low level taps (in clk_pwm) When POLARITY[3] is 0, this value is the number of low level beats, when POLARITY[3] is 1, this value is the number of high level beats.	0x1

21.9.5.8 PERIOD3

Table 21.273: HLPERIOD3, Offset Address: 0x01c

Bits	Name	Access	Description	Reset
29:0	PERIOD3	R/W	PWM[3] Square wave period beats (in clk_pwm) Note The PERIOD value must be greater than the HLPERIOD value.	0x2

21.9.5.9 POLARITY

Table 21.274: POLARITY, Offset Address: 0x040

Bits	Name	Access	Description	Reset
3:0	POLARITY	R/W	PWM[0]~[3] signal polarity [n] = 0: PWM[n] Default is low level output [n] = 1: PWM[n] Default is high level output	0x0
7:4	Reserved			
11:8	PWMMODE	R/W	PWM[0]~[3] operating mode [n+8] = 0: PWM[n] Operation in Continuous Out Mode [n+8] = 1: PWM[n] Operation in fixed output mode	0x0
15:12	Reserved			
16	SHIFTMODE	R/W	Enable PWM synchronous phase output mode 0 = PWM[0]~[3] operate in general mode 1 = PWM[0]~[3] operate in 4-way synchronised output mode	0x0
19:17	Reserved			
20	pclk_force_en	R/W	APB Clock Gating Control 0 = Enable APB clock gating to automatically turn off the clock when turn off the clock when idle 1 = APB clock held constant on	0x0
31:21	Reserved			

21.9.5.10 PWMSTART

Table 21.275: PWMSTART, Offset Address: 0x044

Bits	Name	Access	Description	Reset
3:0	PWMSTART	R/W	Enable PWM[0]~[3] [n] = 0: Stop PWM[n] [n] = 1: Output PWM[n] When PWMMODE is 0, write bit n to 0 and then write 1 to start PWM[n] output, until bit n is written to 0 to stop output. When PWMMODE is 1, write bit n to 1 to start PWM[n] output, and stop output automatically when the number of pulses output output equals to the value of PCOUNTn. When SHIFTMODE is set to 1, PWM-START[3:0] will be output enable for PWM[0]~[3]. Start of PWM will be controlled by SHIFTSTART.	0x0
31:4	Reserved			

21.9.5.11 PWMDONE

Table 21.276: PWMDONE, Offset Address: 0x048

Bits	Name	Access	Description	Reset
3:0	PWMDONE	RO	PWM[0]~[3] End output state [n] = 1: PWMn Closed Output When PWMSTART[n] is set to 1 from 0, the register value is cleared to 0.	
31:4	Reserved			

21.9.5.12 PWMUPDATE

Table 21.277: PWMUPDATE, Offset Address: 0x04c

Bits	Name	Access	Description	Reset
3:0	PWMUPDATE	R/W	Dynamically loaded PWM parameters When PWMSTART is written from 0 to 1, the register values (HLPERIODn,PERIODn) are held temporarily inside the PWM. If you want to change the waveform dynamically in the PWM output, write the new value to HLPERIODn and PERIODn, then write 1 to PWMUPDATE[n] and then write 0 to make new value effective.	0x0
31:4	Reserved			

21.9.5.13 PCOUNT0

Table 21.278: PCOUNT0, Offset Address: 0x050

Bits	Name	Access	Description	Reset
23:0	PCOUNT0	R/W	Number of PWM[0] pulses (set value must be greater than 0) Valid only when PWMMODE[0] is set to 1.	0x1
31:24	Reserved			

21.9.5.14 PCOUNT1

Table 21.279: PCOUNT1, Offset Address: 0x054

Bits	Name	Access	Description	Reset
23:0	PCOUNT1	R/W	Number of PWM[1] pulses (set value must be greater than 0) Valid only when PWMMODE[1] is set to 1.	0x1
31:24	Reserved			

21.9.5.15 PCOUNT2

Table 21.280: PCOUNT2, Offset Address: 0x058

Bits	Name	Access	Description	Reset
23:0	PCOUNT2	R/W	Number of PWM[2] pulses (set value must be greater than 0) Valid only when PWMMODE[2] is set to 1.	0x1
31:24	Reserved			

21.9.5.16 PCOUNT3

Table 21.281: PCOUNT3, Offset Address: 0x05c

Bits	Name	Access	Description	Reset
23:0	PCOUNT3	R/W	Number of PWM[3] pulses (set value must be greater than 0) Valid only when PWMMODE[3] is set to 1.	0x1
31:24	Reserved			

21.9.5.17 PULSECOUNT0

Table 21.282: PULSECOUNT0, Offset Address: 0x060

Bits	Name	Access	Description	Reset
23:0	PULSECOUNT0	RO	PWM[0] Number of Output Pulses Status	
31:24	Reserved			

21.9.5.18 PULSECOUNT1

Table 21.283: PULSECOUNT1, Offset Address: 0x064

Bits	Name	Access	Description	Reset
23:0	PULSECOUNT1	RO	PWM[1] Number of Output Pulses Status	
31:24	Reserved			

21.9.5.19 PULSECOUNT2

Table 21.284: PULSECOUNT2, Offset Address: 0x068

Bits	Name	Access	Description	Reset
23:0	PULSECOUNT2	RO	PWM[2] Number of Output Pulses Status	
31:24	Reserved			

21.9.5.20 PULSECOUNT3

Table 21.285: PULSECOUNT3, Offset Address: 0x06c

Bits	Name	Access	Description	Reset
23:0	PULSECOUNT3	RO	PWM[3] Number of Output Pulses Status	
31:24	Reserved			

21.9.5.21 SHIFTCOUNT0

Table 21.286: SHIFTCOUNT0, Offset Address: 0x080

Bits	Name	Access	Description	Reset
23:0	SHIFTCOUNT0	R/W	PWM[0] Phase difference of the first pulse output (in clk_pwm) Valid only when SHIFTMODE is set to 1.	0x0
31:24	Reserved			

21.9.5.22 SHIFTCOUNT1

Table 21.287: SHIFTCOUNT1, Offset Address: 0x084

Bits	Name	Access	Description	Reset
23:0	SHIFTCOUNT1	R/W	PWM[1] Phase difference of the first pulse output (in clk_pwm) Valid only when SHIFTMODE is set to 1.	0x0
31:24	Reserved			

21.9.5.23 SHIFTCOUNT2

Table 21.288: SHIFTCOUNT2, Offset Address: 0x088

Bits	Name	Access	Description	Reset
23:0	SHIFTCOUNT2	R/W	PWM[2] Phase difference of the first pulse output (in clk_pwm) Valid only when SHIFTMODE is set to 1.	0x0
31:24	Reserved			

21.9.5.24 SHIFTCOUNT3

Table 21.289: SHIFTCOUNT3, Offset Address: 0x08c

Bits	Name	Access	Description	Reset
23:0	SHIFTCOUNT3	R/W	PWM[3] Phase difference of the first pulse output (in clk_pwm) Valid only when SHIFTMODE is set to 1.	0x0
31:24	Reserved			

21.9.5.25 SHIFTSTART

Table 21.290: SHIFTSTART, Offset Address: 0x090

Bits	Name	Access	Description	Reset
0	SHIFTSTART	R/W	Enable PWM output in synchronous mode When SHIFTMODE is set to 1, this register starts to output PWM[0]~[3] after writing 1.	0x0
31:1	Reserved			

21.9.5.26 PWM_OE

Table 21.291: PWM_OE, Offset Address: 0x0d0

Bits	Name	Access	Description	Reset
3:0	PWM_OE	R/W	PWM[0]~[3] IO output enable 1 = IO is output, 0 = IO is input.	0x0
31:4	Reserved			

21.10 KeyScan

21.10.1 Overview

Keyscan supports a matrix of up to 8 x 8 = 64 keys. If you don't need so many keys, you can freely decide which rows or columns to mask or retain. Depending on the software needs, snapshot mode and FIFO mode can be selected to obtain key information.

21.10.2 Way of Working

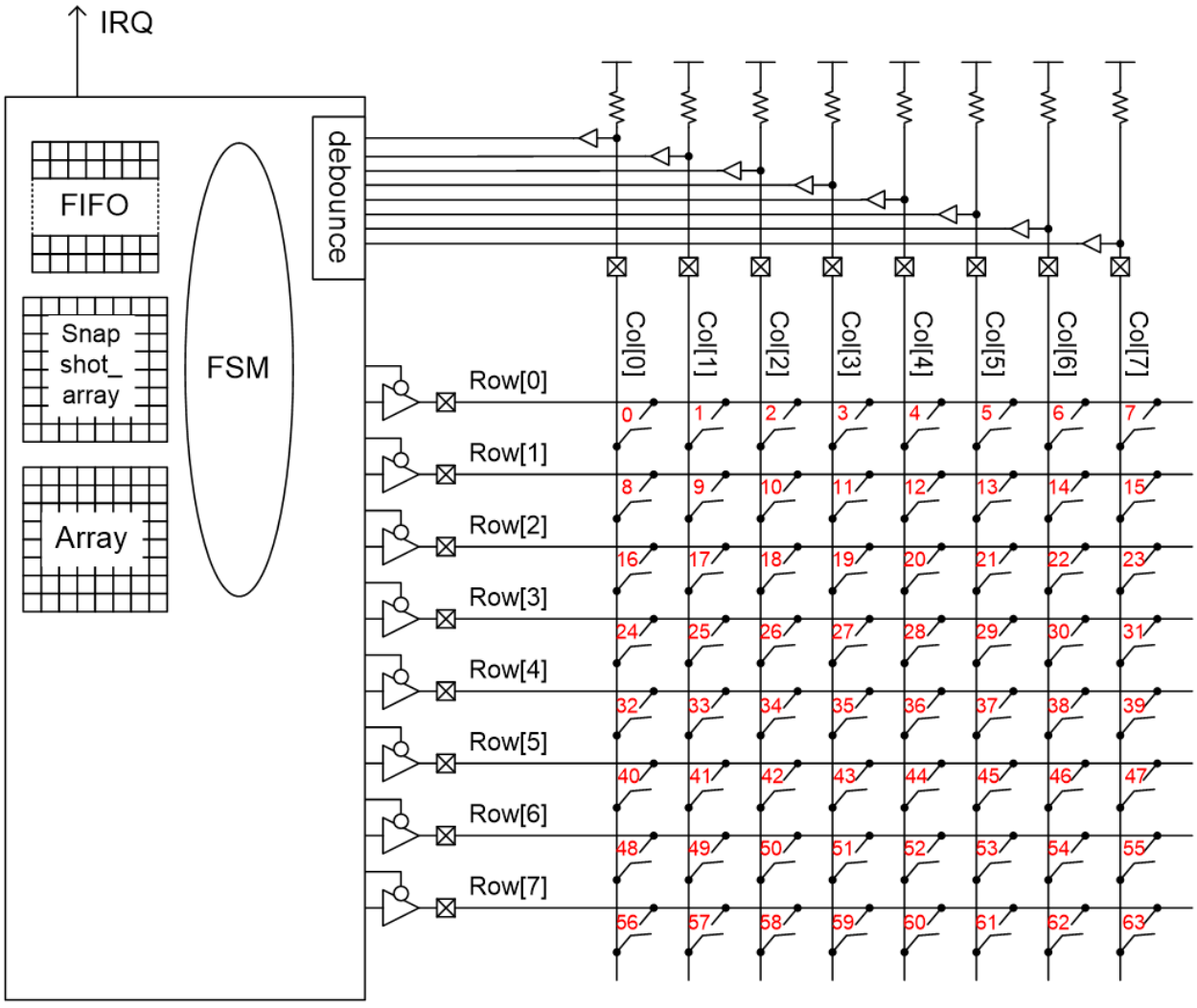


Diagram 21.47: Keyscan architecture block diagram

When the state machine (FSM) is in rest mode (no keys are pressed), all rows output 0, and col is in input mode with weak pull-up enabled (the weak pull-up is set in the register corresponding to ioblk, not in keyscan module). When any key is pressed, the col end will see a value other than all 1 after debounce, indicating that a key is pressed. At this time, FSM will start a scan, sequentially making Row [0] -> Row [7] output 0 only one bit at a time (the rest are in the HiZ high-impedance state). Each result will be updated into an array.

FSM will continue to scan in a loop until the col returned by all rows is all 1, indicating that the keys are not pressed, and then it will enter the rest mode again (all rows output 0).

21.10.3 Basic Settings

The reg_row_mask, reg_col_mask and reg_enable in KEYSCAN0 allow you to selectively block certain IOs without outputting or referring to their inputs when the 8x8 matrix is not used. Default is fully off. So it needs to be opened. reg_db_col in KEYSCAN_CONFIG2 determines how long the column input needs to debounce before it can be used.

The reg_slow_div in KEYSCAN_CONFIG1 determines the stay time of each stage of IP's FSM. Remember that this number must be larger than the debounce time, otherwise the interpretation will be confusing if the debounce is not completed after the IO transition.

reg_wait_cntr in KEYSCAN_CONFIG3 can be used to reduce the scanning speed. Because as long as the key is pressed, the keyscan module will continue to scan. This counter can control a fixed waiting time before starting a new round of scanning to reduce the scanning frequency.

21.10.4 Use FIFO mode

When using FIFO mode, the 64 key values scanned in by IP will be stored in the array. Every time the status of any key is different from the last scanned content, it will push the key's index and current value (0/1) into the FIFO. So the number in [5:0] specifies which button it is. [6] Indicates whether to press (0) or release (1). When the FIFO is not empty, an IRQ is issued. The advantage of this mode is that it eliminates the need for software to check which bit has changed bit by bit. The disadvantage is that KEY_SCAN_FIFO is a register that will automatically pop when read, so be careful when operating it.

Turn on reg_irq_fifo_not_empty_enable of KEYSCAN_IRQ_ENABLE. After receiving the IRQ, read reg_irq_fifo_not_empty in KEYSCAN_IRQ_FLAG, and then check reg_fifo_not_empty of KEYSCAN_FIFO_STATUS. Then start reading the contents of KEYSCAN_FIFO. Until it is cleared, clear KEY_SCAN_IRQ_CLEAR and end IRQ retine.

21.10.5 Use snapshot array mode

When using the snapshot array, the values of the 64 keys currently scanned in the IP will be stored in an array. If the content of this array is different from the content of KEYSCAN_SNAPSHOT_ARRAY, an IRQ will be sent. The software can trigger KEYSCAN_SNAPSHOT_TRIG, capture the current array content into the snapshot array, and then slowly compare what content has changed from the previous knowledge.

Turn on reg_irq_snapshot_change_enable of KEYSCAN_IRQ_ENABLE. After receiving the IRQ, read trigger KEYSCAN_SNAPSHOT_TRIG, and then interpret the contents of KEYSCAN_SNAPSHOT_ARRAY. Then clear KEY_SCAN_IRQ_CLEAR to end IRQ retine.

21.10.6 KeyScan Register Overview

Table 21.292: Key scan Registers Overview

Name	Address Offset	Description
KEYSCAN_CONFIG0	0x000	
KEYSCAN_CONFIG1	0x004	
KEYSCAN_CONFIG2	0x008	
KEYSCAN_CONFIG3	0x00c	
KEYSCAN_SNAPSHOT_ARRAY	0x014	
KEYSCAN_SNAPSHOT_TRIG	0x01c	
KEYSCAN_FIFO_STATUS	0x020	
KEYSCAN_FIFO	0x024	
KEYSCAN_IRQ_ENABLE	0x028	
KEYSCAN_IRQ_FLAG	0x02c	
KEYSCAN_IRQ_CLEAR	0x030	

21.10.7 KeyScan Register Description

21.10.7.1 KEYSCAN_CONFIG0

Table 21.293: KEYSCAN_CONFIG0, Offset Address: 0x000

Bits	Name	Access	Description	Reset
7:0	reg_row_mask	R/W	ROW[7:0] Mask 0 = enable 1 = disable	0xff
15:8	reg_col_mask	R/W	COL[7:0] Mask 0 = enable 1 = disable	0xff
16	reg_enable	R/W	keyscan enable 0 = disable 1 = enable	0x0
31:17	Reserved			

21.10.7.2 KEYSCAN_CONFIG0

Table 21.294: KEYSCAN_CONFIG1, Offset Address: 0x004

Bits	Name	Access	Description	Reset
23:0	reg_slow_div	R/W	slow divider (MUST BE BIGGER THAN reg_db_col) Each step is IP clock frequency divide by reg_slow_div Scan frequency = IP clock freq / ((reg_slow_div+1) * (9+reg_wait_count+1)) IDLE -> ROW0 -> ROW1 -> ROW2->ROW3 ->ROW4->ROW5->ROW6->RO W7->UPDATE->WAIT->IDLE	0xff
31:24	Reserved			

21.10.7.3 KEYSCAN_CONFIG2

Table 21.295: KEYSCAN_CONFIG2, Offset Address: 0x008

Bits	Name	Access	Description	Reset
15:0	reg_db_col	R/W	column input debounce counter (IP clock cycle)	0x64
31:16	Reserved			

21.10.7.4 KEYSCAN_CONFIG3

Table 21.296: KEYSCAN_CONFIG3, Offset Address: 0x00c

Bits	Name	Access	Description	Reset
7:0	reg_wait_cntr	R/W	wait interval between each scan (unit is reg_slow_div count)	0x10
31:8	Reserved			

21.10.7.5 KEYSCAN_SNAPSHOT_ARRAY

Table 21.297: KEYSCAN_SNAPSHOT_ARRAY, Offset Address: 0x014

Bits	Name	Access	Description	Reset
63:0	reg_cpu_snapshot_array	RO	CPU snapshot array result (0 = press, 1 = not press) [0] = Row 0 , Col 0 [1] = Row 0 , Col 1 … [7] = Row 0, Col 7 [8] = Row 1, Col 0 [63] = Row 7, Col 7 [N] = Row Y, Col X (N = Y*8 + X)	

21.10.7.6 KEYSCAN_SNAPSHOT_TRIG

Table 21.298: KEYSCAN_SNAPSHOT_TRIG, Offset Address: 0x01c

Bits	Name	Access	Description	Reset
0	reg_cpu_snapshot_toggle	W1T	Write 1 to Trigger snapshot array to update When current result is different from snapshot result, irq happen To solve the IRQ, write 1 to trigger the snapshot array to copy from current array and start checking which bit is different from previous state	
31:1	Reserved			

21.10.7.7 KEYSCAN_FIFO_STATUS

Table 21.299: KEYSCAN_FIFO_STATUS, Offset Address: 0x020

Bits	Name	Access	Description	Reset
3:0	reg_fifo_count	RO	FIFO content count 0 = Empty 1 = one content in FIFO N = N content in FIFO	
4	reg_fifo_not_empty	RO	FIFO not empty flag 0 = Empty 1 = Not empty	
31:5	Reserved			

21.10.7.8 KEYSCAN_FIFO

Table 21.300: KEYSCAN_FIFO, Offset Address: 0x024

Bits	Name	Access	Description	Reset
6:0	reg_fifo_rdata	ROC	read data from FIFO (Auto POP) - check FIFO empty-ness before read [6] 0 = press, 1 = not-press [5:0] = index Row = INT(index/8) Col = mod(index,8) 63 = Row 7 , Column 7 13 = Row 1 , Clumne 5	
31:7	Reserved			

21.10.7.9 KEYSCAN_IRQ_ENABLE

Table 21.301: KEYSCAN_IRQ_ENABLE, Offset Address: 0x028

Bits	Name	Access	Description	Reset
0	reg_irq_fifo_not_empty_enable	R/W	FIFO mode IRQ Enable 0 = Disable 1 = Enable	0x0
3:1	Reserved			
4	reg_irq_snapshot_change_enable	R/W	Snapshot mode IRQ Enable 0 = Disable 1 = Enable	0x0
31:5	Reserved			

21.10.7.10 KEYSCAN_IRQ_FLAG

Table 21.302: KEYSCAN_IRQ_FLAG, Offset Address: 0x02c

Bits	Name	Access	Description	Reset
0	reg_irq_fifo_not_empty	RO	FIFO not empty IRQ flag 0 = Empty 1 = Not empty	
3:1	Reserved			
4	reg_irq_snapshot_change	RO	Snapshot change IRQ flag 0 = No change 1 = Change	
31:5	Reserved			

21.10.7.11 KEYSCAN_IRQ_CLEAR

Table 21.303: KEYSCAN_IRQ_CLEAR, Offset Address: 0x030

Bits	Name	Access	Description	Reset
0	reg_irq_fifo_not_empty_clear_w1t	W1T	FIFO not empty IRQ Clear (Write 1 clear)	
3:1	Reserved			
4	reg_irq_snapshot_change_clear_w1t	W1T	Snapshot Change IRQ Clear (Write 1 clear)	
31:5	Reserved			

21.11 Wiegand

21.11.1 Overview

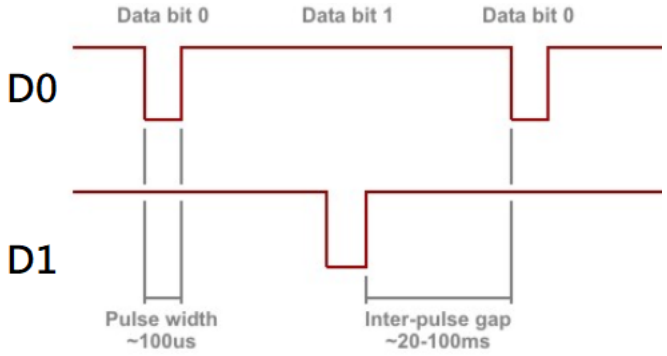


Diagram 21.48: The way wiegand signal bus transmits 0/1

The Wiegand interface uses two single-ended signals, D0/D1. When the bus is idle, it is always high. A low pulse appears on D0, which means a “0” is transmitted. A low pulse is found on D1, which means a “1” is transmitted.

Wiegand is commonly used in access control systems. There are two commonly used formats, Wiegand 26/34, which represent the number of bits in the packet respectively. An introduction to these two formats is as follows.

21.11.1.1 Wiegand 26

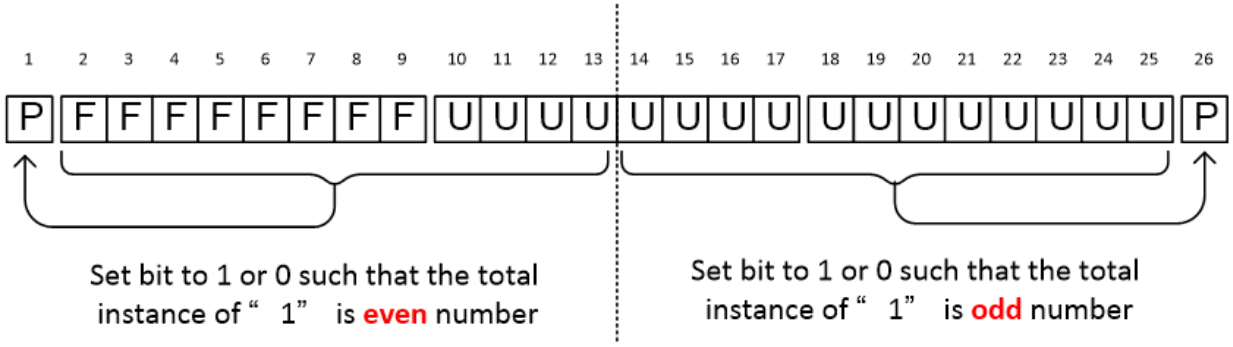


Diagram 21.49: Wiegand 26 format

21.11.1.2 Wiegand 34

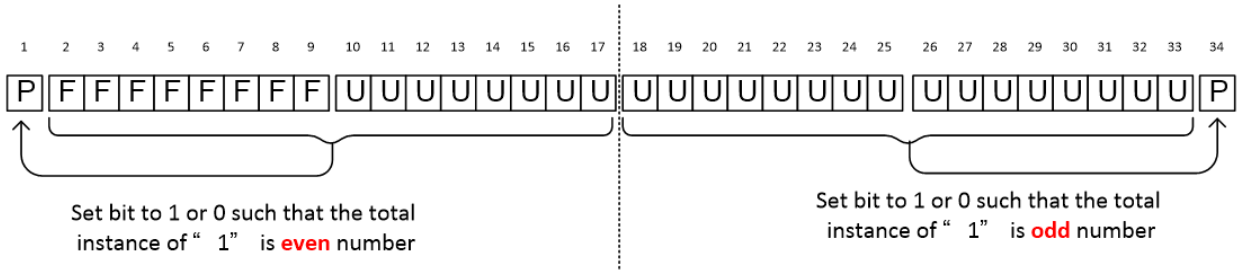


Diagram 21.50: Wiegand 34 format

F = Facility Code

U = User code

Some access cards have a string of numbers behind them. After converting them into hex,

Dec 0002262506 Hex : 22_85_EA

0x22 is Facility code (Dec = 34)

0x85EA is user code (Dec = 34282)

Diagram 21.51: Common magnetic cards, the digital meaning of magnetic buckles

The 34 and 34282 seen later are the Facility code & user code that were disassembled and expressed in decimal again.

Note: This IP TX RX does not handle parity insert or checking, which are all handled by software.

21.11.2 Way of Working

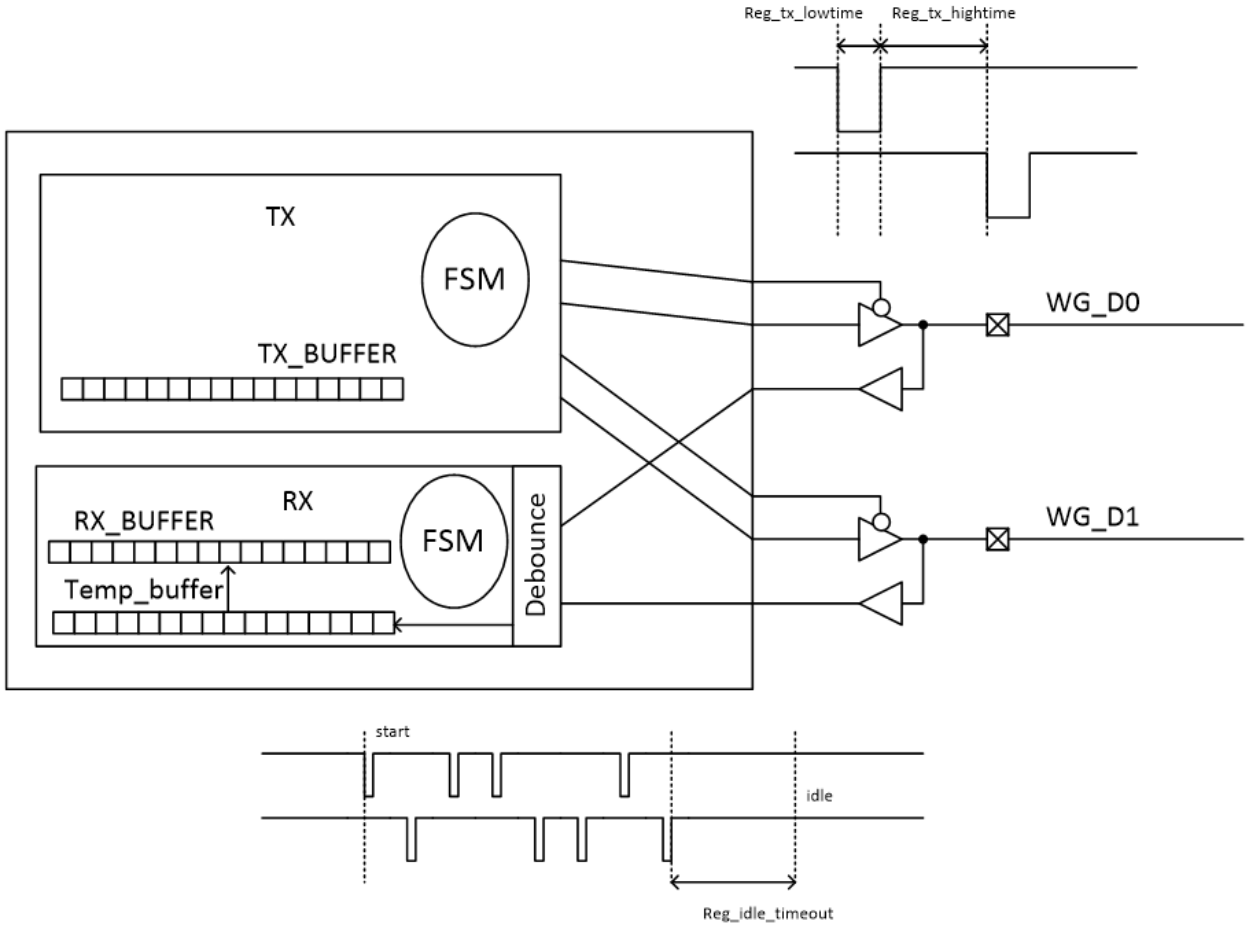


Diagram 21.52: wiegand architecture block diagram

The Wiegand module contains TX and RX and can be used in one or two directions. When TX goes out, RX will block it, so RX will not receive the signal it sends. The TX supports push pull mode or open drain mode.

21.11.2.1 TX

Before TX, set the high time and low time of TX, and the number of bits in the packet is determined by MSB 1st or LSB 1st. Then put the data in the TX_BUFFER register and use TX_TRIG to send the data.

After the complete TX transmission is completed, you can rely on the interrupt of TX finish or the status of polling TX_BUSY to determine when the next packet can be sent.

21.11.2.2 RX

Before RX, set the debounce time and the number of bits expected to receive the packet. When D0 D1 has a low pulse, RX starts pushing data into the temp buffer. When the number of received bits reaches the expected number of bits for a packet, the temp buffer will be pushed into RX_BUFFER, an interrupt will be issued, and the software will be asked to process it. The temp buffer continues to receive the next data.

If an idle timeout occurs on D0 D1, even if the number of bits has not been reached, it will be forced to be regarded as a packet.

The high bits of the RX BUFFER will record the total number of bits received by this packet, whether it is caused by timeout, or an overflow occurs before the software reads it.

Each time a packet is received, you can rely on the rx_buffer_rececived interrupt or RX_BUFFER_VALID to determine whether there is valid data in RX_BUFFER. After the RX Data is taken away, trigger RX_BUFFER_CLEAR to clear RX_BUFFER to receive the next packet.

21.11.3 Wiegand Register Overview

Table 21.304: Wiegand Registers Overview

Name	Address Offset	Description
TX_CONFIG0	0x000	
TX_CONFIG1	0x004	
TX_CONFIG2	0x008	
TX_BUFFER	0x00c	
TX_TRIG	0x014	
TX_BUSY	0x018	
TX_DEBUG	0x01c	
RX_CONFIG0	0x020	
RX_CONFIG1	0x024	
RX_CONFIG2	0x028	
RX_BUFFER	0x02c	
RX_BUFFER_VALID	0x038	
RX_BUFFER_CLEAR	0x03c	
RX_DEBUG	0x040	
IRQ_ENABLE	0x044	
IRQ_FLAG	0x048	
IRQ_CLEAR	0x04c	

21.11.4 Wiegand Register Description

21.11.4.1 TX_CONFIG0

Table 21.305: TX_CONFIG0, Offset Address: 0x000

Bits	Name	Access	Description	Reset
23:0	reg_tx_lowtime	R/W	TX Low width , unit = cycle	0xff
31:24	Reserved			

21.11.4.2 TX_CONFIG1

Table 21.306: TX_CONFIG1, Offset Address: 0x004

Bits	Name	Access	Description	Reset
23:0	reg_tx_hightime	R/W	TX High width , unit = cycle	0xff
31:24	Reserved			

21.11.4.3 TX_CONFIG2

Table 21.307: TX_CONFIG2, Offset Address: 0x008

Bits	Name	Access	Description	Reset
6:0	reg_tx_bitcount	R/W	TX Frame bit count per transmit , unit = bit	0x18
7	Reserved			
8	reg_tx_msb1st	R/W	TX Transmit from MSB or LSB 0 : LSB 1st , from tx_buffer[0] –> tx_b uffer[reg_tx_bitcount] 1 : MSB 1st , from tx_b uffer[reg_tx_bitcount] –> tx_buffer[0]	0x0
15:9	Reserved			
16	reg_tx_opendrain	R/W	TX using push-pull mode or opendrain mode 0 : push-pull mode 1 : opendrain mode	0x0
31:17	Reserved			

21.11.4.4 TX_BUFFER

Table 21.308: TX_BUFFER, Offset Address: 0x00c

Bits	Name	Access	Description	Reset
63:0	reg_tx_buffer	R/W	TX buffer content	0x00

21.11.4.5 TX_TRIG

Table 21.309: TX_TRIG, Offset Address: 0x014

Bits	Name	Access	Description	Reset
0	reg_tx_trig_w1t	W1T	Trigger transmittion Write 1 trigger (please check reg_tx_busy before transmit)	
31:1	Reserved			

21.11.4.6 TX_BUSY

Table 21.310: TX_BUSY, Offset Address: 0x018

Bits	Name	Access	Description	Reset
0	reg_tx_busy	RO	0 = idle, allow to trigger transmission 1 = busy, do not trigger any more tranmission or it will be ignore.	
31:1	Reserved			

21.11.4.7 TX_BUSY

Table 21.311: TX_DEBUG, Offset Address: 0x01c

Bits	Name	Access	Description	Reset
2:0	reg_tx_fsm	RO	TX Finite State Machine current state 0 : idle 1 : wait bus idle 2 : tx_start 3 : transmit low 4 : transmit high 5 : tx_stop	
7:3	Reserved			
14:8	reg_tx_pointer	RO	TX pointer current position indicate how many bit is still not yet send	
31:15	Reserved			

21.11.4.8 RX_CONFIG0

Table 21.312: RX_CONFIG0, Offset Address: 0x020

Bits	Name	Access	Description	Reset
15:0	reg_rx_debounce	R/W	RX input debounce time (unit is cycle)	0xff
31:16	Reserved			

21.11.4.9 RX_CONFIG1

Table 21.313: RX_CONFIG1, Offset Address: 0x024

Bits	Name	Access	Description	Reset
31:0	reg_idle_timeout	R/W	Bus timeout cycle count When bus is idle for idle_timeout cycle, bus is expected to be back to idle If some bit has received but not yet accumulate to rx_bitcount, it will also treat as a complete packet.	0 xfff

21.11.4.10 RX_CONFIG2

Table 21.314: RX_CONFIG2, Offset Address: 0x028

Bits	Name	Access	Description	Reset
6:0	reg_rx_bitcount	R/W	RX Expected Frame bit count , unit = bit	0x18
7	Reserved			
8	reg_rx_msb1st	R/W	RX Received sequence 0 : LSB 1st, 1st data is put in reg_rx _buffer[0]->[1]->[2].... 1 : MSB 1st, 1st data is put in reg_rx_buffer [reg_rx_bitcount]->[0]	0x0
11:9	Reserved			
12	reg_rx_enable	R/W	RX Enable 0 : disable 1 : Enable	0x0
31:13	Reserved			

21.11.4.11 RX_BUFFER

Table 21.315: RX_BUFFER, Offset Address: 0x02c

Bits	Name	Access	Description	Reset
72:0	reg_rx_buffer	RO	RX Buffer [63:0] = fifo = Indicate received content [70:64] = fifo_bit_count = How many effective bit is in rx_buffer[63:0] [71] = idle_reach = This RX is terminate by bus idle timeout [72] = overflow = This RX just overwrite an unread message	
95:73	Reserved			

21.11.4.12 RX_BUFFER_VALID

Table 21.316: RX_BUFFER_VALID, Offset Address: 0x038

Bits	Name	Access	Description	Reset
0	reg_rx_buffer_valid	RO	reg_rx_buffer validness 0 : not valid 1 : valid	
31:1	Reserved			

21.11.4.13 RX_BUFFER_CLEAR

Table 21.317: RX_BUFFER_CLEAR, Offset Address: 0x03c

Bits	Name	Access	Description	Reset
0	reg_rx_buffer_clear_w1t	W1T	reg_rx_buffer clear (write 1 clear)	
31:1	Reserved			

21.11.4.14 RX_DEBUG

Table 21.318: RX_DEBUG, Offset Address: 0x040

Bits	Name	Access	Description	Reset
0	reg_businidle	RO	bus in idle indication 0 : bus is not in idle 1 : bus is in idle more than reg_rx_idle_timeout cycle	
31:1	Reserved			

21.11.4.15 IRQ_ENABLE

Table 21.319: IRQ_ENABLE, Offset Address: 0x044

Bits	Name	Access	Description	Reset
0	reg_irq_tx_finish_enable	R/W	TX Finish IRQ Enable (to inform all data has being transmit, ready for next) 0 : Disable 1 : Enable	0x0
3:1	Reserved			
4	reg_irq_rx_overflow_enable	R/W	RX Overflow IRQ Enable 0 : Disable 1 : Enable	0x0
7:5	Reserved			
8	reg_irq_rx_received_enable	R/W	RX Received IRQ Enable 0 : Disable 1 : Enable	0x0
31:9	Reserved			

21.11.4.16 IRQ_FLAG

Table 21.320: IRQ_FLAG, Offset Address: 0x048

Bits	Name	Access	Description	Reset
0	reg_irq_tx_finish	RO	TX Finish IRQ Flag 0 : no IRQ 1 : IRQ (one tranmission has being completed)	
3:1	Reserved			
4	reg_irq_rx_overflow	RO	RX overflow IRQ Flag 0 : no IRQ 1 : IRQ (rx buffer is not pop and new data has overwrited)	
7:5	Reserved			
8	reg_irq_rx_received	RO	RX received IRQ Flag 0 : no IRQ 1 : RX buffer has new data	
31:9	Reserved			

21.11.4.17 IRQ_CLEAR

Table 21.321: IRQ_CLEAR, Offset Address: 0x04c

Bits	Name	Access	Description	Reset
0	reg_irq_tx_finish_clear_w1t	W1T	TX Finish IRQ Clear , Write 1 to clear reg_irq_tx_finish flag	
3:1	Reserved			
4	reg_irq_rx_overflow_clear_w1t	W1T	RX Overflow IRQ Clear . Write 1 to clear reg_irq_rx_overflow flag	
7:5	Reserved			
8	reg_irq_rx_received_clear_w1t	W1T	RX Received IRQ Clear . Write 1 to clear reg_irq_rx_received flag	
31:9	Reserved			

21.12 IRRX

21.12.1 Overview

Receives external infrared data via the IRRX unit.

21.12.2 Features

1. Support NEC encoding mode (including repeat code)
2. Support Philips RC5/RC6 encoding mode
3. Support Sony encoding mode
4. Support infrared wake-up function

21.12.3 Way of Working

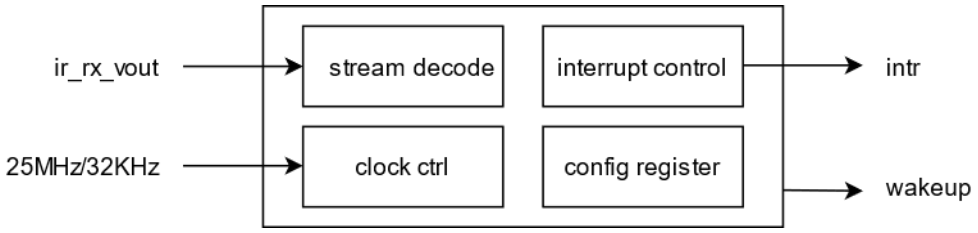


Diagram 21.53: working ways of IRRX

The software pre-sets the received infrared data format. When the IRRX module receives the infrared communication signal, it decodes it. The code that conforms to the predetermined format is transmitted to the CPU through an interrupt, and the CPU performs corresponding operations based on the code.

21.12.4 IRRX Register Overview

There is only 1 No-die Domain. Its base address (RTCSYS_IRRX) is: 0x0502E000.

Table 21.322: IRRX Registers Overview

Name	Address Offset	Description
IR_EN	0x000	
IR_MODE	0x004	
IR_CFG	0x008	
IR_FRAME	0x00c	
int_en	0x010	
int_clr	0x014	
int_msk	0x018	
int	0x01c	
int_raw	0x020	
IR_SYMBOL_CFG0	0x030	
IR_SYMBOL_CFG1	0x034	

continues on next page

Table 21.322 – continued from previous page

Name	Address Offset	Description
IR_SYMBOL_CFG2	0x038	
IR_SYMBOL_CFG3	0x03c	
IR_SYMBOL_CFG4	0x040	
IR_SYMBOL_CFG5	0x044	
IR_SYMBOL_CFG6	0x048	
IR_SYMBOL_CFG7	0x04c	
IR_CLOCK_CTRL	0x050	
IR_DATA0	0x080	
IR_DATA1	0x084	
IR_DATA2	0x088	
IR_DATA3	0x08c	
IR_DATA4	0x090	
IR_NEC_DATA0	0x0a8	
IR_SONY_DATA0	0x0ac	
IR_SONY_DATA1	0x0b0	
IR_PHILIPS_DATA0	0x0b4	
IR_PHILIPS_DATA1	0x0b8	
IR_PRD_REC0	0x0e0	
IR_PRD_REC1	0x0e4	
IR_PRD_REC2	0x0e8	
IR_PRD_REC3	0x0ec	
IR_PRD_REC4	0x0f0	
IR_PRD_REC5	0x0f4	
SPARE_0	0xff0	
SPARE_1	0xff4	
SPARE_RO	0xff8	
DATA_CODE	0xffc	

21.12.5 IRRX Register Description

21.12.5.1 IR_EN

Table 21.323: IR_EN, Offset Address: 0x000

Bits	Name	Access	Description	Reset
0	reg_ir_rx_en	R/W	ir receiver enable	0x0
1	reg_ir_rx_rst	R/W	ir receiver reset	0x0
15:2	Reserved			
16	reg_ir_init_done	RO	ir receiver ready	
31:17	Reserved			

21.12.5.2 IR_MODE

Table 21.324: IR_MODE, Offset Address: 0x004

Bits	Name	Access	Description	Reset
1:0	reg_ir_mode	R/W	ir receiver mode 0x0: pulse distance coding 0x2: bi-phase coding, RC5 0x3: bi-phase coding, RC6	0x0
7:2	Reserved			
8	reg_periodic_mode	R/W	periodic sample mode 0x1: periodic sample, reg_ic_mode is ignored	0x0
31:9	Reserved			

21.12.5.3 IR_CFG

Table 21.325: IR_CFG, Offset Address: 0x008

Bits	Name	Access	Description	Reset
7:0	reg_tick_prd	R/W	tick period	0x18
11:8	reg_sample_prd	R/W	sample period	0x9
14:12	reg_debounce	R/W	input signal debounce control	0x4
15	Reserved			
16	reg_import_inv	R/W	input signal polarity control	0x1
17	reg_export_inv	R/W	output data polarity control	0x0
18	reg_sony_format	R/W	tx transmit stop burst at frame end 0x1: SONY formate 0x0: NEC formate	0x0
19	reg_repeat_support	R/W	support simplfied repeat code	0x1
20	reg_bit_edge_sel	R/W	data bit selection used in pulse distance coding 0x1: SONY formate 0x0: NEC formate	0x0
31:21	Reserved			

21.12.5.4 IR_FRAME

Table 21.326: IR_FRAME, Offset Address: 0x00c

Bits	Name	Access	Description	Reset
7:0	reg_length	R/W	ir receiver data length	0x20
8	reg_slength	R/W	ir receiver repeate code length 0x1: TC9012 0x0: others	0x0
15:9	Reserved			
18:16	reg_lead_p_ratio	R/W	ratio of leading symbol period and strobe period, used in periodic sample mode	0x4
31:19	Reserved			

21.12.5.5 int_en

Table 21.327: int_en, Offset Address: 0x010

Bits	Name	Access	Description	Reset
0	reg_rx_done_int_en	R/W	rx_done interrupt enable	0x1
1	reg_frame_err_int_en	R/W	frame_err interrupt enable	0x1
2	reg_frame_ovf_int_en	R/W	frame_ovf interrupt enable	0x1
3	reg_release_int_en	R/W	release interrupt enable	0x1
4	reg_repeat_int_en	R/W	repeat interrupt enable	0x1
31:5	Reserved			

21.12.5.6 int_clr

Table 21.328: int_clr, Offset Address: 0x014

Bits	Name	Access	Description	Reset
0	reg_rx_done_int_clr	W1T	rx_done interrupt clear	
1	reg_frame_err_int_clr	W1T	frame_err interrupt clear	
2	reg_frame_ovf_int_clr	W1T	frame_ovf interrupt clear	
3	reg_release_int_clr	W1T	release interrupt clear	
4	reg_repeat_int_clr	W1T	repeat interrupt clear	
31:5	Reserved			

21.12.5.7 int_clr

Table 21.329: int_msk, Offset Address: 0x018

Bits	Name	Access	Description	Reset
0	reg_rx_done_int_msk	R/W	rx_done interrupt mask	0x0
1	reg_frame_err_int_msk	R/W	frame_err interrupt mask	0x0
2	reg_frame_ovf_int_msk	R/W	frame_ovf interrupt mask	0x0
3	reg_release_int_msk	R/W	release interrupt mask	0x0
4	reg_repeat_int_msk	R/W	repeat interrupt mask	0x0
31:5	Reserved			

21.12.5.8 int

Table 21.330: int, Offset Address: 0x01c

Bits	Name	Access	Description	Reset
0	reg_rx_done_int	RO	rx_done interrpt	
1	reg_frame_err_int	RO	frame_err interrpt	
2	reg_frame_ovf_int	RO	frame_ovf interrpt	
3	reg_release_int	RO	release interrpt	
4	reg_repeat_int	RO	repeat interrpt	
31:5	Reserved			

21.12.5.9 int_raw

Table 21.331: int_raw, Offset Address: 0x020

Bits	Name	Access	Description	Reset
0	reg_rx_done_int_raw	RO	rx_done interrupt raw value	
1	reg_frame_err_int_raw	RO	frame_err interrupt raw value	
2	reg_frame_ovf_int_raw	RO	frame_ovf interrupt raw value	
3	reg_release_int_raw	RO	release interrupt raw value	
4	reg_repeat_int_raw	RO	repeat interrupt raw value	
31:5	Reserved			

21.12.5.10 IR_SYMBOL_CFG0

Table 21.332: IR_SYMBOL_CFG0, Offset Address: 0x030

Bits	Name	Access	Description	Reset
11:0	reg_ir_rx_lead_p	R/W	lead symbol postive interval	0x383
15:12	Reserved			
23:16	reg_ir_rx_lead_p_tol	R/W	lead symbol postive interval tolerance reg_ir_rx_lead_p + reg_ir_rx_lead_p_tol <= 12'FFF reg_ir_rx_lead_p - reg_ir_rx_lead_p_tol >= 12'000	0x48
31:24	Reserved			

21.12.5.11 IR_SYMBOL_CFG1

Table 21.333: IR_SYMBOL_CFG0, Offset Address: 0x034

Bits	Name	Access	Description	Reset
11:0	reg_ir_rx_lead_n	R/W	lead symbol negtive interval	0x1c1
15:12	Reserved			
23:16	reg_ir_rx_lead_n_tol	R/W	lead symbol negtive interval tolerance reg_ir_rx_lead_n + reg_ir_rx_lead_n_tol <= 12'FFF reg_ir_rx_lead_n - reg_ir_rx_lead_n_tol >= 12'000	0x24
31:24	Reserved			

21.12.5.12 IR_SYMBOL_CFG2

Table 21.334: IR_SYMBOL_CFG2, Offset Address: 0x038

Bits	Name	Access	Description	Reset
7:0	reg_ir_rx_stop	R/W	stop symbol postive interval	0x37
11:8	reg_ir_rx_stop_tol	R/W	stop symbol postive interval tolerance reg_ir_rx_stop + reg_ir_rx_stop_tol <= 8'FF reg_ir_rx_stop - reg_ir_rx_stop_tol >= 8'00	0x3
15:12	Reserved			
23:16	reg_ir_rx_bit_p	R/W	data symbol postive interval	0x37
31:24	reg_ir_rx_bit_p_tol	R/W	data symbol postive interval tolerance reg_ir_rx_bit_p + reg_ir_rx_bit_p_tol <= 8'FF reg_ir_rx_bit_p - reg_ir_rx_bit_p_tol >= 8'00	0x3

21.12.5.13 IR_SYMBOL_CFG3

Table 21.335: IR_SYMBOL_CFG3, Offset Address: 0x03c

Bits	Name	Access	Description	Reset
11:0	reg_ir_rx_bit_one	R/W	data one totol interval	0xe0
15:12	Reserved			
23:16	reg_ir_rx_bit_one_tol	R/W	data one total interval tolerance reg_ir_rx_bit_one + reg_ir_rx_bit_one_tol <= 12'FFF reg_ir_rx_bit_one - reg_ir_rx_bit_one_tol >= 12'000	0x18
31:24	Reserved			

21.12.5.14 IR_SYMBOL_CFG4

Table 21.336: IR_SYMBOL_CFG4, Offset Address: 0x040

Bits	Name	Access	Description	Reset
11:0	reg_ir_rx_bit_zero	R/W	data zero totol interval	0x6f
15:12	Reserved			
23:16	reg_ir_rx_bit_zero_tol	R/W	data zero total interval tolerance reg_ir_rx_bit_zero + reg_ir_rx_bit_zero_tol <= 12'FFF reg_ir_rx_bit_zero - reg_ir_rx_bit_zero_tol >= 12'000	0x09
31:24	Reserved			

21.12.5.15 IR_SYMBOL_CFG5

Table 21.337: IR_SYMBOL_CFG5, Offset Address: 0x044

Bits	Name	Access	Description	Reset
15:0	reg_ir_rx_release_time	R/W	time for wait repeat code, >108ms.	0x27de
31:16	Reserved			

21.12.5.16 IR_SYMBOL_CFG6

Table 21.338: IR_SYMBOL_CFG6, Offset Address: 0x048

Bits	Name	Access	Description	Reset
11:0	reg_ir_rx_slead_p	R/W	repeate code lead symbol postive interval	0x383
15:12	Reserved			
23:16	reg_ir_rx_slead_p_tol	R/W	repeate code lead symbol postive interval tolerance reg_ir_rx_slead_p + reg_ir_rx_slead_p_tol <= 12'FFF reg_ir_rx_slead_p - reg_ir_rx_slead_p_tol >= 12'000	0x48
31:24	Reserved			

21.12.5.17 IR_SYMBOL_CFG7

Table 21.339: IR_SYMBOL_CFG7, Offset Address: 0x04c

Bits	Name	Access	Description	Reset
11:0	reg_ir_rx_slead_n	R/W	repeate code lead symbol negtive interval	0x1c1
15:12	Reserved			
23:16	reg_ir_rx_slead_n_tol	R/W	repeate code lead symbol negtive interval tolerance reg_ir_rx_slead_n + reg_ir_rx_slead_n_tol <= 12'FFF reg_ir_rx_slead_n - reg_ir_rx_slead_n_tol >= 12'000	0x24
31:24	Reserved			

21.12.5.18 IR_CLOCK_CTRL

Table 21.340: IR_CLOCK_CTRL, Offset Address: 0x050

Bits	Name	Access	Description	Reset
0	reg_pclock_auto_ctrl	R/W	pclk auto-gating control	0x0
1	reg_lpclock_switch_en	R/W	ip clock could be switch to 32KHz	0x0
7:2	Reserved			
8	reg_skip_lead_p	R/W	skip lead pulse check	0x0
31:9	Reserved			

21.12.5.19 IR_DATA0

Table 21.341: IR_DATA0, Offset Address: 0x080

Bits	Name	Access	Description	Reset
31:0	reg_ir_rx_data0	RO	recevier data[31:0]	

21.12.5.20 IR_DATA1

Table 21.342: IR_DATA1, Offset Address: 0x084

Bits	Name	Access	Description	Reset
31:0	reg_ir_rx_data1	RO	recevier data[63:32]	

21.12.5.21 IR_DATA2

Table 21.343: IR_DATA2, Offset Address: 0x088

Bits	Name	Access	Description	Reset
31:0	reg_ir_rx_data2	RO	recevier data[95:64]	

21.12.5.22 IR_DATA3

Table 21.344: IR_DATA3, Offset Address: 0x08c

Bits	Name	Access	Description	Reset
31:0	reg_ir_rx_data3	RO	recevier data[127:96]	

21.12.5.23 IR_DATA4

Table 21.345: IR_DATA4, Offset Address: 0x090

Bits	Name	Access	Description	Reset
31:0	reg_ir_rx_data4	RO	recevier data[159:128]	

21.12.5.24 IR_NEC_DATA0

Table 21.346: IR_NEC_DATA0, Offset Address: 0x0a8

Bits	Name	Access	Description	Reset
31:0	reg_ir_rx_nec_32bit	RO	receiver data, nec format	

21.12.5.25 IR_SONY_DATA0

Table 21.347: IR_SONY_DATA0, Offset Address: 0x0ac

Bits	Name	Access	Description	Reset
11:0	reg_ir_rx_sony_12bit	RO	receiver data, sony D7C5 format	
15:12	Reserved			
30:16	reg_ir_rx_sony_15bit	RO	receiver data, sony D7C8 format	
31	Reserved			

21.12.5.26 IR_SONY_DATA1

Table 21.348: IR_SONY_DATA1, Offset Address: 0x0b0

Bits	Name	Access	Description	Reset
19:0	reg_ir_rx_sony_20bit	RO	receiver data, sony D7C13 format	
31:20	Reserved			

21.12.5.27 IR_PHILIPS_DATA0

Table 21.349: IR_PHILIPS_DATA0, Offset Address: 0x0b4

Bits	Name	Access	Description	Reset
11:0	reg_ir_rx_philips_rc5	RO	receiver data, RC5 format	
31:12	Reserved			

21.12.5.28 IR_PHILIPS_DATA1

Table 21.350: IR_PHILIPS_DATA1, Offset Address: 0x0b8

Bits	Name	Access	Description	Reset
19:0	reg_ir_rx_philips_rc6	RO	receiver data, RC6 format	
31:20	Reserved			

21.12.5.29 IR_PRD_REC0

Table 21.351: IR_PRD_REC0, Offset Address: 0x0e0

Bits	Name	Access	Description	Reset
11:0	reg_start_p0_min	RO	start phase0 minimul width	
15:12	Reserved			
27:16	reg_start_p0_max	RO	start phase0 maximul width	
31:28	Reserved			

21.12.5.30 IR_PRD_REC1

Table 21.352: IR_PRD_REC1, Offset Address: 0x0e4

Bits	Name	Access	Description	Reset
11:0	reg_start_p1_min	RO	start phase1 minimul width	
15:12	Reserved			
27:16	reg_start_p1_max	RO	start phase1 maximul width	
31:28	Reserved			

21.12.5.31 IR_PRD_REC2

Table 21.353: IIR_PRD_REC2, Offset Address: 0x0e8

Bits	Name	Access	Description	Reset
11:0	reg_bit_p0_min	RO	bit phase0 minimul width	
15:12	Reserved			
27:16	reg_bit_p0_max	RO	bit phase0 maximul width	
31:28	Reserved			

21.12.5.32 IR_PRD_REC3

Table 21.354: IR_PRD_REC3, Offset Address: 0x0ec

Bits	Name	Access	Description	Reset
11:0	reg_bit_p1_min	RO	bit phase1 minimul width	
15:12	Reserved			
27:16	reg_bit_p1_max	RO	bit phase1 maximul width	
31:28	Reserved			

21.12.5.33 IR_PRD_REC4

Table 21.355: IR_PRD_REC4, Offset Address: 0x0f0

Bits	Name	Access	Description	Reset
11:0	reg_end_min	RO	end phase minimuml width	
15:12	Reserved			
27:16	reg_end_max	RO	end phase maximuml width	
31:28	Reserved			

21.12.5.34 IR_PRD_REC5

Table 21.356: IR_PRD_REC5, Offset Address: 0x0f4

Bits	Name	Access	Description	Reset
15:0	reg_frame_min	RO	frame minimul width	
31:16	reg_frame_max	RO	frame maximul width	

CHAPTER

TWENTYTWO

SECURITY SUBSYSTEM

The chip provides an independent security subsystem module responsible for providing specific security functions.

The safety subsystem module includes the following safety function modules

- Crypto DMA unit (CryptoDMA)
- True Random Number Generator
- Secure Debug Protection

The cryptographic operation unit (CryptoDMA) provides hardware acceleration of symmetric key encryption, decryption and hashing (Hash). The security eFuse unit is responsible for providing system security settings and security keys for use by the security subsystem. The true random number generation unit provides qualified random numbers. Random numbers are used by security systems.

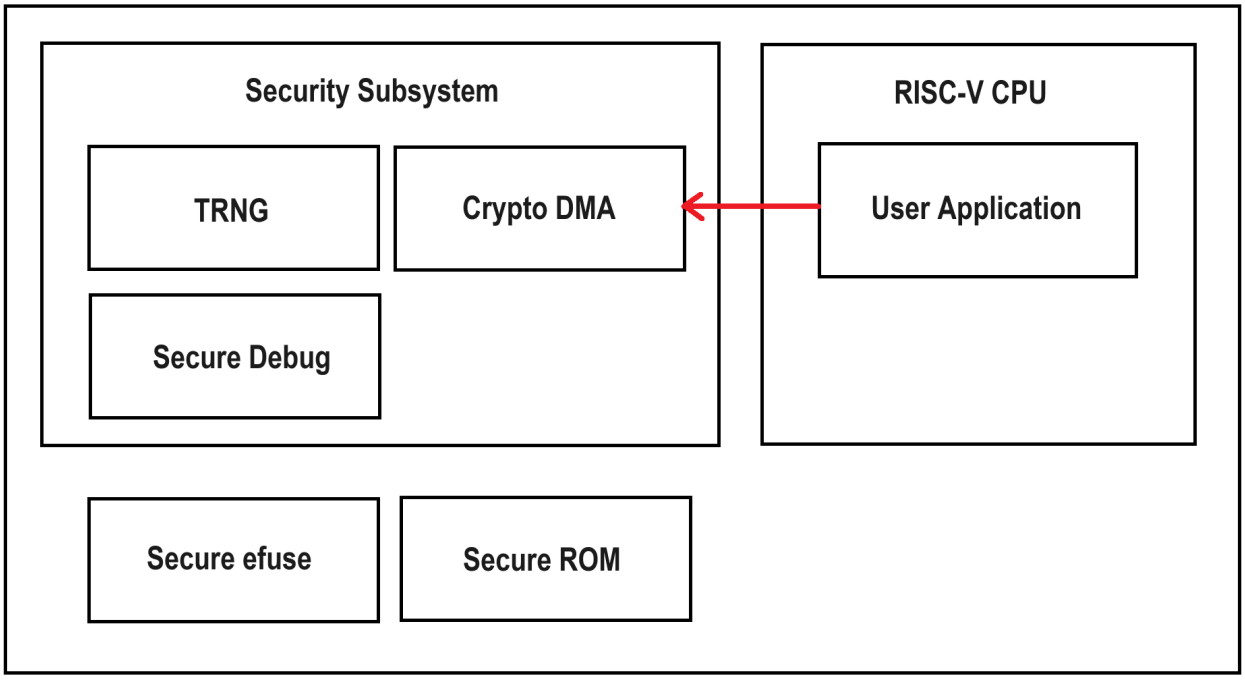


Diagram 22.1: Security subsystem module

22.1 CryptoDMA

22.1.1 Overview

CryptoDMA is a hardware accelerator that implements symmetric key algorithms, hash algorithms and BASE64 conversion. It supports symmetric algorithms: AES 128/192/256, DES/TDES, SM4 and hash algorithms: SHA-1/SHA256 and other cryptographic operations. Direct memory access of key encryption and decryption or hash operation functions of data blocks is achieved through linked-list command strings.

- The symmetric algorithm is suitable for accelerated hardware encryption and decryption of data, and supports a variety of group encryption and block concatenation processing methods, including ECB, CBC, and CTR.
- The implementation of the AES (Advanced Encryption Standard) algorithm complies with the FIPS 197 standard. The implementation of the DES (Data Encryption Standard)/TDES algorithm complies with ISO/IEC 18033-3.
- The hash algorithm is suitable for data integrity checking and digital signature operation acceleration. SHA1 and SHA256, compliant with FIPS180-2 standard.
- BASE64 operations are suitable for processing text data and storing 2-bit data, such as MIME email or URL data.

22.1.2 Features

The CryptoDMA module has the following functional features:

- Supports symmetric encryption and decryption algorithm AES and block encryption mode ECB/CBC/CTR. The key length supports 128 bits and 256 bits, and the key can be configured by the secure operating system or linked list instructions.
- Supports symmetric encryption and decryption algorithm SM4 and block encryption mode ECB/CBC/CTR.
- Supports symmetric encryption and decryption algorithms DES/TDES and block encryption modes ECB/CBC/CTR.
- Supports hash algorithms SHA1, SHA256.
- Supports CPU configuration input PIO data and DMA method to read table command input data.
- Supports circular linked list structure and supports splicing of multiple linked list data.
- Provides interrupt status query, interrupt masking and interrupt clearing functions.

22.1.3 DMA Function Description

CryptoDMA provides memory direct access DMA function. The application only needs to provide a linked list instruction to the target data block to start the CryptoDMA DMA function. Until the completed interrupt notification is received, the block encryption, decryption or hash operation is completed and the The operation result is output to the target address.

22.1.4 Symmetric key algorithm block encryption mode function description

Symmetric key algorithms AES/DES/SM4 all support ECB/CBC/CTR block encryption mode.

22.1.4.1 ECB Mode

In ECB (Electronic CodeBook) mode, the encryption and decryption algorithms are directly applied to each grouped data by the operation of each group. This feature allows plaintext encryption and ciphertext decryption to be performed independently for any group of block data.

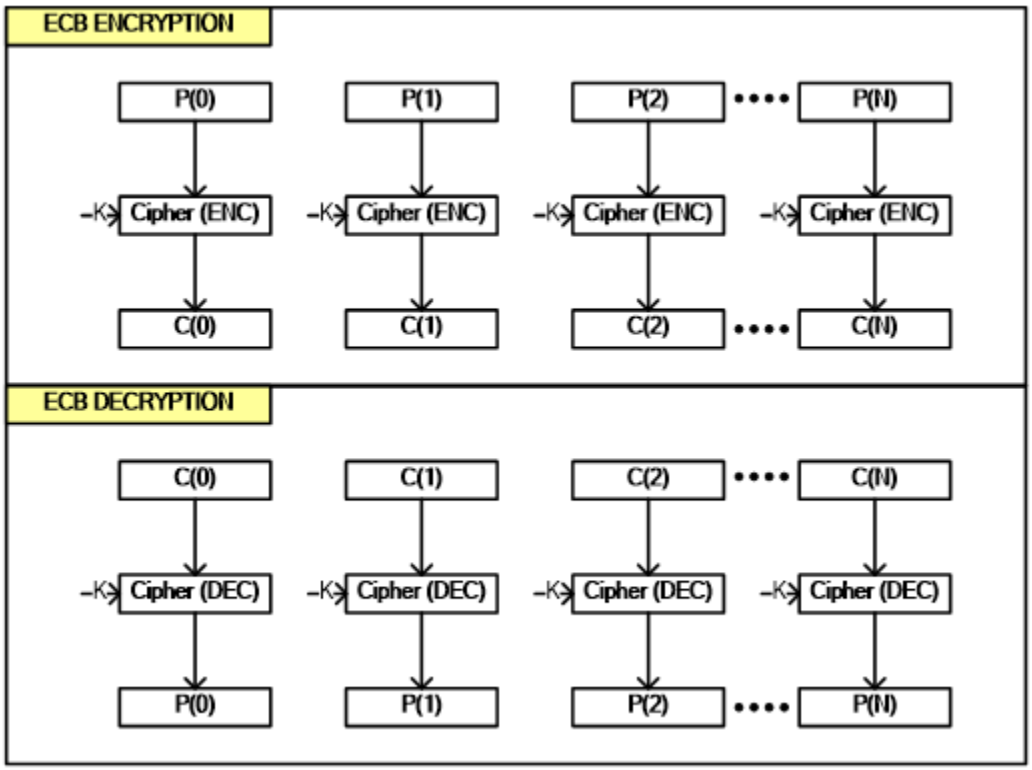


Diagram 22.2: ECB mode

22.1.4.2 CBC Mode

CBC (Cipher Block Chaining), the input plaintext group is first XORed with the input vector IV (Intialization Vector) or the previous group ciphertext result, and then the encryption operation is performed. The encryption operation in CBC mode must start from the first block. Block data grouping begins, and subsequent encryption operations require the ciphertext obtained from the previous group for encryption. During decryption, the plaintext can be obtained by decrypting the current ciphertext and XORing the previous set of ciphertexts.

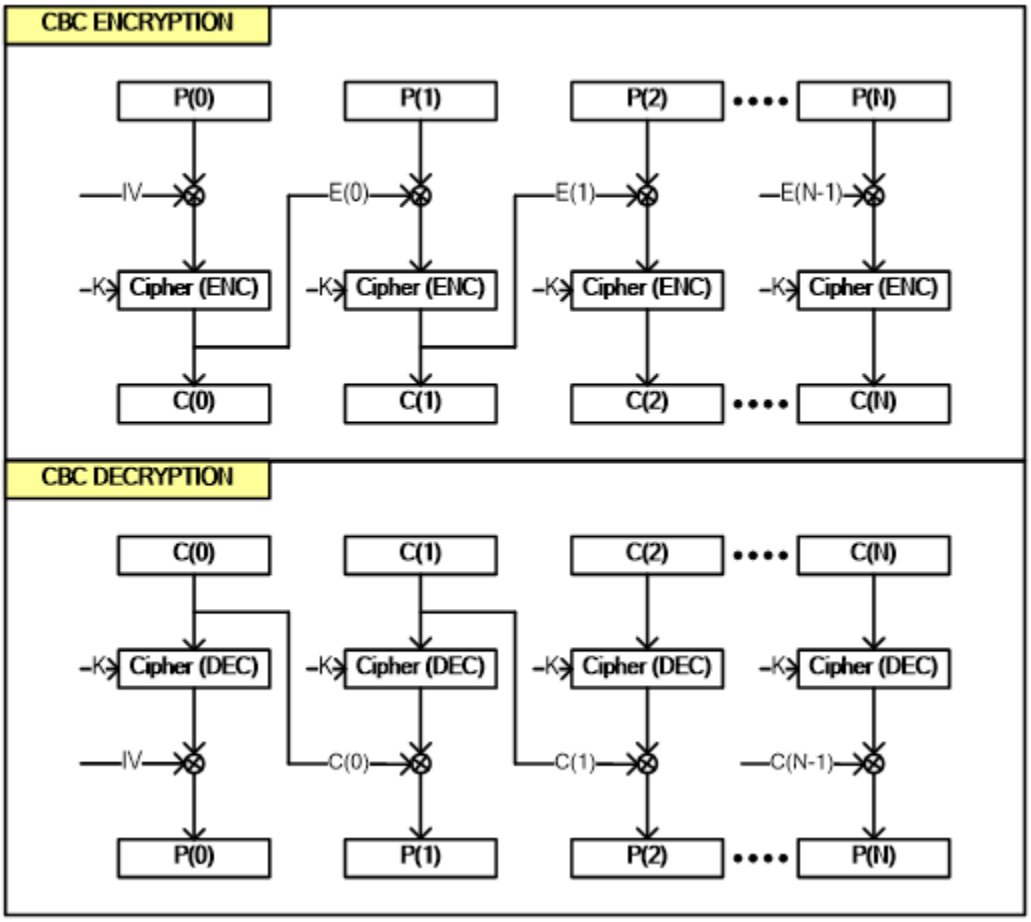


Diagram 22.3: CBC mode

22.1.4.3 CTR Mode

CTR (Counter) uses encryption or decryption to encrypt or decrypt a set of different arrays to ensure the independence and security of encrypted data processing. It generally uses an encrypted accumulation array and then performs an XOR operation with the plain text.

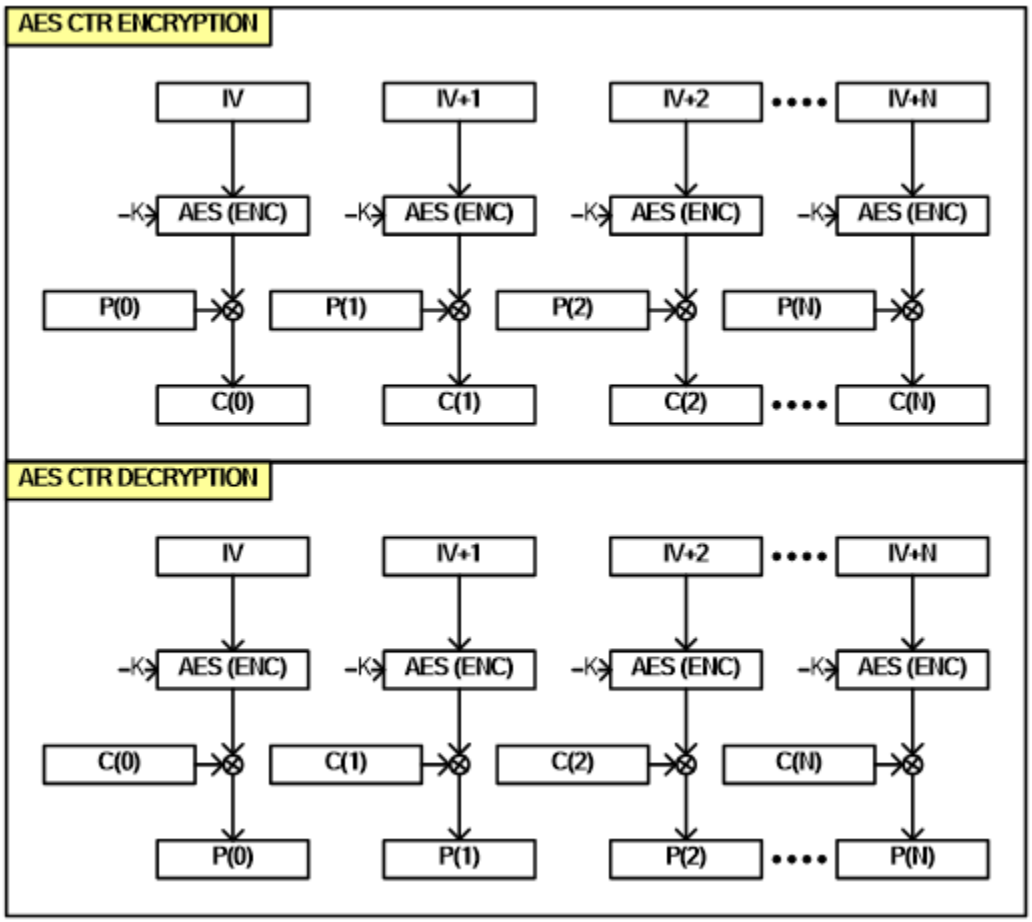


Diagram 22.4: CTR mode

22.1.5 CryptoDMA Register Overview

Table 22.1: CryptoDMA Registers Overview

Name	Address Offset	Description
dma_ctrl	0x000	DMA controll register
int_mask	0x004	interrupt mask
des_base_0	0x008	descriptor base low address
des_base_1	0x00c	descriptor base high address
spacc_int_raw	0x010	interrupt
secure_key_valid	0x014	key valid
des_addr_0	0x018	current descirptor low address
des_addr_1	0x01c	current descirptor high address
PIO_cmd_data_0	0x080	PIO command0

continues on next page

Table 22.1 – continued from previous page

Name	Address Offset	Description
PIO_cmd_data_1	0x084	PIO command1
PIO_cmd_data_2	0x088	PIO command2
PIO_cmd_data_3	0x08c	PIO command3
PIO_cmd_data_4	0x090	PIO command4
PIO_cmd_data_5	0x094	PIO command5
PIO_cmd_data_6	0x098	PIO command6
PIO_cmd_data_7	0x09c	PIO command7
PIO_cmd_data_8	0x0a0	PIO command8
PIO_cmd_data_9	0x0a4	PIO command9
PIO_cmd_data_10	0x0a8	PIO command10
PIO_cmd_data_11	0x0ac	PIO command11
PIO_cmd_data_12	0x0b0	PIO command12
PIO_cmd_data_13	0x0b4	PIO command13
PIO_cmd_data_14	0x0b8	PIO command14
PIO_cmd_data_15	0x0bc	PIO command15
PIO_cmd_data_16	0x0c0	PIO command16
PIO_cmd_data_17	0x0c4	PIO command17
PIO_cmd_data_18	0x0c8	PIO command18
PIO_cmd_data_19	0x0cc	PIO command19
PIO_cmd_data_20	0x0d0	PIO command20
PIO_cmd_data_21	0x0d4	PIO command21
key_data_0	0x100	cipher key data 0
key_data_1	0x104	cipher key data 1
key_data_2	0x108	cipher key data 2
key_data_3	0x10c	cipher key data 3
key_data_4	0x110	cipher key data 4
key_data_5	0x114	cipher key data 5
key_data_6	0x118	cipher key data 6
key_data_7	0x11c	cipher key data 7
key_data_8	0x120	cipher key data 8
key_data_9	0x124	cipher key data 9
key_data_10	0x128	cipher key data 10
key_data_11	0x12c	cipher key data 11
key_data_12	0x130	cipher key data 12
key_data_13	0x134	cipher key data 13
key_data_14	0x138	cipher key data 14
key_data_15	0x13c	cipher key data 15
key_data_16	0x140	cipher key data 16
key_data_17	0x144	cipher key data 17
key_data_18	0x148	cipher key data 18
key_data_19	0x14c	cipher key data 19
key_data_20	0x150	cipher key data 20
key_data_21	0x154	cipher key data 21
key_data_22	0x158	cipher key data 22
key_data_23	0x15c	cipher key data 23
ini_data_0	0x180	initial vector data 0
ini_data_1	0x184	initial vector data 1
ini_data_2	0x188	initial vector data 2
ini_data_3	0x18c	initial vector data 3

continues on next page

Table 22.1 – continued from previous page

Name	Address Offset	Description
ini_data_4	0x190	initial vector data 4
ini_data_5	0x194	initial vector data 5
ini_data_6	0x198	initial vector data 6
ini_data_7	0x19c	initial vector data 7
ini_data_8	0x1a0	initial vector data 8
ini_data_9	0x1a4	initial vector data 9
ini_data_10	0x1a8	initial vector data 10
ini_data_11	0x1ac	initial vector data 11
sha_data_0	0x1c0	SHA paramenter0
sha_data_1	0x1c4	SHA paramenter1
sha_data_2	0x1c8	SHA paramenter2
sha_data_3	0x1cc	SHA paramenter3
sha_data_4	0x1d0	SHA paramenter4
sha_data_5	0x1d4	SHA paramenter5
sha_data_6	0x1d8	SHA paramenter6
sha_data_7	0x1dc	SHA paramenter7

22.1.6 CryptoDMA Register Description

(Base address: 0x02060000)

22.1.6.1 dma_ctrl

Table 22.2: dma_ctrl, Offset Address: 0x000

Bits	Name	Access	Description	Reset
0	dma_en	R/W	DMA channel enable control 0: Channel disabled; 1: Channel enabled.	0x0
1	decriptor_mode	R/W	Channel instruction mode 0: PIO mode 1: Descriptor table mode	0x0
15:2	Reserved			
23:16	max_read_burst	R/W	Maximum read burst value	0x0
31:24	max_write_burst	R/W	Maximum write burst value	0x0

22.1.6.2 int_mask

Table 22.3: int_mask, Offset Address: 0x004

Bits	Name	Access	Description	Reset
0	int_enc_mask	R/W	Encryption interrupt mask	0x0
1	int_hash_mask	R/W	Hash interrupt mask	0x0
31:2	Reserved		Reserved	

22.1.6.3 des_base_0

Table 22.4: des_base_0, Offset Address: 0x008

Bits	Name	Access	Description	Reset
31:0	des_base_0	R/W	Descriptor table address (Low)	0x0

22.1.6.4 des_base_1

Table 22.5: des_base_1, Offset Address: 0x00c

Bits	Name	Access	Description	Reset
31:0	des_base_1	R/W	Descriptor table address (High)	0x0

22.1.6.5 spacc_int_raw

Table 22.6: spacc_int_raw, Offset Address: 0x010

Bits	Name	Access	Description	Reset
31:0	spacc_int_raw	R/W	DMA interrupt	0x0

22.1.6.6 secure_key_valid

Table 22.7: secure_key_valid, Offset Address: 0x014

Bits	Name	Access	Description	Reset
31:0	secure_key_valid	R/W	Key validity **One-Way:** Enabled	0x0

22.1.6.7 des_addr_0

Table 22.8: des_addr_0, Offset Address: 0x018

Bits	Name	Access	Description	Reset
31:0	des_addr_0	R/W	Description key table address offset (low)	0x0

22.1.6.8 des_addr_1

Table 22.9: des_addr_1, Offset Address: 0x01c

Bits	Name	Access	Description	Reset
31:0	des_addr_1	R/W	Description key table address offset (high)	0x0

22.1.6.9 PIO_cmd_data_0

Table 22.10: PIO_cmd_data_0, Offset Address: 0x080

Bits	Name	Access	Description	Reset
0	PIO_cmd_data_0	R/W	PIO mode descriptor table, 0: Mode descriptor table invalid 1: Mode descriptor table valid	0x0
1	PIO_cmd_data_0_1	R/W	Last descriptor of descriptor table	0x0
2	PIO_cmd_data_0_2	R/W	PIO mode interrupt enable 0: Interrupt disabled; 1: Interrupt enabled.	0x0
3	PIO_cmd_data_0_3	R/W	Continuation descriptor table address select 0: Use current address continuation 1: Use continuation register value	0x0
7:4	PIO_cmd_data_0_4	R/W	Reserved	0x0
8	PIO_cmd_data_0_8	R/W	DMA bypass	0x0
9	PIO_cmd_data_0_9	R/W	AES encryption/decryption	0x0
10	PIO_cmd_data_0_10	R/W	DES encryption/decryption	0x0
11	PIO_cmd_data_0_11	R/W	SM4 encryption/decryption	0x0
12	PIO_cmd_data_0_12	R/W	SHA operations	0x0
13	PIO_cmd_data_0_13	R/W	BASE64 encoding/decoding	0x0
15:14	Reserved			
19:16	PIO_cmd_data_0_16	R/W	Key selection for operations For aes/des/sm4 operations: 1: Descriptor key 2: Key 2 4: Key 1 8: Key 0 For SHA operations: 1: Descriptor parameter 2: Reserved 4: Reserved 8: Use SHA parameter register Other values reserved	0x0
23:20	PIO_cmd_data_0_20	R/W	Initial IV selection for operation For aes/des/sm4 operations: 1: Descriptor IV 2: IV2 4: IV1 8: IV0	0x0
26:24	PIO_cmd_data_0_24	R/W	Continuation IV selection for operation For aes/des/sm4 operations: 1: Descriptor IV 2: IV2 4: IV1 8: IV0	0x0
27	PIO_cmd_data_0_27	R/W	Key selection enable	0x0
31:28	PIO_cmd_data_0_28	R/W	Reserved	0x0

22.1.6.10 PIO_cmd_data_1

Table 22.11: PIO_cmd_data_1, Offset Address: 0x084

Bits	Name	Access	Description	Reset
0	PIO_cmd_data_1_0	R/W	Encryption/Decryption selection 1-Encryption/0-Decryption Hash parameter selection 1-Need parameter/0-No need	0x0
1	PIO_cmd_data_1_1	R/W	CBC mode 0-ECB/1-CBC	0x0
2	PIO_cmd_data_1_2	R/W	CTR mode 1-CTR	0x0
5:3	PIO_cmd_data_1_3	R/W	Key mode 100-128bit/010-192bit/001-256bit for aes 0-DES/1-TDES for DES SHA mode 0-SHA1/1-SHA256	0x0
31:6	PIO_cmd_data_1_6	R/W	Reserved	0x0

22.1.6.11 PIO_cmd_data_2

Table 22.12: PIO_cmd_data_2, Offset Address: 0x088

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_2	R/W	Reserved	0x0

22.1.6.12 PIO_cmd_data_3

Table 22.13: PIO_cmd_data_3, Offset Address: 0x08c

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_3	R/W	Continuation descriptor address (low)	0x0

22.1.6.13 PIO_cmd_data_4

Table 22.14: PIO_cmd_data_4, Offset Address: 0x090

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_4	R/W	Continuation descriptor address (high)	0x0

22.1.6.14 PIO_cmd_data_5

Table 22.15: PIO_cmd_data_5, Offset Address: 0x094

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_5	R/W	DMA source address (low)	0x0

22.1.6.15 PIO_cmd_data_6

Table 22.16: PIO_cmd_data_6, Offset Address: 0x098

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_6	R/W	DMA source address (high)	0x0

22.1.6.16 PIO_cmd_data_7

Table 22.17: PIO_cmd_data_7, Offset Address: 0x09c

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_7	R/W	DMA target address (low)	0x0

22.1.6.17 PIO_cmd_data_8

Table 22.18: PIO_cmd_data_8, Offset Address: 0x0a0

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_8	R/W	DMA target address (high)	0x0

22.1.6.18 PIO_cmd_data_9

Table 22.19: PIO_cmd_data_9, Offset Address: 0x0a4

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_9	R/W	SHA information size	0x0

22.1.6.19 PIO_cmd_data_10

Table 22.20: PIO_cmd_data_10, Offset Address: 0x0a8

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_10	R/W	reservation	0x0

22.1.6.20 PIO_cmd_data_11

Table 22.21: PIO_cmd_data_11, Offset Address: 0x0ac

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_11	R/W	BASE64 target information size	0x0

22.1.6.21 PIO_cmd_data_12

Table 22.22: PIO_cmd_data_12, Offset Address: 0x0b0

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_12	R/W	reservation	0x0

22.1.6.22 PIO_cmd_data_13

Table 22.23: PIO_cmd_data_13, Offset Address: 0x0b4

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_13	R/W	reservation	0x0

22.1.6.23 PIO_cmd_data_14

Table 22.24: PIO_cmd_data_14, Offset Address: 0x0b8

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_14	R/W	reservation	0x0

22.1.6.24 PIO_cmd_data_15

Table 22.25: PIO_cmd_data_15, Offset Address: 0x0bc

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_15	R/W	reservation	0x0

22.1.6.25 PIO_cmd_data_16

Table 22.26: PIO_cmd_data_16, Offset Address: 0x0c0

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_16	R/W	reservation	0x0

22.1.6.26 PIO_cmd_data_17

Table 22.27: PIO_cmd_data_17, Offset Address: 0x0c4

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_17	R/W	reservation	0x0

22.1.6.27 PIO_cmd_data_18

Table 22.28: PIO_cmd_data_18, Offset Address: 0x0c8

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_18	R/W	reservation	0x0

22.1.6.28 PIO_cmd_data_19

Table 22.29: PIO_cmd_data_19, Offset Address: 0x0cc

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_19	R/W	reservation	0x0

22.1.6.29 PIO_cmd_data_20

Table 22.30: PIO_cmd_data_20, Offset Address: 0x0d0

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_20	R/W	reservation	0x0

22.1.6.30 PIO_cmd_data_21

Table 22.31: PIO_cmd_data_21, Offset Address: 0x0d4

Bits	Name	Access	Description	Reset
31:0	PIO_cmd_data_21	R/W	reservation	0x0

22.1.6.31 key_data_0

Table 22.32: key_data_0, Offset Address: 0x100

Bits	Name	Access	Description	Reset
31:0	key_data_0	RO	key	

22.1.6.32 key_data_1

Table 22.33: key_data_1, Offset Address: 0x104

Bits	Name	Access	Description	Reset
31:0	key_data_1	RO	key	

22.1.6.33 key_data_2

Table 22.34: key_data_2, Offset Address: 0x108

Bits	Name	Access	Description	Reset
31:0	key_data_2	RO	key	

22.1.6.34 key_data_3

Table 22.35: key_data_3, Offset Address: 0x10c

Bits	Name	Access	Description	Reset
31:0	key_data_3	RO	key	

22.1.6.35 key_data_4

Table 22.36: key_data_4, Offset Address: 0x110

Bits	Name	Access	Description	Reset
31:0	key_data_4	RO	key	

22.1.6.36 key_data_5

Table 22.37: key_data_5, Offset Address: 0x114

Bits	Name	Access	Description	Reset
31:0	key_data_5	RO	key	

22.1.6.37 key_data_6

Table 22.38: key_data_6, Offset Address: 0x118

Bits	Name	Access	Description	Reset
31:0	key_data_6	RO	key	

22.1.6.38 key_data_7

Table 22.39: key_data_7, Offset Address: 0x11c

Bits	Name	Access	Description	Reset
31:0	key_data_7	RO	key	

22.1.6.39 key_data_8

Table 22.40: key_data_8, Offset Address: 0x120

Bits	Name	Access	Description	Reset
31:0	key_data_8	RO	key	

22.1.6.40 key_data_9

Table 22.41: key_data_9, Offset Address: 0x124

Bits	Name	Access	Description	Reset
31:0	key_data_9	RO	key	

22.1.6.41 key_data_10

Table 22.42: key_data_10, Offset Address: 0x128

Bits	Name	Access	Description	Reset
31:0	key_data_10	RO	key	

22.1.6.42 key_data_11

Table 22.43: key_data_11, Offset Address: 0x12c

Bits	Name	Access	Description	Reset
31:0	key_data_11	RO	key	

22.1.6.43 key_data_12

Table 22.44: key_data_12, Offset Address: 0x130

Bits	Name	Access	Description	Reset
31:0	key_data_12	RO	key	

22.1.6.44 key_data_13

Table 22.45: key_data_12, Offset Address: 0x134

Bits	Name	Access	Description	Reset
31:0	key_data_13	RO	key	

22.1.6.45 key_data_14

Table 22.46: key_data_14, Offset Address: 0x138

Bits	Name	Access	Description	Reset
31:0	key_data_14	RO	key	

22.1.6.46 key_data_15

Table 22.47: key_data_15, Offset Address: 0x13c

Bits	Name	Access	Description	Reset
31:0	key_data_15	RO	key	

22.1.6.47 key_data_16

Table 22.48: key_data_16, Offset Address: 0x140

Bits	Name	Access	Description	Reset
31:0	key_data_16	RO	key	

22.1.6.48 key_data_17

Table 22.49: key_data_17, Offset Address: 0x144

Bits	Name	Access	Description	Reset
31:0	key_data_17	RO	key	

22.1.6.49 key_data_18

Table 22.50: key_data_18, Offset Address: 0x148

Bits	Name	Access	Description	Reset
31:0	key_data_18	RO	key	

22.1.6.50 key_data_19

Table 22.51: key_data_19, Offset Address: 0x14c

Bits	Name	Access	Description	Reset
31:0	key_data_19	RO	key	

22.1.6.51 key_data_20

Table 22.52: key_data_20, Offset Address: 0x150

Bits	Name	Access	Description	Reset
31:0	key_data_20	RO	key	

22.1.6.52 key_data_21

Table 22.53: key_data_21, Offset Address: 0x154

Bits	Name	Access	Description	Reset
31:0	key_data_21	RO	key	

22.1.6.53 key_data_22

Table 22.54: key_data_22, Offset Address: 0x158

Bits	Name	Access	Description	Reset
31:0	key_data_22	RO	key	

22.1.6.54 key_data_23

Table 22.55: key_data_23, Offset Address: 0x15c

Bits	Name	Access	Description	Reset
31:0	key_data_23	RO	key	

22.1.6.55 ini_data_0

Table 22.56: ini_data_0, Offset Address: 0x180

Bits	Name	Access	Description	Reset
31:0	ini_data_0	RO	initial parameters	

22.1.6.56 ini_data_1

Table 22.57: ini_data_1, Offset Address: 0x184

Bits	Name	Access	Description	Reset
31:0	ini_data_1	RO	initial parameters	

22.1.6.57 ini_data_2

Table 22.58: ini_data_2, Offset Address: 0x188

Bits	Name	Access	Description	Reset
31:0	ini_data_2	RO	initial parameters	

22.1.6.58 ini_data_3

Table 22.59: ini_data_3, Offset Address: 0x18c

Bits	Name	Access	Description	Reset
31:0	ini_data_3	RO	initial parameters	

22.1.6.59 ini_data_4

Table 22.60: ini_data_4, Offset Address: 0x190

Bits	Name	Access	Description	Reset
31:0	ini_data_4	RO	initial parameters	

22.1.6.60 ini_data_5

Table 22.61: ini_data_5, Offset Address: 0x194

Bits	Name	Access	Description	Reset
31:0	ini_data_5	RO	initial parameters	

22.1.6.61 ini_data_6

Table 22.62: ini_data_5, Offset Address: 0x198

Bits	Name	Access	Description	Reset
31:0	ini_data_6	RO	initial parameters	

22.1.6.62 ini_data_7

Table 22.63: ini_data_7, Offset Address: 0x19c

Bits	Name	Access	Description	Reset
31:0	ini_data_7	RO	initial parameters	

22.1.6.63 ini_data_8

Table 22.64: ini_data_8, Offset Address: 0x1a0

Bits	Name	Access	Description	Reset
1:0	ini_data_8	RO	initial parameters	
31:2	Reserved			

22.1.6.64 ini_data_9

Table 22.65: ini_data_9, Offset Address: 0x1a4

Bits	Name	Access	Description	Reset
1:0	ini_data_9	RO	initial parameters	
31:2	Reserved			

22.1.6.65 ini_data_10

Table 22.66: ini_data_10, Offset Address: 0x1a8

Bits	Name	Access	Description	Reset
31:0	ini_data_10	RO	initial parameters	

22.1.6.66 ini_data_11

Table 22.67: ini_data_11, Offset Address: 0x1ac

Bits	Name	Access	Description	Reset
31:0	ini_data_11	RO	initial parameters	

22.1.6.67 sha_data_0

Table 22.68: sha_data_0, Offset Address: 0x1c0

Bits	Name	Access	Description	Reset
31:0	sha_data_0	RO	SHA parameters	

22.1.6.68 sha_data_1

Table 22.69: sha_data_1, Offset Address: 0x1c4

Bits	Name	Access	Description	Reset
31:0	sha_data_1	RO	SHA parameters	

22.1.6.69 sha_data_2

Table 22.70: sha_data_2, Offset Address: 0x1c8

Bits	Name	Access	Description	Reset
31:0	sha_data_2	RO	SHA parameters	

22.1.6.70 sha_data_3

Table 22.71: sha_data_3, Offset Address: 0x1cc

Bits	Name	Access	Description	Reset
31:0	sha_data_3	RO	SHA parameters	

22.1.6.71 sha_data_4

Table 22.72: sha_data_4, Offset Address: 0x1d0

Bits	Name	Access	Description	Reset
31:0	sha_data_4	RO	SHA parameters	

22.1.6.72 sha_data_5

Table 22.73: sha_data_5, Offset Address: 0x1d4

Bits	Name	Access	Description	Reset
31:0	sha_data_5	RO	SHA parameters	

22.1.6.73 sha_data_6

Table 22.74: sha_data_6, Offset Address: 0x1d8

Bits	Name	Access	Description	Reset
31:0	sha_data_6	RO	SHA parameters	

22.1.6.74 sha_data_7

Table 22.75: sha_data_7, Offset Address: 0x1dc

Bits	Name	Access	Description	Reset
31:0	sha_data_7	RO	SHA parameters	

22.2 Security Debugging Firewall

In order to provide reading or control of related internal functions of the chip during debugging or testing, the chip provides several debugging interfaces, such as JTAG, I2C and other different external interfaces. Without appropriate protection mechanisms, these interfaces can easily be exploited to directly or indirectly attack the chip security mechanism or read internal information that needs to be kept confidential. To protect and control these interfaces, the chip uses a security debug firewall.

22.2.1 Overview

The chip supports three main debugging interfaces,

1. RISCV JTAG: The RISCV processor has a built-in debugging interface that allows users to access ARM internal registers through the JTAG interface.
2. I2C: The chip provides a debugging interface
3. Test interface: The chip provides a dedicated debugging interface for production testing.

For the JTAG/I2C interface, the secure debug firewall provides specific protection controls on access to it

For the Test interface interface, the security debugging interface provides an independent type of protection control (Test access) for its access.

For these debugging categories, the security debugging firewall provides three connection control states

- Open: Allows external connections via connectors without imposing additional controls.
- Protected: Does not allow external connections through the connector until the corresponding password is input through the I2C interface to unlock the protection.
- Closed: External connections are not allowed through this interface and cannot be opened again by other methods.

22.2.2 Status query and password input interface (I2C)

The security debugging firewall provides an independently operating I2C interface for the outside of the chip to query the current status of the debugging interface through I2C and enter the corresponding password to reopen the interface in the protected state. Externally, you need to specify the correct I2C ID to connect to the firewall interface.

The I2C interface register address is as follows

Table 22.76: Status query I2C interface register address

Byte address	Bitwidth	Singal Name	Description
0	32	i2c_REE_password [31:0]	128-bit debug interface password
4	32	i2c_REE_password [63:32]	128-bit debug interface password
8	32	i2c_REE_password [95:64]	128-bit debug interface password
C	32	i2c_REE_password [127:96]	128-bit debug interface password
10	32	Reserved	Reserved
14	32	Reserved	Reserved
18	32	Reserved	Reserved
1C	32	Reserved	Reserved
20	32	Reserved	Reserved
24	32	Reserved	Reserved
28	32	Reserved	Reserved
2C	32	Reserved	Reserved
30	32	i2c_TST_password [31:0]	128-bit password entry field for test ports
34	32	i2c_TST_password [63:32]	128-bit password entry field for test ports
38	32	i2c_TST_password [95:64]	128-bit password entry field for test ports
3C	32	i2c_TST_password [127:96]	128-bit password entry field for test ports
40	1	REE_PW_update	Update password comparison results
44	1	Reserved	Reserved
48	1	Reserved	Reserved
4C	1	TST_PW_update	Update test interface password comparison results
80	32	Chip_UID0 (LSB of ID)	Chip identification number (Device ID)
84	32	Chip_UID1 (MSB of ID)	Chip identification number (Device ID)
88	32	MKTSEG	Market Segment number
8C	32	DBG_MODE	Chip debugging interface protection setting status [1:0] : Interface protection mode, 0:open;1:protected:2/3:closed [3:2] : Reserved [5:4] : Reserved [7:6] : Test interface protection mode, 0:open;1:protected:2/3:closed [8] : Compare using HASH
90	32	reserved	Reserved
94	4	DBG_PROT_STATUS	Current debug protection status [0]: Interface protection status, 0:open 1:closed [1]: Reserved [2]: Reserved [3]: Test interface protection status, 0:open 1:closed

22.2.3 Status inquiry and password entry process

22.2.3.1 Status inquiry process

- (Step 1) Control external I2C to send out start signal.
- (Step 2) Send firewall I2C ID by I2C (default 0x56).
- (Step 3) Read address 0x04001A94 from I2C to obtain the current debug interface protection status.

22.2.3.2 Password input process

- (Step 1) Control external I2C to send out start signal.
- (Step 2) The I2C ID of the debug interface firewall is sent by I2C (default 0x56).
- (Step 3) Read the address 0x04001A80 / 0x04001A84 from I2C to obtain the device ID, and read 0x04001A88 to obtain the market distinction number.
- (Step 4) Prepare the unlocking password corresponding to each category through the device serial number and market identification number.
- (Step 5) Read address 0x94 from I2C to obtain the current debug interface protection status and check whether it is locked.
- (Step 6) Taking the non-secure debug interface as an example, I2C writes the non-secure password to the address 0x04001A00 / 0x04001A04 / 0x04001A08 / 0x04001A0C.
- (Step 7) I2C writes any value to address 0x04001A10 to update the password comparison value.
- (Step 8) Read address 0x94 from I2C to obtain the current debug interface protection status and confirm whether it is unlocked.

22.3 Efuse Controller

The chip integrates 4Kbit eFuse space, and uses Efuse Ctrl to program and read Efuse.

The main functions of Efuse Ctrl include:

- Provides a dual eFuse bit (Double bit) protection mechanism, which consists of two physical eFuse bits forming a single bit logical effective value, which is equivalent to providing a 2Kbit register space to improve the robustness of eFuse programming or data maintenance.
- After power-on reset (Power-On-Reset), the contents of efuse are automatically loaded into the register to provide the required configuration settings for the chip system and reduce the number of times required to read Efuse to increase the service life.
- Provides efuse programming, reading, verification reading and power-on and power-off instructions and content security protection mechanism.

The Efuse data register is divided into two areas, one is a non-safe area and the other is a safe area. The data in the non-safe area is allowed to be accessed by all modules, and the safe area only allows access to the safe modules. The non-secure area stores system configuration and public information, and the secure area stores security configuration, keys and passwords.

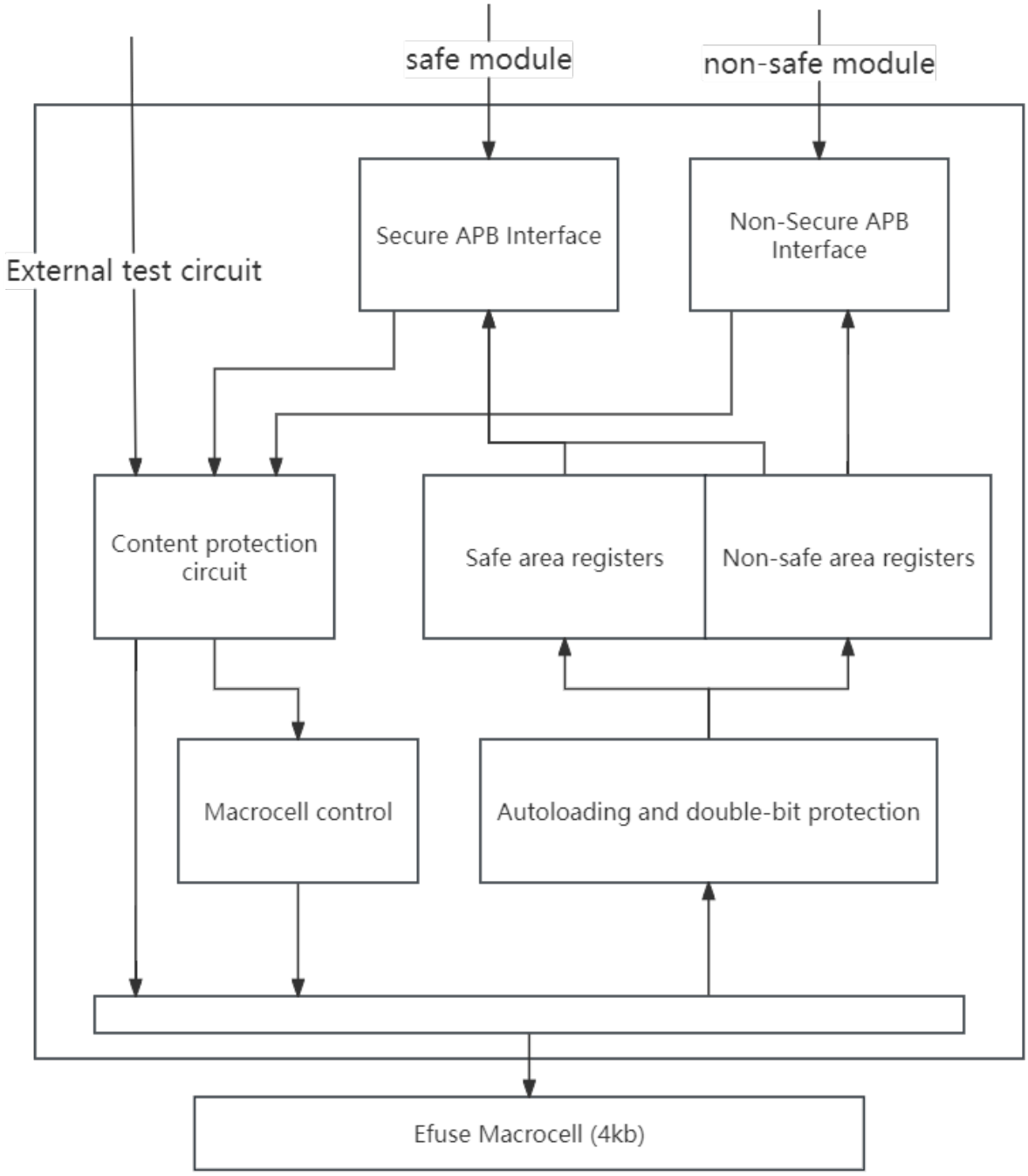


Diagram 22.5: eFuse CTRL modular architecture