



K230 Technical Reference Manual

Document version: V0.3.1
Release date: 20241118

Disclaimer

The products, services or features you purchase shall be subject to the commercial contracts and terms of Canaan Creative Co., LTD (the "Company") and its affiliates, and all or part of the products, services or features described in this document may not be within the scope of your purchase or use. Unless otherwise agreed in the contract, the Company does not provide any express or implied representations or warranties as to the correctness, reliability, completeness, merchantability, fitness for a particular purpose and non-aggression of any statements, information, or content in this document. Unless otherwise agreed, this document is intended as a guide for use only. Due to product version upgrades or other reasons, the content of this document may be updated or modified from time to time without any notice.

Trademark Notice



"Canaan" and other Canaan trademarks are trademarks of Canaan Creative Co., LTD and its affiliates. All other trademarks or registered trademarks that may be mentioned in this document are owned by their respective owners.

All rights reserved©2024Canaan Creative Co., LTD

This document is only suitable for the development and design of K230 platform, without the written permission of the company, any unit or individual shall not disseminate part or all of the content of this document in any form.

Canaan Creative Co., LTD

Website : canaan-creative.com

Business Inquiry : salesAI@canaan-creative.com

Preface

Purpose

This document describes the features, logical structures, functions, operating modes, and related registers of each module of Canaan K230.

Intended Audience

This document is intended for:

- Design and maintenance personnel for electronics
- Sales personnel for electronic parts and components

...

Revision history

The revision record accumulates a description of each document update. The latest version of the documentation contains updates for all previous versions

Version number	Modified by	Revision date	Revision Instructions
V0.1.0	XXX	2023-03-09	The first interim version release
V0.2.0	XXX	2023-06-28	The second interim version release
V0.3.0	Jimmy Huang	2024-11-10	Markdown format correction
V0.3.1	Jimmy Huang	2024-11-18	Add IOMUX section

illustrate : The initial version is V0.1.0, and the second/revision number can be adjusted during the editing process before release.

1 Product Description

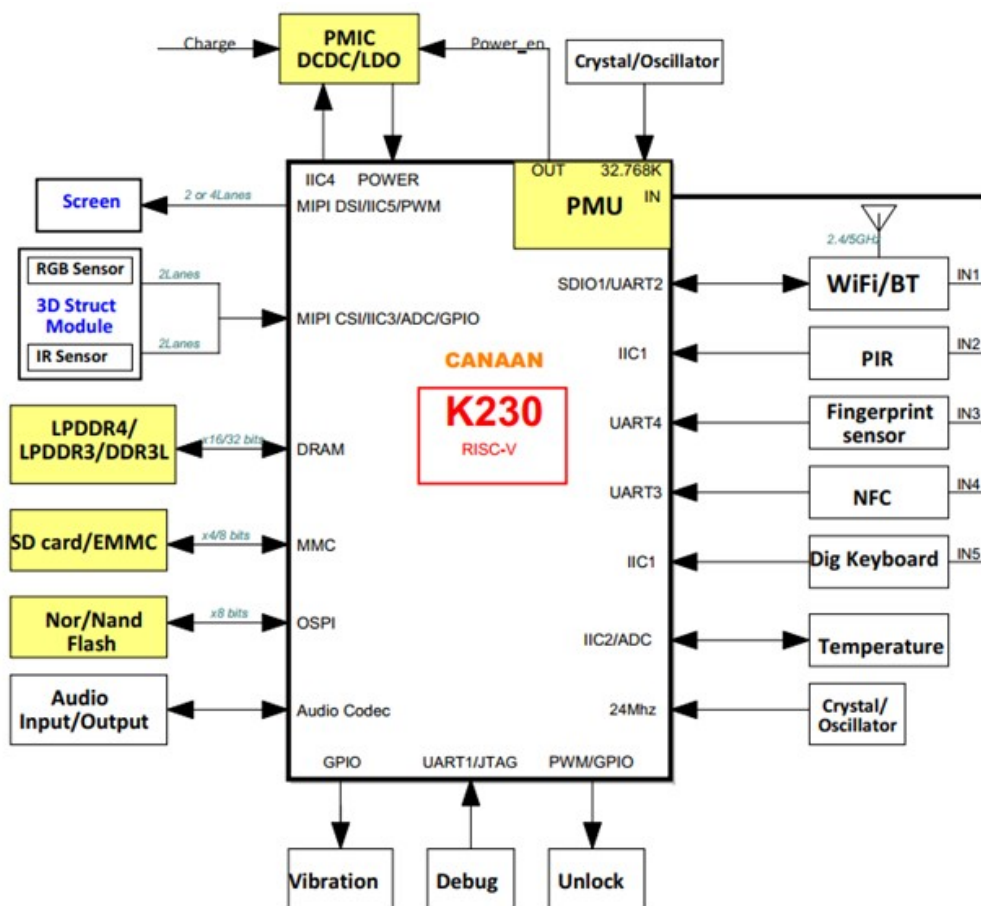
1.1 Description

K230 is an on-device AI chip for intelligent door locks, dictionary pens, consumer IPC, payment modules, entrance guards, and industrial code readers. It introduces a new generation ISP that can access up to three channels simultaneously, supports multi-channel codecs including H265, depth map extraction unit (DPU), and high-performance neural network processor (KPU), giving K230 paytion-level recognition accuracy. Integrated with PMU, audio codec and security modules, K230 can greatly help users reduce the cost of EBOM.

1.2 Application Scenarios

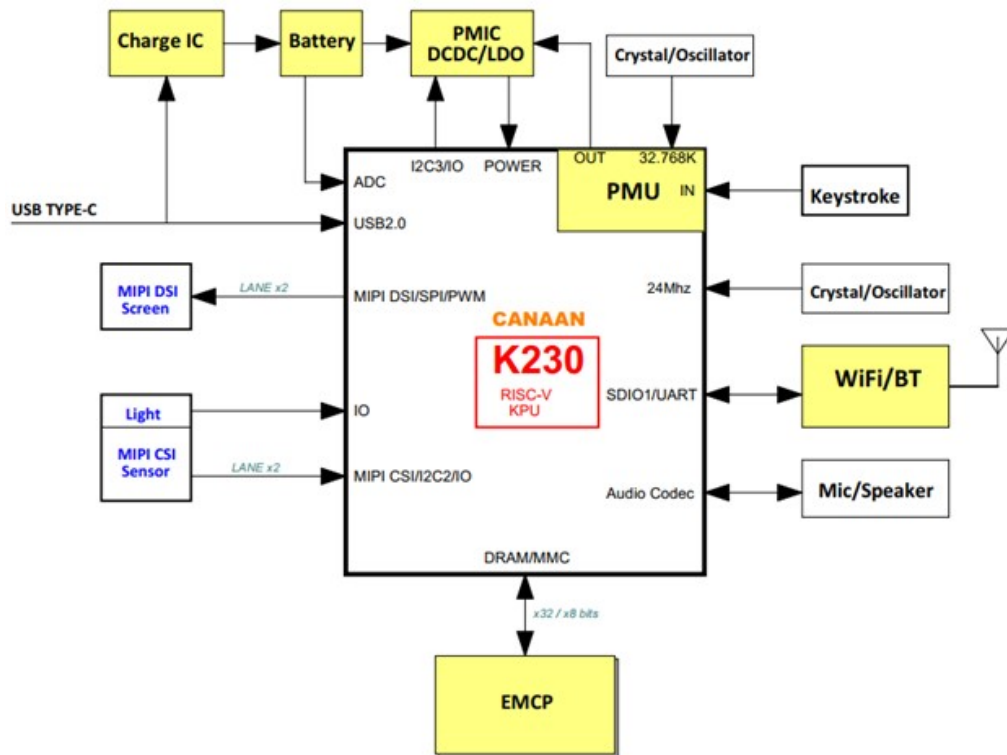
1.2.1 K230 intelligent door locks solution

Figure 1-1 shows intelligent door locks solution of K230



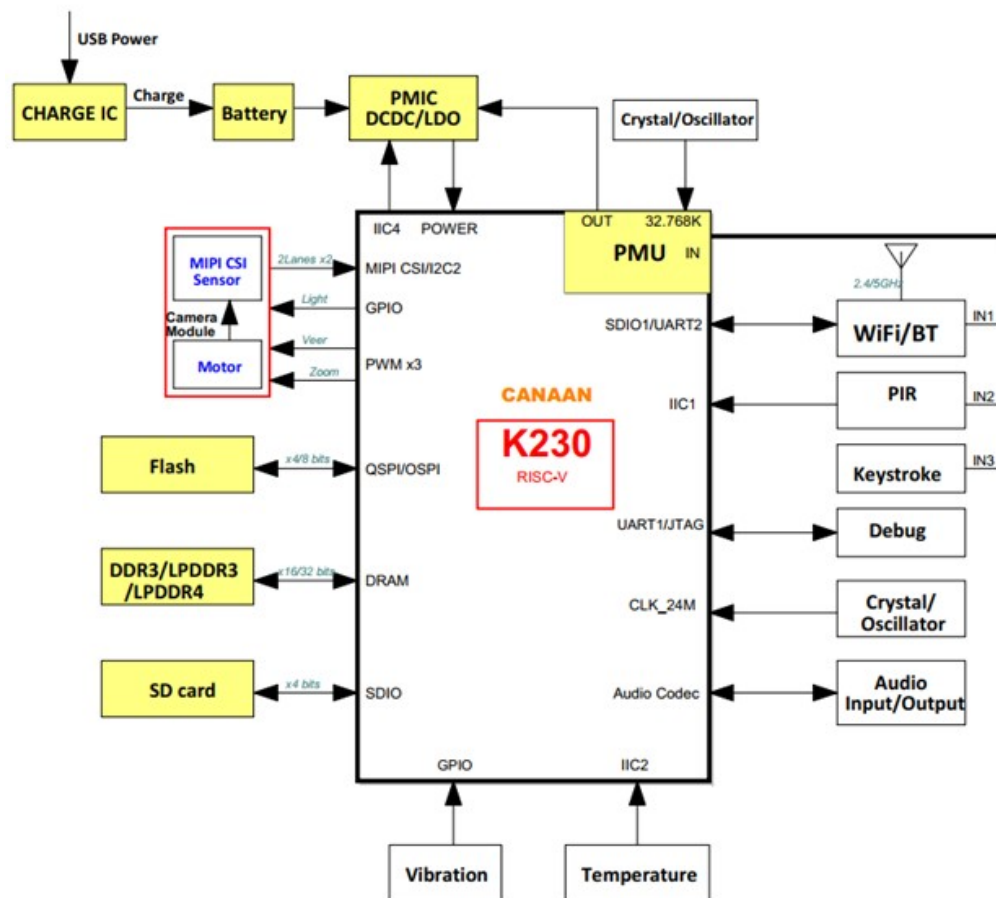
1.2.2 K230 dictionary pens solution

Figure 1-2 shows dictionary pens solution of K230



1.2.3 K230 IPC solution

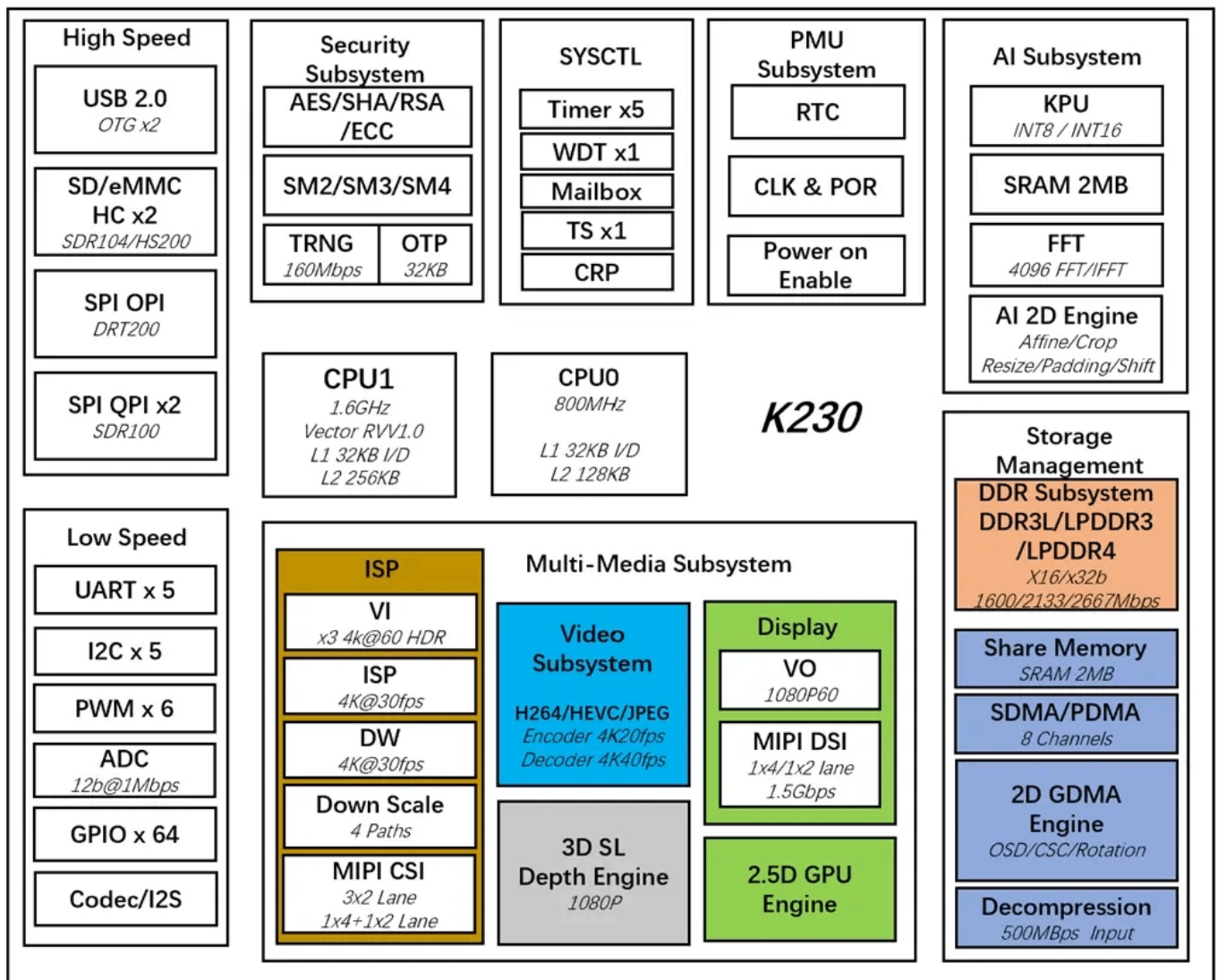
Figure 1-3 shows IPC solution of K230



1.3 Architecture

1.3.1 Overview

Figure 1-4 shows the logic block diagram of K230



1.3.2 Process Core

- Big core(CPU1) XuanTie C908@ 1600 Mhz, 32KB I-cache, 32KB D-cache, 256KB L2 Cache with 128bits vector extension unit
- Small core(CPU0) XuanTie C908@ 800Mhz, 32KB I-cache, 32KB D-cache, 128KB L2 Cache

1.3.3 KPU

- Supports multiple precision data types: Compatible with int8 and int16 to meet diverse model requirements.
- Enables weight sparsification and compression: Optimizes weight storage and computation for improved efficiency and resource utilization.
- Performance:
 - mobilenetv2@560fps
 - resnet50@70fps
- Framework compatibility: Fully supports popular deep learning frameworks and model formats, including TVM, TensorFlow, PyTorch, and ONNX.

1.3.4 Video CODEC

- Up to 4096 x 4096 resolution for integrated H.264/H.265 encoding and decoding.
- Real-time multi-stream
 - 2960x1666@25fps H.264/H.265 encoding + 720x576@30fps H.264/H.265 encoding + 2960x1666@5fps JPEG encoding
 - 2960x1666@25fps H.264/H.265 decoder + 2960x1666@5fps JPEG decoder
- JPEG encoding and decoding support up to 8K (8192 x 8192) resolution
- Support Fixed QP operation or rate-controlled operation
- Up to 50MBps output bit rate
- Support for rotation of the decoded frame in 90 degree steps before output.
- Support for downscaling the decoded frame horizontally and vertically by a factor of 2 or 4

1.3.5 ISP

1.3.5.1 ISP

- Support input format 8/10/12/14 bit Bayer RAW
- Support Output Format YCbCr, RAW, RGB
- 3A functions (AE, AF, and AWB)
- 2DNR and 3DNR
- Mechanical shutter control
- WDR in single wide dynamic frame
- multi-exposure HDR, includes multi exposure stitching and local tone mapping
 - 2960x1666@60fps, DOL2
 - 1920x1080@90fps, DOL3
- Multi-context management to support up to three sensors within a single ISP
- Performance:
 - 3840x2160@30fps
 - 2*1280x1080@30fps + 1920x1080@30fps ;

1.3.5.2 DW

- Fisheye correction
- Wide Field of View (FOV) correction
- Keystone correction
- Scaling up from 1x to 4x, apply in dewarp images
- Fisheye 180° /360° view
- 4PTZ view
- Mirror, Flip
- Multi-resize outputs, arbitrary scale ratio

1.3.6 Video Interfaces

- VI
 - 1x4 channel and 1x2 channel, or 3x2 channel MIPI VI
 - RAW-8/10/12/14/16
 - Compatibility with mainstream HDR sensors such as Sony, OV, semiconshop, SmartSens
 - Support flash trigger
 - Programmable sensor clock output
 - Configurable hardware frame drop
 - Hardware time stamp
- VO
 - 1x4 channel, or 1x2 channel DSI MIPI VO

- 4 video layers, 1 background layer, 8 OSD layers
 - video layer0 and layer1 support rotation, x-mirror, and flip, resize
- Hardware time stamp
- Maximum resolution of 1080P30

1.3.7 3D Processing Engine

- Using 3D structured light depth calculation
- Support multiple operators, including Img_check / LCN / SAD / Post_proc / Align/ Disptodepth
- The maximum resolution of the landscape image is 19201080, and the maximum resolution of the portrait image is 10801440
- Typical performance is 1280*800@30fps, 1280*1080@15fps,1920*1080@9fps

1.3.8 Audio Interface

- Audio codec, supporting 24-bit input and output
- Stereo and mono modes PDM
- support two differential or single-end dual-channel input, and two single-end dual-channel output
- support 2 I2S MIC interface
- support up to 8 PDM MIC interface

1.3.9 Peripherals

- I2C
 - 5 I2C master interfaces
- UART
 - 5 UART interfaces
- SPI
 - SPI interface
- eMMC/SD/SDIO
 - Two eMMC/SD/SDIO interfaces with protocol SDIO3.0 and eMMC5.1, supporting the 3.3V/1.8V level
 - SDO support 1/4/8bits
 - SD1 support 1/4 bits
- SPI
 - OSPI master with 4/8bits, supporting SDR 166Mhz and DTR 200Mhz
 - Two QSPI master with 1/2/4bits, supporting SDR 104Mhz
- GPIO
 - 64 GPIOs
- USB
 - Two USB2.0 OTG
- PWM
 - 4 PWM interfaces

1.3.10 Security

- Secure boot
- Hardware-based encryption and decryption algorithms (including AES, SM4 RSA, SM2, ECC)
- Hardware-based HASH algorithms (SHA256/SHA512/HMAC_SHA/SM3)
- Hardware TRNG
- 4K bits storage OTP

1.3.11 PMU

- Ultra-low power PMU @ uAs

1.3.12 Memory Interface

- SDRAM interface
 - Support 16/32-bit LPDDR4 and 32-bit LPDDR3
 - Support dram max capacity 2GB
 - Up to 3200Mbps for LPDDR4
 - Up to 2133Mbps for LPDDR3

1.3.13 Startup

Booting from SPI NOR flash, SPI NAND Flash, or eMMC.

1.3.14 Physical Specifications

- Power Consumption -2MP@30fps+1280*800@60fps, typical power consumption(TBD)
- Operating voltage
 - 0.8V core voltage
 - 1.8/3.3V IO voltage
 - 1.35V/1.2V/1.1V LPDDR3/LPDDR4 SDRAM interface voltage
- package
 - BGA, 0.65mm pitch
 - 13 x 13mm
 - 11 x 11mm (SIP 128MB 16bit LPDDR4)

1.4 Boot and Upgrade Modes

1.4.1 Overview

The chip has two CPUs named CPU0 (Small Core) and CPU1 (Big Core). During the startup, the embedded boot ROM (BOOTROM) starts on CPU0 (small core) after chip reset is deasserted. The dereset of CPU1 (large core) is controlled by the CPU0 (small core).

Boot media Selection

The chip can be booted from the following storage media:

- SPI NorFLASH
- SPI NandFLASH
- SD Card
- eMMC Card

The boot medium selection is determined by BOOT signal.

Burnig(Upgrade) Mode

The chip can also burn (upgrade) boot media through a USB port, or a serial port. The update mode priority is first through USB port, then serial port when the chip hasn't booted from storage media.

1.4.2 Mapping Relation Between Boot Modes

The boot is determined by BOOT signals. These signals can be obtained by querying PADs level status.

Table 1-1 describes the relationship between boot modes signals.

Boot mode

BOOT1	BOOT0	Boot Mode
0	0	Boot from SPI NorFlash memory
0	1	Boot from SPI NandFlash memory
1	0	Boot from eMMC
1	1	Boot from SD card

1.4.3 Secure Boot

K230 supports both secure and non-secure boot modes. The value (obtained from the secure boot flag bit in the OTP module) determines whether the secure or non-secure boot mode is preferred.

In secure boot mode, the chip verifies signatures for the U-Boot. If the signature verification succeeds, the system starts. Otherwise, the system does not start.

In non-secure boot mode, the chip does not verify signatures for the U-Boot.

1.5 Address Space mapping

For details about the address space, see Table 1-2

Address space mapping

Start Address	End Address	Function	Size
0x0000_0000	0x8000_0000	DDRC	2GB
0x8000_0000	0x8020_0000	KPU L2 Cache	2MB
0x8020_0000	0x8040_0000	SRAM	2MB
0x8040_0000	0x8040_0800	KPU Configuration	2KB
0x8040_0800	0x8040_0C00	FFT	1KB
0x8040_0C00	0x8040_1000	AI 2D Engine	2KB
0x8080_0000	0x8080_4000	gsdma	16KB
0x8080_4000	0x8080_8000	DMA	16KB
0x8080_8000	0x8080_C000	decomp_gzip	16KB
0x8080_C000	0x8081_0000	non_ai_2d	16KB
0x9000_0000	0x9000_8000	ISP	32KB
0x9000_8000	0x9000_9000	DeWarp	4KB
0x9000_9000	0x9000_B000	RX CSI	8KB
0x9040_0000	0x9041_0000	H264/HEVC/JPEG Encoder/Decoder	64KB
0x9080_0000	0x9084_0000	2.5D	256KB
0x9084_0000	0x9085_0000	VO	64KB

Start Address	End Address	Function	Size
0x9085_0000	0x9085_1000	VO_cfg/DSI	4KB
0x90A0_0000	0x90A0_0800	3D Engine	2KB
0x9100_0000	0x9100_0C00	PMU	3KB
0x9100_0C00	0x9100_1000	RTC	1KB
0x9110_0000	0x9110_1000	CMU	4KB
0x9110_1000	0x9110_2000	RMU	4KB
0x9110_2000	0x9110_3000	BOOT	4KB
0x9110_3000	0x9110_4000	PWR	4KB
0x9110_4000	0x9110_5000	mailbox	4KB
0x9110_5000	0x9110_5800	iomux	2KB
0x9110_5800	0x9110_6000	timer	2KB
0x9110_6000	0x9110_6800	wdt0	2KB
0x9110_6800	0x9110_7000	wdt1	2KB
0x9110_7000	0x9110_7800	TS	2KB
0x9110_7800	0x9110_8000	HDI	2KB
0x9110_8000	0x9110_9000	STC	4KB
0x9120_0000	0x9121_0000	BootRom	64KB
0x9121_0000	0x9121_8000	Security	32KB
0x9140_0000	0x9140_1000	UART0	4KB
0x9140_1000	0x9140_2000	UART1	4KB
0x9140_2000	0x9140_3000	UART2	4KB
0x9140_3000	0x9140_4000	UART3	4KB
0x9140_4000	0x9140_5000	UART4	4KB
0x9140_5000	0x9140_6000	I2C0	4KB
0x9140_6000	0x9140_7000	I2C1	4KB
0x9140_7000	0x9140_8000	I2C2	4KB
0x9140_8000	0x9140_9000	I2C3	4KB
0x9140_9000	0x9140_A000	I2C4	4KB
0x9140_A000	0x9140_B000	PWM	4KB
0x9140_B000	0x9140_C000	GPIO0	4KB
0x9140_C000	0x9140_D000	GPIO1	4KB
0x9140_D000	0x9140_E000	ADC	4KB
0x9140_E000	0x9140_F000	CODEC	4KB
0x9140_F000	0x9141_0000	Audio (I2S)	4KB

Start Address	End Address	Function	Size
0x9150_0000	0x9158_0000	USB2.0 OTG x2	256KB
0x9158_0000	0x9158_2000	SD HC X2	8KB
0x9158_2000	0x9158_4000	SPI QOPI x2	9KB
0x9158_4000	0x9158_5000	SPI OPI	4kB
0x9158_5000	0x9158_5400	Hi_sys_config space	1KB
0x9800_0000	0x9A00_0000	DDRC configuration space	32MB
0xC000_0000	0xC800_0000	SPI OPI XIP FLASH	128MB
0xf_0000_0000	0xf_0020_3004	CPU0 (Small core)Plic	~2MB
0xf_0000_0000	0xf_0020_3004	CPU1 (Big core)Plic	~2MB

2 System

2.1 Reset

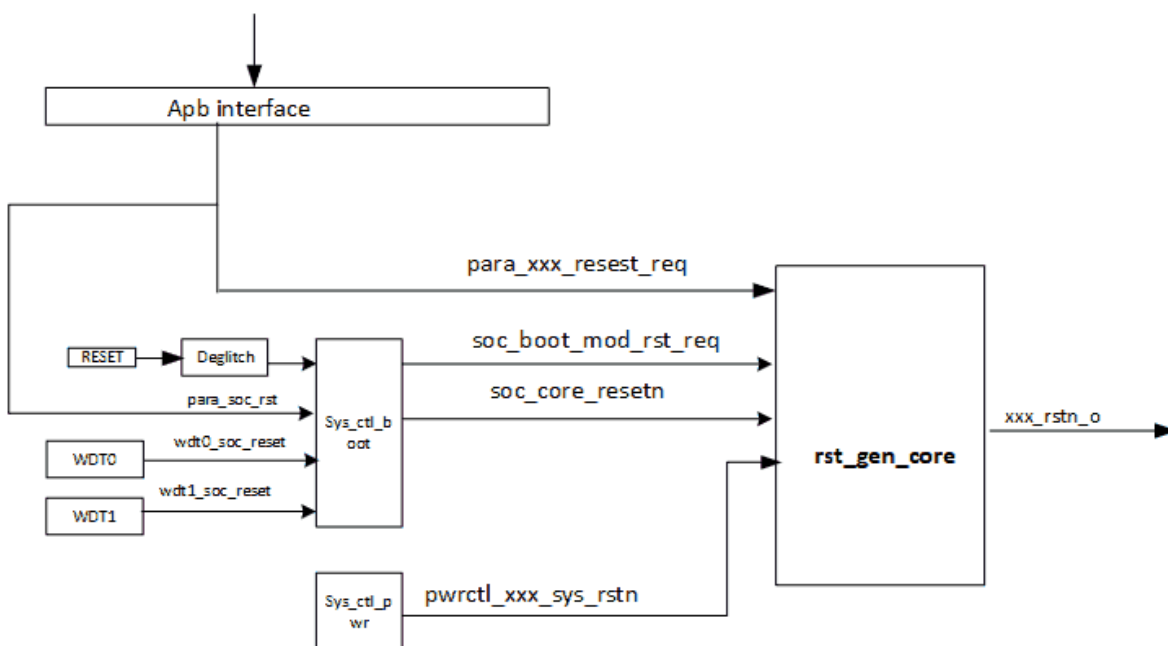
2.1.1 overview

Reset management module (RMU) is responsible for power on reset deglitch, manage wdt reset , chip soft reset and generate sub-module reset.

- Deglitch power on reset. Reset glitch length less than 8ms will be filtered. Genertate soc reset soc_core_resetrn.
- Manage two watchdog reset and generate soc reset soc_core_resetrn.
- Manage software soc reset para_soc_rst and generate soc reset soc_core_resetrn.
- Manage sub-module software reset request para_XXX_reset_req and generate sub-module reset xxx_rstn_o
- Manage sub-module reset from power ctrl and generate sub-module reset xxx_rstn_o.

2.1.2 block diagram

Figure 2-1-1 reset block diagram



There are four chip level reset source:

- RESET: reset from IO
- wdt0_soc_reset from watch-dog0
- wdt1_soc_reset from watch-dog1
- software configured reset para_soc_rst through apb bus.

rst_gen_core is responsible for sub-module reset generation. There are three sub-module reset source:

- software configured reset para_xxx_reset_req through apb bus.
- reset request soc_boot_mod_rst_req and reset level signal soc_core_resetsn from sys_ctl_boot
- pwrctl_xxx_sys_rstn from power control module sys_ctl_pwr.

2.1.3 Function description

2.1.3.1 deglitch

The instability of reset from offchip will affect correct function of soc chip. Glitch of reset must be removed. Deglitch module is responsible for reset deglitch. The 24M OSC clock and internal counter is used during this process. Glitch length less than 8ms will be removed.

2.1.3.2 Apb

CPU can configure reset module through apb interface.

2.1.3.3 rst_gen_core

rst_gen_core is responsible for sub-module reset generation.

2.1.3.4 reset for DDRC

During reset process clock must be stable. Reset sequence must be met:

- After soc power up apb reset will be deasserted at first automatically. DDRC core and AXI clock domains remain at reset status.
- Software need to check register MCTL_LPI_STAT.ddrc_init to get whether apb reset is removed. When MCTL_LPI_STAT.ddrc_init=1, apb reset is removed. Then software begin to configure DDRC.
- When configuration of DDRC is finished, configure MCTL_PWR_LPI_CTL.mctl_ddrc_init_done to 1.
- DDRC core and AXI clock domains reset deassert automatically.
- Software begin to configure DDR PHY. When configuration of DDR PHY is finished, set MCTL_LPI_STAT.mctl_dfi_init_done to 1.

2.1.3.5 reset for USB

AHB clock domain reset must deassert after deassertion of USB phy reset. Default value of USB_RST_TIM is reasonable and it's better dont modify the default value.

2.1.4 Register summary

global reset base_addr: refer to [global reset base address](#)

Table 2-2-1 global reset

Offset	Register	Description
0x60	SOC_GLB_RST	SOC global reset control

module reset base_addr: refer to [RMU base address](#)

Table 2-1-2 module reset

Offset	Register	Description
0x0	CPU0_RST_TIM	cpu0 reset time control
0x4	CPU0_RST_CTL	cpu0 reset control
0x8	CPU1_RST_TIM	cpu1 reset time control
0xc	CPU1_RST_CTL	cpu1 reset control
0x10	AI_RST_TIM	AI reset time control
0x14	AI_RST_CTL	AI reset control
0x18	VPU_RST_TIM	VPU reset time control
0x1C	VPU_RST_CTL	VPU reset control
0x20	SOC_CTL_RST_CTL	soc ctl reset control
0x24	LOSYS_RST_CTL	lo sys reset control
0x28	HISYS_HCLK_TIM	hi sys reset time control
0x2c	HISYS_RST_CTL	hi sys reset control
0x30	SDCTL_RST_TIM	sd reset time control
0x34	SDC_RST_CTL	sd reset control
0x38	USB_RST_TIM	usb reset time control
0x3c	USB_RST_CTL	usb reset control
0x40	SPI_RST_TIM	spi reset time control
0x44	SPI_RST_CTL	spi reset control
0x48	SEC_SYS_RST_TIM	sec reset time control
0x4c	SEC_RST_CTL	sec reset control
0x50	DMAC_RST_TIM	dmac reset time control
0x54	DMA_RST_CTL	dmac reset control
0x58	DECOMPRESS_RST_TIM	decompress reset time control
0x5c	DECOMPRESS_RST_CTL	decompress reset control
0x60	SRAM_RST_TIM	sram reset time control
0x64	SRAM_RST_CTL	sram reset control
0x68	NONAI2D_RST_TIM	nonai-2d reset time control
0x6c	NONAI2D_RST_CTL	nonai-2d reset control
0x70	MCTL_RST_TIM	ddrc reset time control
0x74	MCTL_RST_CTL	ddrc reset control
0x78	ISP_RST_TIM	isp reset time control

Offset	Register	Description
0x7c	ISP_DW_RST_TIM	isp de reset time control
0x80	ISP_RST_CTL	isp reset control
0x84	DPU_RST_TIM	DPU reset time control
0x88	DPU_RST_CTL	DPU reset control
0x8c	DISP_SYS_RST_TIM	display reset time control
0x90	DISP_RST_CTL	display reset control
0x94	V2P5D_SYS_RST_TIM	video 2.5D reset time control
0x98	V2P5D_RST_CTL	video 2.5D reset control
0xa0	AUDIO_RST_TIM	audio reset time control
0xa4	AUDIO_RST_CTL	audio reset control

2.1.5 register description

SOC_GLB_RST

Offset=0x60, Total Reset Value: 0x0000_0000, Write_enable=yes

Bits	Access	Name	Descriptions	Reset
4:3	RW	wdt1_reset_cfg	0x0,0x1: wdt1 reset soc, 0x2: wdt1 reset cpu0, 0x3: wdt1 reset cpu1	0x0
2:1	RW	wdt0_reset_cfg	0x0,0x1: wdt0 reset soc, 0x2: wdt0 reset cpu0, 0x3: wdt0 reset cpu1	0x0
0	WO	soc_global_reset	SoC core global reset. 0x0: no operation; 0x1: trigger a global reset to SoC core.	0x0

CPU0_RST_TIM

Offset=0x0, Total Reset Value: 0x00ff_f880, Write_enable=no

Bits	Access	Name	Descriptions	Reset
27:20	RW	cpu0_rst_ctl_tim2	The time period when reset is removed but clock stopped. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us;; 0xF: 150.25us; other values: forbidden.	0xf
19:12	RW	cpu0_rst_ctl_tim1	The time period when reset is applied but clock stopped. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us;; 0xF: 150.25us; other values: forbidden.	0xff
11:8	RW	cpu0_prst_ctl_tim2	The time period when apb reset is removed but clock stopped. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us;; 0xF: 150.25us; other values: forbidden.	0x8
7:4	RW	cpu0_prst_ctl_tim1	The time period when apb reset is applied but clock stopped. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us;; 0xF: 150.25us; other values: forbidden.	0x8

CPU0_RST_CTL

Offset=0x4, Total Reset Value: 0x0000_7000, Write_enable=yes

Bits	Access	Name	Descriptions	Reset
14	W1C	tdt_rst_done	tdt resetting done status bit. Writing '1' to clear this status bit. 0x1: module resetting done; 0x0: no event.	0x0
13	W1C	cpu0_prst_done	CPU0 system apb resetting done status bit. Writing '1' to clear this status bit. 0x1: module resetting done; 0x0: no event.	0x0
12	W1C	cpu0_rst_done	CPU0 system resetting done status bit. Writing '1' to clear this status bit. 0x1: module resetting done; 0x0: no event.	0x0
4	RWSC	cpu0_flush_req	1: flush cpu l2chche, hard auto clear this bit when receive cpu flush done	0x0
3	W1T	cpu1_preset_req	Cpu1 apb reset request bit. 0: noreset 1: reset	0x0
2	W1T	cpu0_preset_req	Cpu0 apb reset request bit. 0: noreset 1: reset	0x0
1	W1T	tdt_preset_req	tdi apb reset request bit 0: noreset 1: reset	0x0
0	W1T	cpu0_reset_req	CPU0 accelerating subsystem reset request bit. It is a write-only and self-clear bit. 0x0: no action; 0x1: request to reset cpu0 subsystem.	0x0

CPU1_RST_TIM

Offset=0x8, Total Reset Value: 0x0006_6660, 0xWrite_enable=no

Bits	Access	Name	Descriptions	Reset
19:16	RW	cpu1_rst_ctl_tim2	The time period when reset is removed but clock stopped. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us;; 0xF: 150.25us; other values: forbidden.	0x6
15:12	RW	cpu1_rst_ctl_tim1	The time period when reset is applied but clock stopped. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us;; 0xF: 150.25us; other values: forbidden.	0x6
11:8	RW	cpu1_prst_ctl_tim2	The time period when apb reset is removed but clock stopped. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us;; 0xF: 150.25us; other values: forbidden.	0x6
7:4	RW	cpu1_prst_ctl_tim1	The time period when apb reset is applied but clock stopped. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us;; 0xF: 150.25us; other values: forbidden.	0x6

CPU1_RST_CTL

Offset=0xc, Total Reset Value: 0x0000_2001, Write_enable=yes

Bits	Access	Name	Descriptions	Reset
13	W1C	cpu1_prst_done	CPU1 system apb resetting done status bit. Writing '1' to clear this status bit. 0x1: module resetting done; 0x0: no event.	0x0
12	W1C	cpu1_rst_done	CPU1 system resetting done status bit. Writing '1' to clear this status bit. 0x1: module resetting done; 0x0: no event.	0x0
4	RWSC	cpu1_flush_req	1: flush cpu l2chche, hard auto clear this bit when receive cpu flush done	0x0
0	RW	cpu1_reset_req	CPU1 accellerating subsystem reset request bit. 0: noreset 1: reset	0x1

AI_RST_TIM

Offset=0x10, Total Reset Value: 0x0000_0880, Write_enable=no

Bits	Access	Name	Descriptions	Reset
11:8	RW	ai_rst_ctl_tim2	The time period when reset is removed but clock stopped. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us;; 0xF: 150.25us; other values: forbidden.	0x8
7:4	RW	ai_rst_ctl_tim1	The time period when reset is applied but clock stopped. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us;; 0xF: 150.25us; other values: forbidden.	0x8

AI_RST_CTL

Offset=0x14, Total Reset Value: 0x8000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31	W1C	ai_rst_done	AI accellerating system resetting done status bit. Writing '1' to clear this status bit. 0x1: module resetting done; 0x0: no event.	0x0
0	W1T	ai_reset_req	AI accellerating subsystem reset request bit. 0x0: no action; 0x1: request to reset AI subsystem.	0x0

VPU_RST_TIM

Offset=0x18, Total Reset Value: 0x00000880, Write_enable=no

Bits	Access	Name	Descriptions	Reset
11:8	RW	vpu_rst_ctl_tim2	The time period when reset is removed but clock stopped. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us;; 0xF: 150.25us; other values: forbidden.	0x8

Bits	Access	Name	Descriptions	Reset
7:4	RW	vpu_rst_ctl_tim1	The time period when reset is applied but clock stopped. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us;; 0xF: 150.25us; other values: forbidden.	0x8

VPU_RST_CTL

Offset=0x1C, Total Reset Value: 0x8000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31	W1C	vpu_rst_done	VPU accellerating system resetting done status bit. Writing '1' to clear this status bit. 0x1: module resetting done; 0x0: no event.	0x0
0	W1T	vpu_reset_req	VPU accellerating subsystem reset request bit. 0x0: no action; 00x1: request to reset VPU subsystem.	0x0

SOC_CTL_RST_CTL

Offset=0x20, Total Reset Value: 0x000f_f0ff, Write_enable=no

Bits	Access	Name	Descriptions	Reset
19	RW	pmu_reset	Pmu apbreset 0: reset assert, 1: reset disassert	0x1
18	RW	stc_resetrn	Stc core reset 0: reset assert, 1: reset disassert	0x1
17	RW	mailbox_reset	Mailbox apb reset 0: reset assert, 1: reset disassert	0x1
16	RW	ts_preset	Ts apbreset 0: reset assert, 1: reset disassert	0x1
15	RW	wdt_1_preset	Wdt1 apb reset 0: reset assert, 1: reset disassert	0x1
14	RW	wdt_0_preset	Wdt0 apb reset 0: reset assert, 1: reset disassert	0x1
13	RW	wdt_1_reset	Wdt1 core reset 0: reset assert, 1: reset disassert	0x1
12	RW	wdt_0_reset	Wdt0 core reset 0: reset assert, 1: reset disassert	0x1
7	RW	hdi_resetrn	Hdi core reset 0: reset assert, 1: reset disassert	0x1
6	RW	timer_preset	Timer apb reset 0: reset assert, 1: reset disassert	0x1
5	RW	timer_5_reset	Timer core reset 0: reset assert, 1: reset disassert	0x1

Bits	Access	Name	Descriptions	Reset
4	RW	timer_4_reset	0: reset assert, 1: reset disassert	0x1
3	RW	timer_3_reset	0: reset assert, 1: reset disassert	0x1
2	RW	timer_2_reset	0: reset assert, 1: reset disassert	0x1
1	RW	timer_1_reset	0: reset assert, 1: reset disassert	0x1
0	RW	timer_0_reset	0: reset assert, 1: reset disassert	0x1

LOSYS_RST_CTL

Offset=0x24, Total Reset Value: 0x007e_7fff, Write_enable=no

Bits	Access	Name	Descriptions	Reset
22	RW	pwm_presetn	pwmmapb reset 0: reset assert, 1: reset disassert	0x1
21	RW	adc_presetn	Adc apb reset 0: reset assert, 1: reset disassert	0x1
20	RW	adc_resetrn	Adc reset 0: reset assert, 1: reset disassert	0x1
19	RW	gpio_presetn	Gpio apb reset 0: reset assert, 1: reset disassert	0x1
18	RW	gpio_db_resetrn	Gpio debounce clock reset 0: reset assert, 1: reset disassert	0x1
17	RW	codec_presetn	Audio apb reset 0: reset assert, 1: reset disassert	0x1
14	RW	jamlink3_presetn	jamlink3 apb reset 0: reset assert, 1: reset disassert	0x1
13	RW	jamlink2_presetn	jamlink2 apb reset 0: reset assert, 1: reset disassert	0x1
12	RW	jamlink1_presetn	jamlink1 apb reset 0: reset assert, 1: reset disassert	0x1
11	RW	jamlink0_presetn	jamlink0 apb reset 0: reset assert, 1: reset disassert	0x1
10	RW	i2c4_resetrn	I2c core reset 0: reset assert, 1: reset disassert	0x1
9	RW	i2c3_resetrn	0: reset assert, 1: reset disassert	0x1
8	RW	i2c2_resetrn	0: reset assert, 1: reset disassert	0x1
7	RW	i2c1_resetrn	0: reset assert, 1: reset disassert	0x1
6	RW	i2c0_resetrn	0: reset assert, 1: reset disassert	0x1
5	RW	uart4_resetrn	Uart core reset 0: reset assert, 1: reset disassert	0x1
4	RW	uart3_resetrn	0: reset assert, 1: reset disassert	0x1

Bits	Access	Name	Descriptions	Reset
3	RW	uart2_resetrn	0: reset assert, 1: reset disassert	0x1
2	RW	uart1_resetrn	0: reset assert, 1: reset disassert	0x1
1	RW	uart0_resetrn	0: reset assert, 1: reset disassert	0x1
0	RW	ls_presetn	Low speed sys apb reset 0: reset assert, 1: reset disassert	0x1

HISYS_HCLK_TIM

Offset=0x28, Total Reset Value: 0x0008_0800, Write_enable=no

Bits	Access	Name	Descriptions	Reset
19:16	RW	hisys_rst_ctl_tim2	The time period when reset is removed but clock stopped. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us;; 0xF: 150.25us; other values: forbidden.	0x8
11:8	RW	hisys_rst_ctl_tim1	The time period when reset is applied but clock stopped. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us;; 0x1F: 310.25us; other values: forbidden.	0x8

HISYS_RST_CTL

Offset=0x2c, Total Reset Value: 0x0000_0030, Write_enable=no

Bits	Access	Name	Descriptions	Reset
5	W1C	hisys_ahb_rst_done	hi sys 200M ahb bridge resetting done status bit. Writing '1' to clear this status bit. 0x1: module resetting done; 0x0: no event.	0x0
4	W1C	hisys_rst_done	hi sys 400M ahb bridge resetting done status bit. Writing '1' to clear this status bit. 0x1: module resetting done; 0x0: no event.	0x0
1	W1T	hisys_ahb_reset_req	hi sys ahb bridge reset,hs_hclk 200M domain	0x0
0	W1T	hisys_reset_req	hi sys ahb bridge reset,hs_hclk_high 400M domain	0x0

SDCTL_RST_TIM

Offset=0x30, Total Reset Value: 0x0008_0800, Write_enable=no |Bits |Name |Reset |Access |Descriptions|

Bits	Access	Name	Descriptions	Reset
19:16	RW	sdctl_rst_ctl_tim2	The time period when reset is removed but clock stopped. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us;; 0xF: 150.25us; other values: forbidden.	0x8
11:8	RW	sdctl_rst_ctl_tim1	The time period when reset is applied but clock stopped. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us;; 0x1F: 310.25us; other values: forbidden.	0x8

SDC_RST_CTL

Offset=0x34, Total Reset Value: 0x7000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
30	W1C	sd_c_arst_done	Sdio 2m1s bridge reset 0x1: module resetting done; 0x0: no event.	0x0
29	W1C	sd_c1_rst_done	SD host controller 1 resetting done status bit. Writing '1' to clear this status bit. 0x1: module resetting done; 0x0: no event.	0x0
28	W1C	sd_c0_rst_done	SD host controller 0 resetting done status bit. Writing '1' to clear this status bit. 0x1: module resetting done; 0x0: no event.	0x0
2	W1T	sd_c_aret_req	SD axi 2m1s bridge resetting request bit. 0x0: no action; 0x1: request to reset SD host controller 0.	0x0
1	W1T	sd_c1_reset_req	SD host controller 1 resetting request bit. 0x0: no action; 0x1: request to reset SD host controller 0.	0x0
0	W1T	sd_c0_reset_req	SD host controller 0 resetting request bit. 0x0: no action; 0x1: request to reset SD host controller 0.	0x0

USB_RST_TIM

Offset=0x38, Total Reset Value: 0x0008_d288, Write_enable=no

Bits	Access	Name	Descriptions	Reset
23:16	RW	usb_hreset_ctl_tim2	Usb ahb bridge reset timing register. The time period when reset is applied but clock stopped. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us; ; 0xF: 150.25us; other values: forbidden.	0x8
15:8	RW	usb_hreset_ctl_tim1	Usb ahb bridge reset timing register. The time period when reset is applied and clock keeps toggling. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us; ; 0xF: 150.25us; other values: forbidden.	0xd2
7:4	RW	usb_rst_ctl_tim2	Usb core reset timing register The time period when reset is applied but clock stopped. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us; ; 0xF: 150.25us; other values: forbidden.	0x8
3:0	RW	usb_rst_ctl_tim1	Usb core reset timing register . The time period when reset is applied and clock keeps toggling. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us; ; 0xF: 150.25us; other values: forbidden.	0x8

USB_RST_CTL

Offset=0x3c, Total Reset Value: 0xf000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31	W1C	usb1_hreset_done	USB module ahb resetting done status bit. Writing '1' to clear this status bit. 0x1: module resetting done; 0x0: no event.	0x0
30	W1C	usb0_hreset_done	USB module ahb resetting done status bit. Writing '1' to clear this status bit. 0x1: module resetting done; 0x0: no event.	0x0
29	W1C	usb1_rst_done	USB module por, prst_n resetting done status bit. Writing '1' to clear this status bit. 0x1: module resetting done; 0x0: no event.	0x0
28	W1C	usb0_rst_done	USB module por, prst_n resetting done status bit. Writing '1' to clear this status bit. 0x1: module resetting done; 0x0: no event.	0x0
1	W1T	usb1_reset_req	USB module resetting request bit. 0x0: no action; 0x1: request to reset USB.	0x0
0	W1T	usb0_reset_req	USB module resetting request bit. 0x0: no action; 0x1: request to reset USB.	0x0

SPI_RST_TIM

Offset=0x40, Total Reset Value: 0x0008_0c00, Write_enable=no

Bits	Access	Name	Descriptions	Reset
19:16	RW	spi_rst_ctl_tim2	The time period when reset is removed but clock stopped. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us;; 0xF: 150.25us; other values: forbidden.	0x8
11:8	RW	spi_rst_ctl_tim1	The time period when reset is applied but clock stopped. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us;; 0x3F: 630.25us; other values: forbidden.	0xc

SPI_RST_CTL

Offset=0x44, Total Reset Value: 0x7000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
30	W1C	spi2_rst_done	SPI 2 host module resetting done status bit. Writing '1' to clear this status bit. 0x1: module resetting done; 0x0: no event.	0x0
29	W1C	spi1_rst_done	SPI 1 host module resetting done status bit. Writing '1' to clear this status bit. 0x1: module resetting done; 0x0: no event.	0x0
28	W1C	spi0_rst_done	SPI 0 host module resetting done status bit. Writing '1' to clear this status bit. 0x1: module resetting done; 0x0: no event.	0x0

Bits	Access	Name	Descriptions	Reset
2	W1T	spi2_reset_req	SPI 2 host module reset request bit. 0x0: no action; 0x1: request to reset SPI 0 host module.	0x0
1	W1T	spi1_reset_req	SPI 1 host module reset request bit. 0x0: no action; 0x1: request to reset SPI 0 host module.	0x0
0	W1T	spi0_reset_req	SPI 0 host module reset request bit. 0x0: no action; 0x1: request to reset SPI 0 host module.	0x0

SEC_SYS_RST_TIM

Offset=0x48, Total Reset Value: 0x0008_0800, Write_enable=no

Bits	Access	Name	Descriptions	Reset
19:16	RW	sec_sys_rst_ctl_tim2	The time period when reset is removed but clock stopped. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us; ; 0xF: 150.25us; other values: forbidden.	0x8
11:8	RW	sec_sys_rst_ctl_tim1	The time period when reset is applied but clock stopped. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us; ; 0xFF: 2550.25us; other values: forbidden.	0x8

SEC_RST_CTL

Offset=0x4c, Total Reset Value: 0x8000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31	W1C	sec_rst_done	sec module resetting done status bit. Writing '1' to clear this status bit. 0x1: module resetting done; 0x0: no event.	0x0
0	W1T	sec_reset_req	sec host module reset request bit. 0x0: no action; 0x1: request to reset SPI 0 host module.	0x0

DMAC_RST_TIM

Offset=0x50, Total Reset Value: 0x0004_0400, Write_enable=no

Bits	Access	Name	Descriptions	Reset
19:16	RW	dmac_rst_ctl_tim2	The time period when reset is removed but clock stopped. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us; ; 0x7: 70.25us; other values: forbidden.	0x4
11:8	RW	dmac_rst_ctl_tim1	The time period when reset is applied but clock stopped. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us; ; 0x7: 70.25us; other values: forbidden.	0x4

DMA_RST_CTL

Offset=0x54, Total Reset Value: 0x3000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
29	W1C	sdma_rst_done	system DMAC resetting done status bit. Writing '1' to clear this status bit. 0x1: module resetting done; 0x0: no event.	0x0
28	W1C	pdma_rst_done	Peripheral DMAC resetting done status bit. Writing '1' to clear this status bit. 0x1: module resetting done; 0x0: no event.	0x0
1	W1T	sdma_reset_req	system DMAC module resetting request bit. 0x0: no action; 0x1: request to reset Peripheral DMAC.	0x0
0	W1T	pdma_reset_req	Peripheral DMAC module resetting request bit. 0x0: no action; 0x1: request to reset Peripheral DMAC.	0x0

DECOMPRESS_RST_TIM

Offset=0x58, Total Reset Value: 0x0004_0400, Write_enable=no

Bits	Access	Name	Descriptions	Reset
19:16	RW	decompress_rst_ctl_tim2	The time period when reset is removed but clock stopped. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us;; 0x7: 70.25us; other values: forbidden.	0x4
11:8	RW	decompress_rst_ctl_tim1	The time period when reset is applied but clock stopped. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us;; 0x7: 70.25us; other values: forbidden.	0x4

DECOMPRESS_RST_CTL

Offset=0x5c, Total Reset Value: 0x8000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31	W1C	decompress_rst_done	decompress resetting done status bit. Writing '1' to clear this status bit. 0x1: module resetting done; 0x0: no event.	0x0
0	W1T	decompress_reset_req	decompress module resetting request bit. 0x0: no action; 0x1: request to reset Peripheral DMAC.	0x0

SRAM_RST_TIM

Offset=0x60, Total Reset Value: 0x0002_0200, Write_enable=no

Bits	Access	Name	Descriptions	Reset
19:16	RW	sram_rst_ctl_tim2	The time period when reset is removed but clock stopped. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us;; 0xF: 150.25us; other values: forbidden.	0x2

Bits	Access	Name	Descriptions	Reset
11:8	RW	sram_rst_ctl_tim1	The time period when reset is applied but clock stopped. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us; ; 0xF: 150.25us; other values: forbidden.	0x2

SRAM_RST_CTL

Offset=0x64, Total Reset Value: 0xd000_0002, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31	W1C	sram_axis_rst_done	Shrm sys axi slave resetting done status bit. Writing '1' to clear this status bit. 0x1: module resetting done; 0x0: no event.	0x0
30	W1C	sram_axim_rst_done	shrm sys axi master resetting done status bit. Writing '1' to clear this status bit. 0x1: module resetting done; 0x0: no event.	0x0
28	W1C	sram_rst_done	1: Shrm sys sram reset done	0x0
3	W1T	shrm_axis_rst_req	1: shrm axi slave reset request	0x0
2	W1T	shrm_axim_rst_req	1: shrm axi master reset request	0x0
1	RW	shrm_prst_req	0: storeage system apb bridge reset	0x1
0	W1T	sram_reset_req	SRAM resetting request bit. 0x0: no action; 0x1: request to reset SRAM.	0x0

NONAI2D_RST_TIM

Offset=0x68, Total Reset Value: 0x0002_0200, Write_enable=no

Bits	Access	Name	Descriptions	Reset
19:16	RW	nonai2d_rst_ctl_tim2	The time period when reset is removed but clock stopped. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us; ; 0xF: 150.25us; other values: forbidden.	0x2
11:8	RW	nonai2d_rst_ctl_tim1	The time period when reset is applied but clock stopped. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us; ; 0xF: 150.25us; other values: forbidden.	0x2

NONAI2D_RST_CTL

Offset=0x6c, Total Reset Value: 0x8000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31	W1C	nonai2d_rst_done	nonai2d resetting done status bit. Writing '1' to clear this status bit. 0x1: module resetting done; 0x0: no event.	0x0
0	W1T	nonai2d_reset_req	nonai2d module resetting request bit. 0x0: no action; 0x1: request to reset Peripheral DMAC.	0x0

MCTL_RST_TIM

Offset=0x70, Total Reset Value: 0x0000_0304, Write_enable=no

Bits	Access	Name	Descriptions	Reset
13:8	RW	pwrokin_tim	pwrokin remain low time.	0x3
5:0	RW	mctl_rst_ctl_tim0	The time period between ddrc_reset is applied and ddrc_reset is removed. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us;; 0xF: 150.25us; other values: forbidden.	0x4

MCTL_RST_CTL

Offset=0x74, Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31	W1C	mctl_rst_done	Memory controller resetting done status bit. Writing '1' to clear this status bit. 0x1: module resetting done; 0x0: no event.	0x0
0	W1T	mctl_reset_req	Memory controller resetting request bit. 0x0: no action; 0x1: request to reset memory controller.	0x0

ISP_RST_TIM

Offset=0x78, Total Reset Value: 0x0003_0202, Write_enable=no

Bits	Access	Name	Descriptions	Reset
19:16	RW	isp_rst_ctl_tim2	ISP reset time period when reset is removed but clock stopped. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us;; 0xF: 150.25us; other values: forbidden.	0x3
11:8	RW	isp_rst_ctl_tim1	ISP reset time period when reset is applied but clock stopped. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us;; 0xF: 150.25us; other values: forbidden.	0x2
3:0	RW	isp_rst_ctl_tim0	The time period when reset is applied and clock enabled. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us;; 0xFF: 2550.25us; other values: forbidden.	0x2

ISP_DW_RST_TIM

Offset=0x7c, Total Reset Value: 0x0002_0202, Write_enable=no

Bits	Access	Name	Descriptions	Reset
19:16	RW	isp_dw_rst_ctl_tim2	ISP dw reset time period when reset is removed but clock stopped. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us;; 0xF: 150.25us; other values: forbidden.	0x2

Bits	Access	Name	Descriptions	Reset
11:8	RW	isp_dw_rst_ctl_tim1	ISP dw reset time period when reset is applied but clock stopped. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us; ; 0xF: 150.25us; other values: forbidden.	0x2
3:0	RW	isp_dw_rst_ctl_tim0	The time period when reset is applied and clock enabled. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us; ; 0xFF: 2550.25us; other values: forbidden.	0x2

ISP_RST_CTL

Offset=0x80, Total Reset Value: 0x3000_039f, Write_enable=no

Bits	Access	Name	Descriptions	Reset
29	W1C	isp_rst_done	CSI0 module resetting done status bit. Writing '1' to clear this status bit. 0x1: module resetting done; 0x0: no event.	0x0
28	W1C	isp_dw_rst_done	CSI0 module resetting done status bit. Writing '1' to clear this status bit. 0x1: module resetting done; 0x0: no event.	0x0
9	RW	m2_reset_req	Sensor 0 reset, 0: reset	0x1
8	RW	m1_reset_req	Sensor 1 reset, 0: reset	0x1
7	RW	m0_reset_req	Sensor 2 reset, 0: reset	0x1
6	W1T	isp_reset_req	ISP resetting request bit. 0x0: no action; 0x1: request to reset CSI0.	0x0
5	W1T	isp_dw_reset_req	ISP DW resetting request bit. 0x0: no action; 0x1: request to reset CSI0.	0x0
4	RW	isp_hreset_req	vi_hresetn, ahb abridge inside vi sys. 0: reset, 1: not reset	0x1
3	RW	csi_dphy_reset_req	dphy_presetn, csi dphy apb reset. 0: reset, 1: not reset	0x1
2	RW	csi2_reset_req	csi2_presetn, csi2 apb resetn, 0: reset, 1: not reset	0x1
1	RW	csi1_reset_req	csi1_presetn, csi2 apb resetn, 0: reset, 1: not reset	0x1
0	RW	csi0_reset_req	csi0_presetn, csi2 apb resetn, 0: reset, 1: not reset	0x1

DPU_RST_TIM

Offset=0x84, Total Reset Value: 0x0002_0200, Write_enable=no

Bits	Access	Name	Descriptions	Reset
19:16	RW	dpu_rst_ctl_tim2	The time period when reset is removed but clock stopped. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us; ; 0xF: 150.25us; other values: forbidden.	0x2

Bits	Access	Name	Descriptions	Reset
11:8	RW	dpu_rst_ctl_tim1	The time period when reset is applied but clock stopped. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us;; 0xF: 150.25us; other values: forbidden.	0x2

DPU_RST_CTL

Offset=0x88, Total Reset Value: 0x8000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31	W1C	dpu_rst_done	DPU host module resetting done status bit. Writing '1' to clear this status bit. 0x1: module resetting done; 0x0: no event.	0x0
0	W1T	dpu_reset_req	DPU module reset request bit. 0x0: no action; 0x1: request to reset UART 0 host module.	0x0

DISP_SYS_RST_TIM

Offset=0x8c, Total Reset Value: 0x0004_0404, Write_enable=no

Bits	Access	Name	Descriptions	Reset
19:16	RW	disp_sys_rst_ctl_tim2	The time period when reset is removed but clock stopped. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us;; 0xF: 150.25us; other values: forbidden.	0x4
11:8	RW	disp_sys_rst_ctl_tim1	The time period when reset is applied but clock stopped. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us;; 0xFF: 2550.25us; other values: forbidden.	0x4
3:0	RW	disp_sys_rst_ctl_tim0	The time period when reset is applied and clock enabled. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us;; 0xFF: 2550.25us; other values: forbidden.	0x4

DISP_RST_CTL

Offset=0x90, Total Reset Value: 0x8000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31	W1C	disp_rst_done	Display module resetting done status bit. Writing '1' to clear this status bit. 0x1: module resetting done; 0x0: no event.	0x0
0	W1T	disp_reset_req	Display module reset request bit. 0x0: no action; 0x1: request to reset UART 0 host module.	0x0

V2P5D_SYS_RST_TIM

Offset=0x94, Total Reset Value: 0x0004_0404, Write_enable=no

Bits	Access	Name	Descriptions	Reset

Bits	Access	Name	Descriptions	Reset
19:16	RW	v2p5d_rst_ctl_tim2	The time period when reset is removed but clock stopped. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us;; 0xF: 150.25us; other values: forbidden.	0x4
11:8	RW	v2p5d_rst_ctl_tim1	The time period when reset is applied but clock stopped. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us;; 0xFF: 2550.25us; other values: forbidden.	0x4
3:0	RW	v2p5d_rst_ctl_tim0	The time period when reset is applied and clock enabled. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us;; 0xFF: 2550.25us; other values: forbidden.	0x4

V2P5D_RST_CTL

Offset=0x98, Total Reset Value: 0x8000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31	W1C	v2p5d_rst_done	gpu module resetting done status bit. Writing '1' to clear this status bit. 0x1: module resetting done; 0x0: no event.	0x0
0	W1T	v2p5d_reset_req	Gpu module reset request bit. 0x0: no action; 0x1: request to reset UART 0 host module.	0x0

AUDIO_RST_TIM

Offset=0xa0, Total Reset Value: 0x0000_0880, Write_enable=no

Bits	Access	Name	Descriptions	Reset
11:8	RW	audio_rst_ctl_tim2	The time period when reset is removed but clock stopped. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us;; 0xF: 150.25us; other values: forbidden.	0x8
7:4	RW	audio_rst_ctl_tim1	The time period when reset is applied but clock stopped. 0x1: 10.25us; 0x2: 20.25us; 0x3: 30.25us;; 0xF: 150.25us; other values: forbidden.	0x8

AUDIO_RST_CTL

Offset=0xa4, Total Reset Value: 0x8000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31	W1C	audio_rst_done	Audio resetting done status bit. Writing '1' to clear this status bit. 0x1: module resetting done; 0x0: no event.	0x0
0	W1T	audio_reset_req	Audio reset request bit. 0x0: no action; 0x1: request to reset AI subsystem.	0x0

2.1.6 Application note

- It is not recommended to reset sub-module configuration interface regardless of unfinished bus transaction.
- It is recommend to check sub-module status before apply reset.
- It is recommend to negotiate through NOC LPI interface to ensure current bus interface is at idle state and then apply sub-module reset.

2.2 Clock

2.2.1 Overview

Clock management module (CMU) is designed to manage generation of sub-module and system clock、 clock division、 clock switch and clock gating. The clock source include external 24MHz OSC and four internal PLLs.

CPU1 and KPU module support DVFS. Their frequency should meet current voltage. When CPU1 and KPU use the same clock source, their frequency can be changed synchronously. When CPU1 and KPU use different clock source, their frequency can be changed independently.

2.2.2 Block diagram

Figure 2-2-1 clock block diagram

![clock block diagram](./clock-diagram.png)

2.2.3 Function description

2.2.3.1 Clock source

There are five clock source : one external 24MHz OSC and four internal PLLs.

2.2.3.2 Apb interface

CPU can configure clock module through apb interface.

2.2.3.3 PLL

Clock-Generator PLL Specifications:

Divided reference frequency range	488KHz - 4GHz
Total output frequency range	50MHz - 4GHz
/1 output frequency range	800MHz - 4GHz
Reference divider values	1-64
Feedback divider values	1-4096
Output divider values	1, 2-16 (even only)
/1 output multiples of div. reference	1-4096
Bandwidth adjustment div. range	1-4096
Reset /1 output frequency range	20MHz - 200MHz
Lock time (min allowed)	500 div. reference cycles (actual lock time will be much smaller)
Ref. input jitter (long-term, P-P) (max)	2% div. reference cycle

Reference H/L pulse width (min)	120ps

The output frequency F_{out} at CLKOUT is related to the reference frequency F_{ref} by:

$$F_{out} = F_{ref} * NF / NR / OD$$

PLL counters:

1. CLKR: A 6-bit bus that selects the values 1-64 for the reference divider (NR)

$$NR = CLKR[5:0] + 1$$

2. CLKF: A 13-bit bus that selects the values 1-4096 for the multiplication factor (NF)

$$NF = CLKF[12:0] + 1$$

3. CLKOD: A 4-bit bus that selects the values 1-16 for the post VCO divider (OD)

$$OD = CLKOD[3:0] + 1$$

4. BWADJ: A 12-bit bus that selects the values 1-4096 for the bandwidth divider (NB')

$$NB' = BWADJ[11:0] + 1$$

$$NB = NB' / NB\_ADJ$$

$$NB\_ADJ = (NB\_ADJ1 < 1)? 1 : (NB\_ADJ1 > 1)? 1 : NB\_ADJ1$$

$$NB\_ADJ1 = 1 / \sqrt{(F_{ref}/NR) / 32KHz}$$

PLL configure sequence:

- Configure PLLx_CTL.pllx_pwrn=1 , PLL output will be gated
- Configure PLL parameters PLLx_CFG0 and PLLx_CFG1
- Configure PLLx_CTL.pllx_init , hardware will wait lock automatically and open PLL clock output

Table 2-2-1 PLL default configuration

PLL	frequency	Module requirement
PLL0	1.6GHz	DDR 533 , 1066 , 1600 , 2133 , 3200
PLL1	2.376GHz	SensorMCLK 74.25MHz,72MHz,54MHz,37.125MHz,36MHz, 27MHz,24MHz,12MHz
PLL2	2.667GHz	DDR 2667 1333
PLL3	1.6GHz	For cpu1 and kpu DVFS

2.2.3.4 DDR Clock

K230 support following DDR requirement:

3200Mbps/2667Mbps/1333Mbps/2133 Mbps/1600Mbps/1066Mbps/800Mbps/533Mbps

To meet these requirement three clock source are used to generate ddrc_core_clk:

pll0_oclk_div2,pll0_oclk_div3,pll2_oclk_div4.

K230 DDR support hardware fast frequency change. Max to four groups of ddr configuration can be saved prior to hardware fast frequency change.

hardware fast frequency change flow:

- Configure MCTL_CLK_SWITCH.csysmode to 1 to enable hardware fast frequency change.
- Configure MCTL_CLK_SWITCH.csysfrequency_ddrc, MCTL_CLK_SWITCH.csysdiscamdrain_ddrc to select correct switch group.
- Configure DDR_CLK_CFG to control clock mux and clock divider.
- Configure MCTL_PWR_LPI_CTL.mctl_go_idle to start frequency change.

2.2.3.5 CPU Clock

CPU1 clock is generated from three clock source : pll0_oclk,pll1_oclk_div2,pll3_oclk. CPU clock and bus clock can be configured as N:1 (N>=2).

For DVFS, PLL3 can be used to generate flexible frequency.

frequency upgrade flow:

- Change CPU1 clock source to PLL0.
- Change PLL3 configuration to meet voltage requirement and wait PLL lock.
- Upgrade CPU1 voltage.
- Switch CPU1 clock source to PLL3.

To accelerate CPU1 DVFS frequency switch, the second step can be performed in advance.

frequency downgrade flow:

- Change CPU1 clock source to PLL0.
- Change PLL3 configuration to meet voltage requirement and wait PLL lock.
- Switch CPU1 clock source to PLL3.
- Downgrade CPU1 voltage.

To accelerate CPU1 DVFS frequency switch, the second step can be performed in advance.

2.2.3.6 sensor Clock

The camera sensor clock must be adequate for different resolution. MCLKn can be set as :

Table 2-2-2 MCLK configuration

frequency	clock source	clock divider
74.25MHz	PlI1_oclk_div4 (594MHz)	8
72MHz	PlI1_oclk_div3 (792MHz)	11
54MHz	PlI1_oclk_div4 (594MHz)	11
37.125MHz	PlI1_oclk_div4 (594MHz)	16
36MHz	PlI1_oclk_div3 (792MHz)	22
27MHz	PlI1_oclk_div4 (594MHz)	22

2.2.3.7 AI Clock

ai_clk has two clock source : pll0_ockl_div2,pll3_ockl_div2. For DVFS, PLL3 can be used to generate flexible frequency.

frequency upgrade flow:

- Change AI clock source to PLL0.
- Change PLL3 configuration to meet voltage requirement and wait PLL lock.
- Upgrade ai voltage.
- Switch ai_clk clock source to PLL3.

To accelerate ai DVFS frequency switch, the second step can be performed in advance.

frequency downgrade flow:

- Change ai clock source to PLL0.
- Change PLL3 configuration to meet voltage requirement and wait PLL lock.
- Switch ai clock source to PLL3.
- Downgrade ai voltage.

To accelerate ai DVFS frequency switch, the second step can be performed in advance.

2.2.3.8 SPI Clock

K230 OSPI support DDR mode, spi0_clk must be large or equal to 4 times of spi serial clock. For QSPI, spi1_clk and spi2_clk must be large or equal to 2 times of spi serial clock.

2.2.3.9 audio Clock

Audio I2S clock aud_dev_clk must be 64 times as sample rate. K230 support following sample rate: 8kHz/12kHz/16kHz/24kHz/32kHz/44.1KHz/48KHz/96KHz/192KHz

Table 2-2-3 audio I2S clock configuration

sample rate	clock frequency	AUDIO_CLKDIV_CFG
8kHz	512k	audio_clk_sum=3125 audio_clk_step=4
12kHz	768k	audio_clk_sum=3125 audio_clk_step=6
16kHz	1.024M	audio_clk_sum=3125 audio_clk_step=8
24kHz	1.536M	audio_clk_sum=3125 audio_clk_step=12
32kHz	2.048M	audio_clk_sum=3125 audio_clk_step=16
44.1KHz	2.8224M	audio_clk_sum=62500 audio_clk_step=441
48KHz	3.072	audio_clk_sum=3125 audio_clk_step=24
96KHz	6.144	audio_clk_sum=3125 audio_clk_step=48

sample rate	clock frequency	AUDIO_CLKDIV_CFG
192KHz	12.288M	audio_clk_sum=3125 audio_clk_step=96

Table 2-2-4 audio PDM clock configuration

sample rate	clock frequency	PDM_CLKDIV_CFG
4kHz	256k	pdm_clk_sum=3125 pdm_clk_step=2
6kHz	384k	pdm_clk_sum=3125 pdm_clk_step=3
8kHz	512k	pdm_clk_sum=3125 pdm_clk_step=4
12kHz	768k	pdm_clk_sum=3125 pdm_clk_step=6
16kHz	1.024M	pdm_clk_sum=3125 pdm_clk_step=8
22.05kHz	1.4112M	pdm_clk_sum=125000 pdm_clk_step=441
24kHz	1.536M	pdm_clk_sum=3125 pdm_clk_step=12
32kHz	2.048M	pdm_clk_sum=3125 pdm_clk_step=16
44.1KHz	2.8224M	pdm_clk_sum=62500 pdm_clk_step=441
48KHz	3.072	pdm_clk_sum=3125 pdm_clk_step=24
64kHz	4.096	pdm_clk_sum=3125 pdm_clk_step=32
88.2kHz	5.6448	pdm_clk_sum=31250 pdm_clk_step=441
96KHz	6.144	pdm_clk_sum=3125 pdm_clk_step=48
192KHz	12.288M	pdm_clk_sum=3125 pdm_clk_step=96
384KHz	24.576M	pdm_clk_sum=3125 pdm_clk_step=192

Audio codec clock codec_mclk_adc and codec_mclk_dac must be 256 times as sample rate. K230 support following sample rate: 8kHz/12kHz/16kHz/24kHz/32kHz/44.1KHz/48KHz/96KHz/192KHz

Table 2-2-5 audio codec clock configuration

sample rate	clock frequency	CODEC_ADC_MCLKDIV_CFG/ CODEC_DAC_MCLKDIV_CFG
8kHz	2.048M	codec_mclk_dac(adc)_sum=3125 codec_mclk_dac(adc)_step=16
12kHz	3.072	codec_mclk_dac(adc)_sum=3125 codec_mclk_dac(adc)_step=24
16kHz	4.096	codec_mclk_dac(adc)_sum=3125 codec_mclk_dac(adc)_step=32
24kHz	6.144	codec_mclk_dac(adc)_sum=3125 codec_mclk_dac(adc)_step=48
32kHz	8.192	codec_mclk_dac(adc)_sum=3125 codec_mclk_dac(adc)_step=64
44.1KHz	11.2896	codec_mclk_dac(adc)_sum=15625 codec_mclk_dac(adc)_step=441
48KHz	12.288	codec_mclk_dac(adc)_sum=3125 codec_mclk_dac(adc)_step=96
96KHz	24.576	codec_mclk_dac(adc)_sum=3125 codec_mclk_dac(adc)_step=192
192KHz	49.152	codec_mclk_dac(adc)_sum=3125 codec_mclk_dac(adc)_step=384

2.2.3.10 shrm clock

There are two source for sram_clk: pll0_oclk_div2,pll3_oclk_div2. For normal operation sram_clk and ai_clk use the same PLL as clock source. That is sram_clk source come from pll0_oclk_div2.

When ai_clk frequency increase above 800MHz, sram_clk can not come up with ai_clk. For this circumstance sram_clk clock source should use clock pll0_oclk_div2.

2.2.3.11 Clock distribution

Table 2-2-6 clock mux

clock	source
cpu1_core_clk	pll0_oclk,pll1_oclk_div2,pll3_oclk
ai_clk	pll0_oclk_div2,pll3_oclk_div2
usb0_refclk	OSC24M,usb_refclk_50m
usb1_refclk	OSC24M,usb_refclk_50m
spi0_clk	pll0_oclk_div2,pll2_oclk_div4
timer0_clk	timer_clk0,timer0_pulse_in
timer1_clk	timer_clk1,timer1_pulse_in
timer2_clk	timer_clk2,timer2_pulse_in
timer3_clk	timer_clk3,timer3_pulse_in

clock	source
timer4_clk	timer_clk4,timer4_pulse_in
timer5_clk	timer_clk5,timer5_pulse_in
sram_clk	pll0_ock_div2,pll3_ock_div2
ddrc_core_clk	pll0_ock_div2,pll0_ock_div3,pll2_ock_div4
isp_clk	pll0_ock_div4,pll2_ock_div4
mclk0	pll1_ock_div3,pll1_ock_div4,pll0_ock_div4
mclk1	pll1_ock_div3,pll1_ock_div4,pll0_ock_div4
mclk2	pll1_ock_div3,pll1_ock_div4,pll0_ock_div4

Table2-2-7 clock frequency

clk_name	freq	default_freq
OSC24M	24MHz	24MHz
PLL0_ock	1.6GHz	1.6GHz
PLL1_ock	2.376GHz	2.376GHz
PLL2_ock	2.667GHz	2.667GHz
PLL3_ock	1.6GHz	1.6GHz
timer0_pulse_in	50MHz	50MHz
timer1_pulse_in	50MHz	50MHz
timer2_pulse_in	50MHz	50MHz
timer3_pulse_in	50MHz	50MHz
timer4_pulse_in	50MHz	50MHz
cpu1_core_clk	1.6G	800M
cpu1_pliclk	800M	400M
cpu1_aclk	800M	400M
cpu_pclk	200M	200M
cpu0_core_clk	800M	800M
cpu0_pliclk	400M	400M
cpu0_aclk	400M	400M
cpu0_pclk	200M	200M
ai_clk	800M	100M
ai_aclk	400M	400M
ddrc_p3_aclk	400MHz	400MHz
vpu_clk	800M	800M
vpu_aclk	400M	400M

clk_name	freq	default_freq
ddrc_p2_aclk	400M	400M
vpu_cfgclk	200M	200M
pmu_pclk	24M	24M
hs_hclk_high	400M	400M
hs_hclk	200M	200M
sd_aclk	333.375MHz	333.375MHz
sd_bclk	333.375MHz	333.375MHz
sd_cclk	200M	200M
qspi_aclk	200M	200M
sd_tmclk	1M	1M
usb0_hclk	200M	200M
usb_refclk_50m	50M	50M
usb0_refclk	50M	24M
usb1_hclk	200M	200M
usb1_refclk	50M	24M
spi0_clk	800M	800M
spi0_aclk	400M	400M
spi1_clk	200M	200M
spi2_clk	200M	200M
ls_pclk	200M	100M
uart0_sclk	50M	50M
uart1_sclk	50M	50M
uart2_sclk	50M	50M
uart3_sclk	50M	50M
uart4_sclk	50M	50M
i2c0_clk	200MHz	100M
i2c1_clk	200MHz	100M
i2c2_clk	200MHz	100M
i2c3_clk	200MHz	100M
i2c4_clk	200MHz	100M
adc_clk	16M	16M
gpio_dbclk	32K	32K
codec_mclk_adc	50M	11.43M
aud_dev_clk	12.5M	2.84M

clk_name	freq	default_freq
sec_aclk	594M	594M
fixclk	49.5M	49.5M
sec_pclk	200M	100M
sys_ctl_pclk	100M	100M
wdt0_pclk	100M	100M
wdt0_clk	24M	750K
wdt1_pclk	100M	100M
wdt1_clk	24M	750K
timer_pclk	100M	100M
timer0_clk	50M	50M
timer1_clk	50M	50M
timer2_clk	50M	50M
timer3_clk	50M	50M
timer4_clk	50M	50M
timer5_clk	50M	50M
mailbox_pclk	100M	100M
stc_clk	27M	27M
ts_clk	100K	100K
ts_pclk	100M	100M
iomux_pclk	100M	100M
shrm_pclk	200M	100M
shrm_axim_clk	400M	400M
sram_clk	800M	800M
shrm_axis_clk	400M	400M
ddrc_core_clk	800M	533M
ddrc_pclk	200M	200M
ddrc_p0_aclk	800M	400MHz
ddrc_p4_aclk	400M	400M
ddr_bypassclk	667M	667M
isp_cfg_clk	27M	27M
csi0_pixel_clk	333.375M	333.375M
csi1_pixel_clk	333.375M	333.375M
csi2_pixel_clk	333.375M	333.375M
isp_clk	400M	400M

clk_name	freq	default_freq
isp_mcm_clk	400M	400M
isp_hclk	200M	200M
isp_aclk	400M	400M
ddrc_p1_aclk	400MHz	400M
ispdwe_clk	533M	533M
ispvse_clk	533M	533M
mclk0	74.25M	74.25M
mclk1	74.25M	74.25M
mclk2	74.25M	74.25M
dpu_clk	594M	594M
dpu_aclk	400M	400M
dpu_pclk	100M	100M
disp_hclk	200M	100M
disp_aclk	400M	400M
dpipclk	148.5M	74.25M
disp_refclk	24M	24M
disp_cfg_clk	27M	27M
disp_clkext	533M	533M
clk_gpu	533M	533M
spi2axi_aclk	100M	100M
usb_clk480	475.2M	475.2M
usb_clk100	100M	100M
dphy_test_clk	320M	320M
hdi_clk	400M	200M
sys_ctl_gpio_dbclk	32K	32K
codec_mclk_dac	50M	11.43M
pdm_clk_root	25M	2.84M
cpu1_pclk	200M	200M
tdi_pclk	200M	200M

2.2.4 Register summary

PLL configuration register base address: refer to [boot base address](#) Table 2-2-8 PLL configure

Offset	Register	Description
0x0	PLL0_CFG0	pll0 configuration

Offset	Register	Description
0x4	PLL0_CFG1	pll0 configuration
0x8	PLL0_CTL	pll0 configuration
0xC	PLL0_STAT	pll0 status
0x10	PLL1_CFG0	pll1 configuration
0x14	PLL1_CFG1	pll1 configuration
0x18	PLL1_CTL	pll1 configuration
0x1C	PLL1_STAT	pll1 status
0x20	PLL2_CFG0	pll2 configuration
0x24	PLL2_CFG1	pll2 configuration
0x28	PLL2_CTL	pll2 configuration
0x2C	PLL2_STAT	pll2 status
0x30	PLL3_CFG0	pll3 configuration
0x34	PLL3_CFG1	pll3 configuration
0x38	PLL3_CTL	pll3 configuration
0x3C	PLL3_STAT	pll3 status

clock configuration register base address: refet to [CMU base addrss](#) Table 2-2-9 clk configure

Offset	Register	description
0x0	CPU0_CLK_CFG	cpu0 clk configuration
0x4	CPU1_CLK_CFG	cpu1 clk cofiguration
0x8	ai_CLK_CFG	ai clk cofiguration
0xc	vpu_CLK_CFG	vpu clk cofiguration
0x10	pmu_CLK_CFG	pmu clk cofiguration
0x18	hs_CLKen_CFG	high speed system clk cofiguration
0x1c	hs_SDCLK_CFG	SD clk cofiguration
0x20	hs_SPI_CFG	SPI clk cofiguration
0x24	LS_CLKEN_CFG0	low speed system clk cofiguration
0x28	LS_CLKEN_CFG1	low speed system clk cofiguration
0x2c	UART_I2C_CLKDIV_CFG	low speed uart, iic clock divider configuration
0x30	LS_CLKDIV_CFG	low speed clk divider configuration
0x34	AUDIO_CLKDIV_CFG	audio clk divider cofiguration
0x38	CODEC_ADC_MCLKDIV_CFG	audio codec adc clk divider cofiguration
0x3c	CODEC_DAC_MCLKDIV_CFG	audio codec dac clk divider cofiguration
0x40	PDM_CLKDIV_CFG0	pdm clk divider cofiguration

Offset	Register	description
0x44	PDM_CLKDIV_CFG1	pdm clk divider cofiguration
0x50	SYSCTL_CLKEN_CFG	sys ctrl clk cofiguration
0x54	TIMER_CLK_CFG	timer clk cofiguration
0x58	SYSCTL_CLK_DIV_CFG	sys ctl clk divider cofiguration
0x5c	shrm_CLK_CFG	share memory system clk cofiguration
0x60	DDR_CLK_CFG	DDR system clk cofiguration
0x64	ISP_CLKEN_CFG	ISP clk cofiguration
0x68	ISP_CLKDIV_CFG	ISP clk divider cofiguration
0x6c	MCLK_CFG	MCLK clk cofiguration
0x70	DPU_CLK_CFG	DPU clk cofiguration
0x74	VO_CLK_CFG	display clk cofiguration
0x78	DISP_CLK_DIV	display clk divider cofiguration
0x80	SEC_CLK_DIV	security clk divider cofiguration
0x100	USB_TEST_CLK_DIV	usb test clk divider cofiguration
0x104	DPHY_TEST_CLK_DIV	MIPI dphy test clk divider cofiguration

2.2.5 Register description

PLL0_CFG0

Offset=0x0, Total Reset Value: 0x0102_018f, Write_enable=no

Bits	Access	Name	Descriptions	Reset
27:24	RW	pll0_out_div	PLL 0 VCO output clock post division ratio. It is strongly suggested not to modify this value. 0x0: 1/1 division; 0x1: 1/2 division; 0x2: 1/3 division; 0x3: 1/4 division;; 0xF: 1/16 division.	0x1
21:16	RW	pll0_ref_div	PLL 0 reference clock path division ratio. 0x0: 1/1 division; 0x1: 1/2 division; 0x2: 1/3 division; 0x3: 1/4 division;; 0x3F: 1/64 division.	0x2
12:0	RW	pll0_fb_div	PLL 0 feedback path clock division ratio. 0x0: 1/1 division; 0x1: 1/2 division; 0x2: 1/3 division; 0x3: 1/4 division;; 0xFFF: 1/4096 division; Other values: forbidden.	0x18f

PLL0_CFG1

Offset=0x4, Total Reset Value: 0x0002_00c7, Write_enable=no

Bits	Access	Name	Descriptions	Reset

Bits	Access	Name	Descriptions	Reset
19	RW	pll0_bypass	PLL 0 bypass mode enable control bit. When bypass mode, reference clock is directly bypass to PLL output clock. 0x0: disable PLL bypass mode; 0x1: enable PLL bypass mode.	0x0
18	RW	pll0_test	PLL 0 test mode enable control bit. 0x0: disable PLL test mode; 0x1: enable PLL test mode.	0x0
17	RW	pll0_ensat	PLL 0 saturation behavior enable control bit. Enabling saturation behavior can ensures analog circuits being biased at a optimal value. It's strongly recommended to set this bit to '1'. 0x0: disable PLL saturation behavior; 0x1: enable PLL saturation behavior.	0x1
16	RW	pll0_fasten	PLL 0 Fast locking circuit enable control bit. When this bit is set to '1', PLL output clock can reach lock state within far less time. It's strongly recommended to set this bit to '0'. 0x0: disable PLL fast locking circuit; 0x1: enable PLL fast locking circuit.	0x0
11:0	RW	pll0_bwadj	PLL 0 Bandwidth Adjustment parameter. The bandwdith can affect the quality of VCO output clock. The recommended value for this field is FBK_DIV/2. Valid values range from 0 to 0xFFFF.	0xc7

PLL0_CTL

Offset=0x8, Total Reset Value: 0x0000_0034, Write_enable=yes

Bits	Access	Name	Descriptions	Reset
11:4	RW	pll0_lock_tim	The time required for PLL 0 to reach lock state. The time unit for this timer is a cycle of 32Khz clock, about 31.25 us. N: N+1 time unit	0x3
2	RW	pll0_clk_oe	PLL1 VCO output clock enable control bit. 0x0: Disable VCO output clock; 0x1: Enable VCO output clock;	0x1
1	W1T	pll0_init	PLL 0 initialization trigger bit. 0x0: no operation; 0x1: start PLL initialization sequence.	0x0
0	W1T	pll0_pwrdown	PLL 0 go power-down mode trigger bit. 0x0: no operation; 0x1: put PLL 0 into power-down mode.	0x0

PLL0_STAT

Offset=0xC, Total Reset Value: 0x0000_0021, Write_enable=no

Bits	Access	Name	Descriptions	Reset

Bits	Access	Name	Descriptions	Reset
5:4	RO	pll0_fsm_stat	PLL 0 controlling FSM current status. 0x0: PLL FSM is in POWER_DOWN state; 0x1: PLL FSM is in RESET_PLL state; 0x3: PLL FSM is in WAIT_LOCK state; 0x2: PLL FSM is in PLL_READY state;	0x0
2	RO	pll0_fb_slip	This is the feedback clock path frequency/phase slip status. 0x0: PLL working OK; 0x1: There are frequency or phase slip on feedback clock.	0x0
1	RO	pll0_ref_slip	This is the reference clock path frequency/phase slip status. 0x0: PLL working OK; 0x1: There are frequency or phase slip on reference clock.	0x0
0	RO	pll0_lock	PLL 0 current lock status. 0x0: PLL not in lock state; 0x1: PLL in lock state.	0x0

PLL1_CFG0

Offset=0x10, Total Reset Value: 0x0000_0062, Write_enable=no

Bits	Access	Name	Descriptions	Reset
27:24	RW	pll1_out_div	PLL 1 VCO output clock post division ratio. It is strongly suggested not to modify this value. 0x0: 1/1 division; 0x1: 1/2 division; 0x2: 1/3 division; 0x3: 1/4 division;; 0xF: 1/16 division.	0x0
21:16	RW	pll1_ref_div	PLL 1 reference clock path division ratio. 0x0: 1/1 division; 0x1: 1/2 division; 0x2: 1/3 division; 0x3: 1/4 division;; 0x3F: 1/64 division.	0x0
12:0	RW	pll1_fb_div	PLL 1 feedback path clock division ratio. 0x0: 1/1 division; 0x1: 1/2 division; 0x2: 1/3 division; x3: 1/4 division;; 0xFFF: 1/4096 division; Other values: forbidden.	0x62

PLL1_CFG1

Offset=0x14, Total Reset Value: 0x0002_0031, Write_enable=no

Bits	Access	Name	Descriptions	Reset
19	RW	pll1_bypass	PLL 1 bypass mode enable control bit. When bypass mode, reference clock is directly bypass to PLL output clock. 0x0: disable PLL bypass mode; 0x1: enable PLL bypass mode.	0x0
18	RW	pll1_test	PLL 1 test mode enable control bit. 0x0: disable PLL test mode; 0x1: enable PLL test mode.	0x0

Bits	Access	Name	Descriptions	Reset
17	RW	pll1_ensat	PLL 1 saturation behavior enable control bit. Enabling saturation behavior can ensures analog circuits being biased at a optimal value. It's strongly recommended to set this bit to '1'. 0x0: disable PLL saturation behavior; 0x1: enable PLL saturation behavior.	0x1
16	RW	pll1_fasten	PLL 1 Fast locking circuit enable control bit. When this bit is set to '1', PLL output clock can reach lock state within far less time. It's strongly recommended to set this bit to '0'. 0x0: disable PLL fast locking circuit; 0x1: enable PLL fast locking circuit.	0x0
11:0	RW	pll1_bwadj	PLL 1 Bandwidth Adjustment parameter. The bandwdith can affect the quality of VCO output clock. The recommended value for this field is FBK_DIV/2. Valid values range from 0 to 0xFFFF.	0x31

PLL1_CTL

Offset=0x18, Total Reset Value: 0x0000_0034, Write_enable=yes

Bits	Access	Name	Descriptions	Reset
11:4	RW	pll1_lock_tim	The time required for PLL 1 to reach lock state. The time unit for this timer is a cycle of 32Khz clock, about 31.25us. N: N+1 time unit	0x3
2	RW	pll1_clk_oe	PLL1 VCO output clock enable control bit. 0x0: Disable VCO output clock; 0x1: Enable VCO output clock;	0x1
1	W1T	pll1_init	PLL 1 initialization trigger bit. 0x0: no operation; 0x1: start PLL initialization sequence.	0x0
0	W1T	pll1_pwrdown	PLL 1 go power-down mode trigger bit. 0x0: no operation; 0x1: put PLL 1 into power-down mode.	0x0

PLL1_STAT

Offset=0x1C, Total Reset Value: 0x0000_0021, Write_enable=no

Bits	Access	Name	Descriptions	Reset
5:4	RO	pll1_fsm_stat	PLL1 controlling FSM current status. 0x0: PLL FSM is in POWER_DOWN state; 0x1: PLL FSM is in RESET_PLL state; 0x3: PLL FSM is in WAIT_LOCK state; 0x2: PLL FSM is in PLL_READY state;	0x0
2	RO	pll1_fb_slip	This is the feedback clock path frequency/phase slip status. 0x0: PLL working OK; 0x1: There are frequency or phase slip on feedback clock.	0x0

Bits	Access	Name	Descriptions	Reset
1	RO	pll1_ref_slip	This is the reference clock path frequency/phase slip status. 0x0: PLL working OK; 0x1: There are frequency or phase slip on reference clock.	0x0
0	RO	pll1_lock	PLL 1 current lock status. 0x0: PLL not in lock state; 0x1: PLL in lock state.	0x0

PLL2_CFG0

Offset=0x20, Total Reset Value: 0x0007_0378, Write_enable=no

Bits	Access	Name	Descriptions	Reset
27:24	RW	pll2_out_div	PLL 2 VCO output clock post division ratio. It is strongly suggested not to modify this value. 0x0: 1/1 division; 0x1: 1/2 division; 0x2: 1/3 division; 0x3: 1/4 division;; 0xF: 1/16 division.	0x0
21:16	RW	pll2_ref_div	PLL 2 reference clock path division ratio. 0x0: 1/1 division; 0x1: 1/2 division; 0x2: 1/3 division; 0x3: 1/4 division;; 0x3F: 1/64 division.	0x7
12:0	RW	pll2_fb_div	PLL 2 feedback path clock division ratio. 0x0: 1/1 division; 0x1: 1/2 division; 0x2: 1/3 division; 0x3: 1/4 division;; 0xFFF: 1/4096 division; Other values: forbidden.	0x378

PLL2_CFG1

Offset=0x24, Total Reset Value: 0x0002_01bb, Write_enable=no

Bits	Access	Name	Descriptions	Reset
19	RW	pll2_bypass	PLL 2 bypass mode enable control bit. When bypass mode, reference clock is directly bypass to PLL output clock. 0x0: disable PLL bypass mode; 0x1: enable PLL bypass mode.	0x0
18	RW	pll2_test	PLL 2 test mode enable control bit. 0x0: disable PLL test mode; 0x1: enable PLL test mode.	0x0
17	RW	pll2_ensat	PLL 2 saturation behavior enable control bit. Enabling saturation behavior can ensures analog circuits being biased at a optimal value. It's strongly recommended to set this bit to '1'. 0x0: disable PLL saturation behavior; 0x1: enable PLL saturation behavior.	0x1
16	RW	pll2_fasten	PLL 2 Fast locking circuit enable control bit. When this bit is set to '1', PLL output clock can reach lock state within far less time. It's strongly recommended to set this bit to '0'. 0x0: disable PLL fast locking circuit; 0x1: enable PLL fast locking circuit.	0x0

Bits	Access	Name	Descriptions	Reset
11:0	RW	pll2_bwadj	PLL 2 Bandwidth Adjustment parameter. The bandwidth can affect the quality of VCO output clock. The recommended value for this field is FBK_DIV/2. Valid values range from 0 to 0xFFFF.	0x1bb

PLL2_CTL

Offset=0x28, Total Reset Value: 0x0000_0034, Write_enable=yes

Bits	Access	Name	Descriptions	Reset
11:4	RW	pll2_lock_tim	The time required for PLL 2 to reach lock state. The time unit for this timer is a cycle of 32Khz clock, about 31.25us. N: N+1 time unit	0x3
2	RW	pll2_clk_oe	PLL2 VCO output clock enable control bit. 0x0: Disable VCO output clock; 0x1: Enable VCO output clock;	0x1
1	W1T	pll2_init	PLL 2 initialization trigger bit. 0x0: no operation; 0x1: start PLL initialization sequence.	0x0
0	W1T	pll2_pwrdown	PLL 2 go power-down mode trigger bit. 0x0: no operation; 0x1: put PLL 0 into power-down mode.	0x0

PLL2_STAT

Offset=0x2C, Total Reset Value: 0x0000_0021, Write_enable=no

Bits	Access	Name	Descriptions	Reset
5:4	RO	pll2_fsm_stat	PLL 2 controlling FSM current status. 0x0: PLL FSM is in POWER_DOWN state; 0x1: PLL FSM is in RESET_PLL state; 0x3: PLL FSM is in WAIT_LOCK state; 0x2: PLL FSM is in PLL_READY state;	0x0
2	RO	pll2_fb_slip	This is the feedback clock path frequency/phase slip status. 0x0: PLL working OK; 0x1: There are frequency or phase slip on feedback clock.	0x0
1	RO	pll2_ref_slip	This is the reference clock path frequency/phase slip status. 0x0: PLL working OK; 0x1: There are frequency or phase slip on reference clock.	0x0
0	RO	pll2_lock	PLL 2 current lock status. 0x0: PLL not in lock state; 0x1: PLL in lock state.	0x0

PLL3_CFG0

Offset=0x30, Total Reset Value: 0x0102_018f, Write_enable=no

Bits	Access	Name	Descriptions	Reset
27:24	RW	pll3_out_div	PLL 3 VCO output clock post division ratio. It is strongly suggested not to modify this value. 0x0: 1/1 division; 0x1: 1/2 division; 0x2: 1/3 division; 0x3: 1/4 division;; 0xF: 1/16 division.	0x1
21:16	RW	pll3_ref_div	PLL 3 reference clock path division ratio. 0x0: 1/1 division; 0x1: 1/2 division; 0x2: 1/3 division; 0x3: 1/4 division;; 0x3F: 1/64 division.	0x2
12:0	RW	pll3_fb_div	PLL 3 feedback path clock division ratio. 0x0: 1/1 division; 0x1: 1/2 division; 0x2: 1/3 division; 0x3: 1/4 division;; 0xFFF: 1/4096 division; Other values: forbidden.	0x18f

PLL3_CFG1

Offset=0x34, Total Reset Value: 0x0002_00c7, Write_enable=no

Bits	Access	Name	Descriptions	Reset
19	RW	pll3_bypass	PLL 3 bypass mode enable control bit. When bypass mode, reference clock is directly bypass to PLL output clock. 0x0: disable PLL bypass mode; 0x1: enable PLL bypass mode.	0x0
18	RW	pll3_test	PLL 3 test mode enable control bit. 0x0: disable PLL test mode; 0x1: enable PLL test mode.	0x0
17	RW	pll3_ensat	PLL 3 saturation behavior enable control bit. Enabling saturation behavior can ensures analog circuits being biased at a optimal value. It's strongly recommended to set this bit to '1'. 0x0: disable PLL saturation behavior; 0x1: enable PLL saturation behavior.	0x1
16	RW	pll3_fasten	PLL 3 Fast locking circuit enable control bit. When this bit is set to '1', PLL output clock can reach lock state within far less time. It's strongly recommended to set this bit to '0'. 0x0: disable PLL fast locking circuit; 0x1: enable PLL fast locking circuit.	0x0
11:0	RW	pll3_bwadj	PLL 3 Bandwidth Adjustment parameter. The bandwidth can affect the quality of VCO output clock. The recommended value for this field is FBK_DIV/2. Valid values range from 0 to 0xFFFF.	0xc7

PLL3_CTL

Offset=0x38, Total Reset Value: 0x0000_0074, Write_enable=yes

Bits	Access	Name	Descriptions	Reset

Bits	Access	Name	Descriptions	Reset
11:4	RW	pll3_lock_tim	The time required for PLL 3 to reach lock state. The time unit for this timer is a cycle of 32Khz clock, about 31.25us. N: N+1 time unit	0x7
2	RW	pll3_clk_oe	PLL3 VCO output clock enable control bit. 0x0: Disable VCO output clock; 0x1: Enable VCO output clock;	0x1
1	W1T	pll3_init	PLL 3 initialization trigger bit. 0x0: no operation; 0x1: start PLL initialization sequence.	0x0
0	W1T	pll3_pwrdown	PLL 3 go power-down mode trigger bit. 0x0: no operation; 0x1: put PLL 1 into power-down mode.	0x0

PLL3_STAT

Offset=0x3C, Total Reset Value: 0x0000_0021, Write_enable=no

Bits	Access	Name	Descriptions	Reset
5:4	RO	pll3_fsm_stat	PLL3 controlling FSM current status. 0x0: PLL FSM is in POWER_DOWN state; 0x1: PLL FSM is in RESET_PLL state; 0x3: PLL FSM is in WAIT_LOCK state; 0x2: PLL FSM is in PLL_READY state;	0x0
2	RO	pll3_fb_slip	This is the feedback clock path frequency/phase slip status. 0x0: PLL working OK; 0x1: There are frequency or phase slip on feedback clock.	0x0
1	RO	pll3_ref_slip	This is the reference clock path frequency/phase slip status. 0x0: PLL working OK; 0x1: There are frequency or phase slip on reference clock.	0x0
0	RO	pll3_lock	PLL 3 current lock status. 0x0: PLL not in lock state; 0x1: PLL in lock state.	0x0

CPU0_CLK_CFG

Offset=0x0, Total Reset Value: 0x0000_e65f, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31	W1T	cpu0_core_clk_root_gdiv_upd	1: configure	0x0
17:15	RW	cpu_pclk_div	Cpu0, cpu1, tdi apb clock divider. Clock source pll0_ockl_div4, N: N-1 divider, 0: bypass	0x1
14	RW	tdi_pclk_en	1: tdi pclk enable	0x1
13	RW	cpu0_pclk_enable	1: cpu0 apb clock enable	0x1
12:10	RW	cpu0_pliclk_div	N: N+1 divider, 0: bypass	0x1

Bits	Access	Name	Descriptions	Reset
9	RW	cpu0_pliclk_enable	1: cpu0 pliclk enable	0x1
8:6	RW	cpu0_aclk_div	N: N+1 divider	0x1
4:1	RW	cpu0_core_clk_root_div	Cpu0 clock diverer parameter 0:1 /16 Pll0/2, 1:2/16 PLL0/2, 2:3/16 PLL0/2? 15: PLL0/2	0xf
0	RW	cpu0_core_clk_enable	cpu0 core clock enable	0x1

CPU1_CLK_CFG

Offset=0x4, Total Reset Value: 0x0009_900d, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31	W1T	cpu1_core_clk_div_upd	1: configure	0x0
19	RW	cpu1_pclk_enable	1: cpu1 apb clock enalbe	0x1
18:16	RW	cpu1_pliclk_div	N: cpu1_core_clk/(N+1)	0x1
15	RW	cpu1_pliclk_enable	1: cpu1 pliclk enable	0x1
14:12	RW	cpu1_aclk_div	N: cpu1_core_clk/(N+1)	0x1
5:3	RW	cpu1_core_clk_root_div	N: (N+1) divider.	0x1
2:1	RW	cpu1_core_clk_sel	0:select clock from pll1_ockl_div2, 1: select clock from pll3_ockl,2: select clock from pll0_ockl Default: 1	0x2
0	RW	cpu1_core_clk_enable	1: cpu1 core clock enable	0x1

ai_CLK_CFG

Offset=0x8, Total Reset Value: 0x0000_043d, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31	W1T	ai_clk_div_upd	1: configure	0x0
10	RW	ai_aclk_enable	1: ai axi clock enable	0x1
5:3	RW	ai_clk_cdiv	N: (N+1) divider.	0x7
2	RW	ai_clk_sel	Ai core clock selection. 0:select clock from pll1_ockl_div2, 1: select clock from pll3_ockl_div2	0x1
0	RW	ai_clk_enable	1: ai core clock enable	0x1

vpu_CLK_CFG

Offset=0xc, Total Reset Value: 0x0000_0c7f, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31	W1T	vpu_clk_gdiv_upd	1: configure	0x0
14:11	RW	vpu_cfgclk_root_div	VPU cfgclk divider. N: (N+1) divider.	0x1

Bits	Access	Name	Descriptions	Reset
10	RW	vpu_cfgclk_enable	1:cpu cfgclk enable	0x1
9:6	RW	vpu_aclk_root_div	N: (N+1) divider.	0x1
5	RW	vpu_aclk_enable	1: vpu axi clock enable	0x1
4:1	RW	vpu_clk_div	N: (N+1)/16 divid of source clock	0xf
0	RW	vpu_clk_en	1: vpu core clock enable	0x1

pmu_CLK_CFG

Offset=0x10, Total Reset Value: 0x0000_0001, Write_enable=no

Bits	Access	Name	Descriptions	Reset
0	RW	pmu_clk_en	1: pmu clk enable	0x1

hs_CLKen_CFG

Offset=0x18, Total Reset Value: 0x7f7f_ffff, Write_enable=no

Bits	Access	Name	Descriptions	Reset
30	RW	ssi2_aclk_enable	Ssi2 axi clock enable. 1'b1: clock enable, 1'b0 clock disable	0x1
29	RW	ssi1_aclk_enable	Ssi1 axi clock enable. 1'b1: clock enable, 1'b0 clock disable	0x1
28	RW	qspi_aclk_enable	Qspi axi clock enable. 1'b1: clock enable, 1'b0 clock disable	0x1
27	RW	ssi0_aclk_enable	OSPI axi clock enable. 1'b1: clock enable, 1'b0 clock disable	0x1
26	RW	ssi2_clk_enable	ssi2 core clock enable. 1: clock enable, 1'b0 clock disable	0x1
25	RW	ssi1_clk_enable	ssi1 core clock enable. 1'b1: clock enable, 1'b0 clock disable	0x1
24	RW	ssi0_clk_enable	ospi core clock enable. 1'b1: clock enable, 1'b0 clock disable	0x1
23	RW	usb_refclk_sel	1: 50M, 0: 24M	0x0
22	RW	usb1_refclk_enable	Usb reference clock enable . 1'b1: clock enable, 1'b0 clock disable	0x1
21	RW	usb0_refclk_enable	Usb reference clock enable .1'b1: clock enable, 1'b0 clock disable	0x1
20	RW	sd1_tmclk_enable	Sd1 tmclk enable. 1'b1: clock enable, 1'b0 clock disable	0x1
19	RW	sd1_cclk_enable	Sd1 card clock enable. 1'b1: clock enable, 1'b0 clock disable	0x1
18	RW	sd1_bclk_enable	Sd1 base clock enable. 1'b1: clock enable, 1'b0 clock disable	0x1
17	RW	sd1_aclk_enable	Sd1 axi clock enable. 1'b1: clock enable, 1'b0 clock disable	0x1
16	RW	sd0_tmclk_enable	Sd0 tmclk enable. 1'b1: clock enable, 1'b0 clock disable	0x1
15	RW	sd0_cclk_enable	Sd0 card clock enable. 1'b1: clock enable, 1'b0 clock disable	0x1
14	RW	sd0_bclk_enable	Sd0 base clock enable. 1'b1: clock enable, 1'b0 clock disable	0x1
13	RW	sd0_aclk_enable	Sd0 axi clock enable. 1'b1: clock enable, 1'b0 clock disable	0x1
12	RW	sd_tmclk_enable	Sd0 and sd1 tmclk enable. 1'b1: clock enable, 1'b0 clock disable	0x1

Bits	Access	Name	Descriptions	Reset
11	RW	sd_cclk_enable	Sd0 and sd1 card clock enable. 1'b1: clock enable, 1'b0 clock disable	0x1
10	RW	sd_bclk_enable	Sd0 and sd1 base clock enable. 1'b1: clock enable, 1'b0 clock disable	0x1
9	RW	sd_aclk_enable	Sd0 and sd1 axi clock enable. 1'b1: clock enable, 1'b0 clock disable	0x1
8	RW	ssi2_hclk_enable	Ssi2 ahb clock enable. 1'b1: clock enable, 1'b0 clock disable	0x1
7	RW	ssi1_hclk_enable	Ssi1 ahb clock enable. 1'b1: clock enable, 1'b0 clock disable	0x1
6	RW	ssi0_hclk_enable	Ssi0 ahb clock enable. 1'b1: clock enable, 1'b0 clock disable	0x1
5	RW	usb1_hclk_enable	Usb1 ahb clock enable,. 1'b1: clock enable, 1'b0 clock disable	0x1
4	RW	usb0_hclk_enable	Usb0 ahb clock enable,.1'b1: clock enable, 1'b0 clock disable	0x1
3	RW	sd1_hclk_enable	sd1 ahb clock enable,.1'b1: clock enable, 1'b0 clock disable	0x1
2	RW	sd0_hclk_enable	sd0 ahb clock enable,.1'b1: clock enable, 1'b0 clock disable	0x1
1	RW	hs_hclk_high_enable	hi_sys hclk_high enalb.e 1'b1: clock enable, 1'b0 clock disable	0x1
0	RW	hs_hclk_enable	hi_sys ahb clock enable. 1'b1: clock enable, 1'b0 clock disable	0x1

hs_SDCLK_CFG

Offset=0x1c, Total Reset Value: 0x000b_9248, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31	W1T	sd_aclk_root_cdiv_upd	1: configure	0x0
19:15	RW	sd_tmclk_div	Sd tmclk divider. N: (N+1) divider.	0x17
14:12	RW	sd_cclk_div	Sd card clock divider. N: (N+1) divider. Sd0、 sd1 cclk is divided from this clock.	0x1
11:9	RW	sd_bclk_div	Sd base cock divider. N: (N+1) divider.	0x1
8:6	RW	sd_aclk_root_div	Sd axi clock divider. N: (N+1) divider.	0x1
5:3	RW	hs_hclk_div	hi_sys ahb clock divider. N: (N+1) divider.	0x1
2:0	RW	hs_hclk_high_root_div	hi_sys hs_hclk_high divider. N: (N+1) divider.	0x0

hs_SPI_CFG

Offset=0x20, Total Reset Value: 0x0003_9048, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31	W1T	ssi0_aclk_root_cdiv_upd	1: configure	0x0
18	RW	ssi0_clk_sel	ssi0_clk divider source select. 0: select pll0_ockl_div2, 1: select pll2_ockl_div4	0x0
17:15	RW	usb_refclk_50m_div	usb reference clock divider. N: N+1 divide	0x7
14:12	RW	qspi_aclk_root_div	qspi axi clock divider. N: N+1 divide	0x1

Bits	Access	Name	Descriptions	Reset
11:9	RW	ssi0_aclk_div	ssi0 axi clock divider. N: N+1 divide	0x0
8:6	RW	ssi2_clk_div	ssi2 core clock divider. N: N+1 divide	0x1
5:3	RW	ssi1_clk_div	ssi1 core clock divider. N: N+1 divide	0x1

LS_CLKEN_CFG0

Offset=0x24, Total Reset Value: 0xffff_ffff, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31	RW	pdm_clk_enable	pdm clock enable. 1'b1: clock enable, 1'b0 clock disable	0x1
30	RW	codec_mclk_dac_enable	1'b1: clock enable, 1'b0 clock disable	0x1
29	RW	codec_mclk_adc_enable	1'b1: clock enable, 1'b0 clock disable	0x1
28	RW	aud_dev_clk_enable	1: audio clock enable.	0x1
27	RW	gpio_dbclk_enable	1: gpio debounce clock enable	0x1
26	RW	adc_clk_enable	1: adc clock enable	0x1
25	RW	i2c4_clk_enable	1: i2c4 clock enable	0x1
24	RW	i2c3_clk_enable	1: i2c3 clock enable	0x1
23	RW	i2c2_clk_enable	1: i2c2 clock enable	0x1
22	RW	i2c1_clk_enable	1: i2c1 clock enable	0x1
21	RW	i2c0_clk_enable	1: i2c0 clock enable	0x1
20	RW	uart4_clk_enable	1: uart4 clock enable	0x1
19	RW	uart3_clk_enable	1: uart3 clock enable	0x1
18	RW	uart2_clk_enable	1: uart2 clock enable	0x1
17	RW	uart1_clk_enable	1: uart1 clock enable	0x1
16	RW	uart0_clk_enable	1: uart0 clock enable	0x1
15	RW	adc_pclk_enable	1: adc apb clock enable	0x1
14	RW	codec_pclk_enable	1: codec apb clock enable	0x1
13	RW	aud_dev_pclk_enable	1: audio apb clock enable	0x1
12	RW	pwm_pclk_enable	1: pwm apb clock enable	0x1
11	RW	gpio_pclk_enable	1: gpio apb clock enable	0x1
10	RW	i2c4_pclk_enable	1: i2c4 apb clock enable	0x1
9	RW	i2c3_pclk_enable	1: i2c3 apb clock enable	0x1
8	RW	i2c2_pclk_enable	1: i2c2 apb clock enable	0x1
7	RW	i2c1_pclk_enable	1: i2c1 apb clock enable	0x1
6	RW	i2c0_pclk_enable	1: i2c0 apb clock enable	0x1
5	RW	uart4_pclk_enable	1: uart4 apb clock enable	0x1

Bits	Access	Name	Descriptions	Reset
4	RW	uart3_pclk_enable	1: uart3 apb clock enable	0x1
3	RW	uart2_pclk_enable	1: uart2 apb clock enable	0x1
2	RW	uart1_pclk_enable	1: uart1 apb clock enable	0x1
1	RW	uart0_pclk_enable	1: uart0 apb clock enable	0x1
0	RW	ls_pclk_enable	1: all low sys apb clock enable	0x1

LS_CLKEN_CFG1

Offset=0x28, Total Reset Value: 0x0000_00ff, Write_enable=no

Bits	Access	Name	Descriptions	Reset
7	RW	jamlink3_pclk_enable	jamlink3 apb clock enable. 1'b1: clock enable, 1'b0 clock disable	0x1
6	RW	jamlink2_pclk_enable	jamlink2 apb clock enable. 1'b1: clock enable, 1'b0 clock disable	0x1
5	RW	jamlink1_pclk_enable	jamlink1 apb clock enable. 1'b1: clock enable, 1'b0 clock disable	0x1
4	RW	jamlink0_pclk_enable	jamlink0 apb clock enable. 1'b1: clock enable, 1'b0 clock disable	0x1
3	RW	jamlink3_CO_enable	jamlink3 CO enable. 1'b1: clock enable, 1'b0 clock disable	0x1
2	RW	jamlink2_CO_enable	jamlink2 CO enable. 1'b1: clock enable, 1'b0 clock disable	0x1
1	RW	jamlink1_CO_enable	jamlink1 CO enable. 1'b1: clock enable, 1'b0 clock disable	0x1
0	RW	jamlink0_CO_enable	jamlink0 CO enable. 1'b1: clock enable, 1'b0 clock disable	0x1

UART_I2C_CLKDIV_CFG

Offset=0x2c, Total Reset Value: 0x1b6d_9249, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31	W1T	i2c0_clk_cdiv_upd	1: configure	0x0
29:27	RW	i2c4_clk_div	i2c4 core clock divider. N: N+1 divide	0x3
26:24	RW	i2c3_clk_div	i2c3 core clock divider. N: N+1 divide	0x3
23:21	RW	i2c2_clk_div	i2c2 core clock divider. N: N+1 divide	0x3
20:18	RW	i2c1_clk_div	i2c1 core clock divider. N: N+1 divide	0x3
17:15	RW	i2c0_clk_div	i2c0 core clock divider. N: N+1 divide	0x3
14:12	RW	uart4_sclk_div	uart4 core clock divider. N: N+1 divide	0x1
11:9	RW	uart3_sclk_div	uart3 core clock divider. N: N+1 divide	0x1
8:6	RW	uart2_sclk_div	uart2 core clock divider. N: N+1 divide	0x1
5:3	RW	uart1_sclk_div	uart1 core clock divider. N: N+1 divide	0x1

Bits	Access	Name	Descriptions	Reset
2:0	RW	uart0_sclk_div	uart0 core clock divider. N: N+1 divide	0x1

LS_CLKDIV_CFG

Offset=0x30, Total Reset Value: 0x01dd_a0c3, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31	W1T	jamlink_CO_cdiv_upd	1: configure	0x0
30:23	RW	jamlink_CO_div	jamlink clock divider. N: 2(N+1) divide	0x3
22:13	RW	gpio_dbclk_div	gpio debounce clock divider.N: N+1 divide	0x2ed
12:3	RW	adc_clk_div	adc clock divider.N: N+1 divide	0x18
2:0	RW	ls_pclk_div	lo_sys apb clock divider. N: N+1 divide	0x3

AUDIO_CLKDIV_CFG

Offset=0x34, Total Reset Value: 0x01b9_f424, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31	W1T	aud_dev_clk_fdiv_upd	1: configure	0x0
30:16	RW	audio_clk_step	audio clock stup. Forexample, audio_clk: 25644.1K, <i>source clock: 400M, 400M/(25644.1K)</i> can be simplified to : 15625/441. sum is set to : 15625, step is set to 441	0x1b9
15:0	RW	audio_clk_sum	audio clock sum	0xf424

CODEC_ADC_MCLKDIV_CFG

Offset=0x38, Total Reset Value: 0x006e_7d09, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31	W1T	codec_mclk_adc_fdiv_upd	1: configure	0x0
26:14	RW	codec_mclk_adc_step	codec clock stup. Forexample, audio_clk: 25644.1K, <i>source clock: 400M, 400M/(25644.1K)</i> can be simplified to : 15625/441. sum is set to : 15625, step is set to 441	0x1b9
13:0	RW	codec_mclk_adc_sum	codec clock sum	0x3d09

CODEC_DAC_MCLKDIV_CFG

Offset=0x3c, Total Reset Value: 0x006e_7d09, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31	W1T	codec_mclk_dac_fdiv_upd	1: configure	0x0
26:14	RW	codec_mclk_dac_step	codec clock stup. Forexample, audio_clk: 25644.1K, <i>source clock: 400M, 400M/(25644.1K)</i> can be simplified to : 15625/441. sum is set to : 15625, step is set to 441	0x1b9

Bits	Access	Name	Descriptions	Reset
13:0	RW	codec_mclk_dac_sum	codec clock sum	0x3d09

PDM_CLKDIV_CFG0

Offset=0x40, Total Reset Value: 0x0000_f424, Write_enable=no

Bits	Access	Name	Descriptions	Reset
16:0	RW	pdm_clk_sum	Pdm clock sum	0xf424

PDM_CLKDIV_CFG1

Offset=0x44, Total Reset Value: 0x0000_01b9, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31	W1T	pdm_clk_fdiv_upd	1: configure	0x0
15:0	RW	pdm_clk_step	Pdm clock step	0x1b9

SYSCTL_CLKEN_CFG

Offset=0x50, Total Reset Value: 0x003f_fffe, Write_enable=no

Bits	Access	Name	Descriptions	Reset
21	RW	hdi_clk_enable	hdi clock enable. 1'b1: clock enable, 1'b0 clock disable	0x1
20	RW	iomux_pclk_enable	iomux apb clock enable. 1'b1: clock enable, 1'b0 clock disable	0x1
19	RW	stc_clk_en	time stamp clock enable . 1'b1: clock enable, 1'b0 clock disable	0x1
18	RW	timer5_clk_enable	Timer 5 core clock enable. 1'b1: clock enable, 1'b0 clock disable	0x1
17	RW	timer4_clk_enable	Timer 4 core clock enable. 1'b1: clock enable, 1'b0 clock disable	0x1
16	RW	timer3_clk_enable	Timer 3 core clock enable. 1'b1: clock enable, 1'b0 clock disable	0x1
15	RW	timer2_clk_enable	Timer 2 core clock enable. 1'b1: clock enable, 1'b0 clock disable	0x1
14	RW	timer1_clk_enable	Timer 1 core clock enable. 1'b1: clock enable, 1'b0 clock disable	0x1
13	RW	timer0_clk_enable	Timer 0 core clock enable. 1'b1: clock enable, 1'b0 clock disable	0x1
12	RW	timer5_clk_sel	1'b1: select pulse in, 0: select pll div	0x1
11	RW	timer4_clk_sel	1'b1: select pulse in, 0: select pll div	0x1
10	RW	timer3_clk_sel	1'b1: select pulse in, 0: select pll div	0x1
9	RW	timer2_clk_sel	1'b1: select pulse in, 0: select pll div	0x1
8	RW	timer1_clk_sel	1'b1: select pulse in, 0: select pll div	0x1
7	RW	timer0_clk_sel	1'b1: select pulse in, 0: select pll div	0x1
6	RW	wdt1_clk_enable	wdt1 clock enable. 1'b1: clock enable, 1'b0 clock disable	0x1
5	RW	wdt0_clk_enable	wdt0 clock enable. 1'b1: clock enable, 1'b0 clock disable	0x1
4	RW	mailbox_pclk_enable	mailbox apb clock enable.	0x1

Bits	Access	Name	Descriptions	Reset
3	RW	timer_pclk_enable	timer apb clock enable	0x1
2	RW	wdt1_pclk_enable	wdt1 apb clock enable	0x1
1	RW	wdt0_pclk_enable	wdt0 apb clock enable	0x1

TIMER_CLK_CFG

Offset=0x54, Total Reset Value: 0x0003_ffff, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31	W1T	timer_clk0_cdiv_upd	1: configure	0x0
17:15	RW	timer_clk5_div	Timer 5 core clock divider. N: N+1 divide	0x7
14:12	RW	timer_clk4_div	Timer 4 core clock divider. N: N+1 divide	0x7
11:9	RW	timer_clk3_div	Timer 3 core clock divider. N: N+1 divide	0x7
8:6	RW	timer_clk2_div	Timer 2 core clock divider. N: N+1 divide	0x7
5:3	RW	timer_clk1_div	Timer 1 core clock divider. N: N+1 divide	0x7
2:0	RW	timer_clk0_div	Timer 0 core clock divider. N: N+1 divide	0x7

SYSCTL_CLK_DIV_CFG

Offset=0x58, Total Reset Value: 0x1efa_bef9, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31	W1T	sys_ctl_pclk_cdiv_upd	1: configure	0x0
30:28	RW	hdi_clk_div	hdi core clock	0x1
27:20	RW	ts_clk_div	temp sensor clock divider.	0xef
19:15	RW	stc_clk_div	time stamp clock divider.	0x15
14:9	RW	wdt1_clk_div	wdt1 clock divider.	0x1f
8:3	RW	wdt0_clk_div	wdt0 clock divider.	0x1f

shrm_CLK_CFG

Offset=0x5c, Total Reset Value: 0x000c_7ffd, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31	W1T	shrm_pclk_root_div_upd	1: configure	0x0
20:18	RW	shrm_pclk_root_div	apb clock divider.	0x3
14	RW	sram_clk_sel	0: select pll0, 1: select pll3	0x1
13	RW	shrm_axim_bridge_clk_en	3 axi master bridge	0x1
12	RW	shrm_axim_clk_en	shrm axi master clock enable	0x1
11	RW	shrm_axis_clk_en	shrm axi slave clock enable	0x1

Bits	Access	Name	Descriptions	Reset
10	RW	sram_aclk_enable	shrm sram clock	0x1
9	RW	nonai2d_aclk_enable	nonai2d axi clock enable	0x1
8	RW	nonai2d_pclk_enable	nonai 2d apb clock enable	0x1
7	RW	decompress_aclk_enable	Decompress axi clock enable	0x1
6	RW	decompress_pclk_enable	Decompress apb clock enable	0x1
5	RW	gsdma_aclk_enable	Gsdma axi clock enable	0x1
4	RW	gsdma_pclk_enable	Gsdma apb clock enable	0x1
3	RW	peri_dma_aclk_enable	Pdma axi clock enable	0x1
2	RW	peri_dma_pclk_enable	Pdma apb clock enable	0x1
0	RW	shrm_pclk_enable	apb clock enable	0x1

DDR_CLK_CFG

Offset=0x60, Total Reset Value: 0x0000_43fd, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31	W1T	ddrc_core_clk_cdiv_upd	1: configure	0x0
17:14	RW	ddrc_pclk_div	ddrc apb clock divider. N:N+1 divider	0x1
13:10	RW	ddrc_core_clk_div	ddrc core clock divider. N:N+1 divider	0x0
9	RW	ddrc_pclk_enable	ddrc apb clock enable	0x1
8	RW	ddr_bypasspclk_enable	clock used to bypass ddr phy pll	0x1
7	RW	ddrc_axi4clk_enable	ddrc port4 clock enable	0x1
6	RW	ddrc_axi3clk_enable	ddrc port3 clock enable	0x1
5	RW	ddrc_axi2clk_enable	ddrc port2 clock enable	0x1
4	RW	ddrc_axi1clk_enable	ddrc port1 clock enable	0x1
3	RW	ddrc_axi0clk_enable	ddrc port0 clock enable	0x1
2	RW	ddrc_clk_enable	ddrc core clock enable	0x1
1:0	RW	ddrc_clk_sel	2'b0:select pll0 div2, 2'b1: select pll0 div3,2鈎2,2鈎3: select pll2_div4	0x1

ISP_CLKEN_CFG

Offset=0x64, Total Reset Value: 0x000f_ffff, Write_enable=no

Bits	Access	Name	Descriptions	Reset
19	RW	isp_dw_aclk_enable	isp dewarp axi clock enable	0x1
18	RW	isp_dw_hclk_enable	isp dewarp ahb clock enable	0x1
17	RW	isp_mcm_clk_enable	isp mcm clock enable	0x1

Bits	Access	Name	Descriptions	Reset
16	RW	ispvse_clk_enable	isp scalar clock enable	0x1
15	RW	ispdwe_clk_enable	isp dewarp clock enable	0x1
14	RW	isp_aclk_enable	isp axi clock enable	0x1
13	RW	isp_hclk_enable	isp ahb clock enable	0x1
12	RW	isp_clk_enable	isp core clock enable	0x1
11	RW	isp_csi2_pclk_enable	csi2 apb clock enable	0x1
10	RW	isp_csi1_pclk_enable	csi1 apb clock enable	0x1
9	RW	isp_csi0_pclk_enable	csi0 apb clock enable	0x1
8	RW	csi2_pixelclk_enable	csi2 pixelclk enable	0x1
7	RW	csi1_pixelclk_enable	csi1 pixelclk enable	0x1
6	RW	csi0_pixelclk_enable	csi0 pixelclk enable	0x1
5	RW	isp_dphy_pclk_enable	isp dphy apb clock enable	0x1
4	RW	isp_cfg2_clk_enable	csi2 phy cfgclk enable	0x1
3	RW	isp_cfg1_clk_enable	csi1 phy cfgclk enable	0x1
2	RW	isp_cfg0_clk_enable	csi0 phy cfgclk enable	0x1
1	RW	isp_pclk_enable	isp abp clock enable	0x1
0	RW	csiphy_cfgclk_enable	csi phy cfgclk enable	0x1

ISP_CLKDIV_CFG

Offset=0x68, Total Reset Value: 0x0082_0835, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31	W1T	isp_cfg_clk_cdiv_upd	1: configure	0x0
29	RW	isp_clk_sel	0:with mcm, 1: without mcm	0x0
28:26	RW	isp_clk_div	isp clock divider. N: N+1 divide	0x0
25:23	RW	isp_pclk_div	isp abp clock divider. N: N+1 divide	0x1
22:17	RW	csi2_pixel_clk_div	csi2 pixel clock divider. N: N+1 divide	0x1
16:11	RW	csi1_pixel_clk_div	csi1 pixel clock divider. N: N+1 divide	0x1
10:5	RW	csi0_pixel_clk_div	csi0 pixel clock divider. N: N+1 divide	0x1
4:0	RW	isp_cfg_clk_div	isp cfgclk divider. N: N+1 divide	0x15

MCLK_CFG

Offset=0x6c, Total Reset Value: 0x003a_74ef, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31	W1T	mclk0_cdiv_upd	1: configure	0x0

Bits	Access	Name	Descriptions	Reset
23:19	RW	mclk2_div	mclk2 clock divider. N: N+1 divider	0x7
18:17	RW	mclk2_sel	0: pll1_ockl_div3,1: pll1_ockl_div4,2: pll0_ockl_div4	0x1
16:12	RW	mclk1_div	mclk1 clock divider. N: N+1 divider	0x7
11:10	RW	mclk1_sel	0: pll1_ockl_div3,1: pll1_ockl_div4,2: pll0_ockl_div4	0x1
9:5	RW	mclk0_div	mclk0 clock divider. N: N+1 divider	0x7
4:3	RW	mclk0_sel	0: pll1_ockl_div3,1: pll1_ockl_div4,2: pll0_ockl_div4	0x1
2	RW	mclk2_enable	mclk2_enable	0x1
1	RW	mclk1_enable	mclk1_enable	0x1
0	RW	mclk0_enable	mclk0_enable	0x1

DPU_CLK_CFG

Offset=0x70, Total Reset Value: 0x0000_01ff, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31	W1T	dpu_clk_div_upd	1: configure	0x0
9:7	RW	dpu_pclk_div	dpu apb clock divider. N: N+1 divide	0x3
6:3	RW	dpu_clk_div	dpu core clock divider. 0~15 : 1/16 ~16/16 divide	0xf
2	RW	dpu_pclk_enable	dpu apb clock enable	0x1
1	RW	dpu_aclk_enable	dpu axi clock enable	0x1
0	RW	dpu_clk_enable	dpu core clock enable	0x1

VO_CLK_CFG

Offset=0x74, Total Reset Value: 0x0000_1fff, Write_enable=no

Bits	Access	Name	Descriptions	Reset
12	RW	aclk_gpu_en	gpu axi clock enable	0x1
11	RW	aclk_vo_en	display vo ahb clock enable	0x1
10	RW	hclk_gpu_en	display gpu ahb clock enable	0x1
9	RW	pclk_dsi_phy_en	display dsi phy apb clock enable	0x1
8	RW	pclk_dsi_enable	display dsi apb clock enable	0x1
7	RW	pclk_vo_enable	display vo apb clock enable	0x1
6	RW	clk_gpu_enable	display gpu clock enable	0x1
5	RW	disp_clkext_enable	display clkext clock enable	0x1
4	RW	disp_cfgclk_enable	display cfgclk clock enable	0x1
3	RW	disp_refclk_enable	display refclk clock enable	0x1

Bits	Access	Name	Descriptions	Reset
2	RW	dpipclk_enable	display dpipclk clock enable	0x1
1	RW	disp_aclk_enable	display axi clock enable	0x1
0	RW	disp_hclk_enable	display ahb clock enable	0x1

DISP_CLK_DIV

Offset=0x78, Total Reset Value: 0x0000_a83b, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31	W1T	disp_hclk_root_cdiv_upd	1: configure	0x0
23:20	RW	disp_clk_gpu_div	gpu clock divider. N: N+1 divide	0x0
19:16	RW	disp_clkext_div	clkext divider. N: N+1 divide	0x0
15:11	RW	disp_cfgclk_div	cfgclk divider. N: N+1 divide	0x15
10:3	RW	dpipclk_div	dpipclk divider. N: N+1 divide	0x7
2:0	RW	disp_hclk_root_div	ahb clock divider. N: N+1 divide	0x3

SEC_CLK_DIV

Offset=0x80, Total Reset Value: 0x0000_02f3, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31	W1T	sec_pclk_div_upd	1: configure	0x0
13:11	RW	sec_aclk_div	sec axi clock divider. N: N+1 divide	0x0
10:6	RW	sec_fixclk_div	fixclk divider. N: N+1 divide	0xb
5	RW	sec_fixclk_enable	fixclk enable	0x1
4	RW	sec_aclk_enable	axi clock enable	0x1
3:1	RW	sec_pclk_div	apb divider. N: N+1 divide	0x1
0	RW	sec_pclk_enable	apb clock enable	0x1

USB_TEST_CLK_DIV

Offset=0x100, Total Reset Value: 0x0000_0039, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31	W1T	usb_clk480_cdiv_upd	1: configure	0x0
6:4	RW	usb_clk100_div	test mode 100m clock enable	0x3
3:1	RW	usb_clk480_div	test mode 480m clock divider	0x4
0	RW	usb_test_clk_enable	test clock enable	0x1

DPHY_TEST_CLK_DIV

Offset=0x104, Total Reset Value: 0x0000_0009, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31	W1T	dphy_test_clk_cdiv_upd	1: configure	0x0
4:1	RW	dphy_test_clk_div	isp dphy dft mode clock divider. N: N+1 divider	0x4
0	RW	dphy_test_clk_enable	isp dphy dft mode clock enable	0x1

2.2.6 Application note

- It is not recommended to close sub-module bus interface clock regardless of unfinished bus transaction.
- It is recommend to check sub-module status before close sub-module configuration clock.
- It is recommend to negotiate through NOC LPI interface to ensure current bus interface is at idle state and then close sub-module configuration clock.
- Three divided-frequency of i2c clock is not supported.
- Cpu1_core_clk , ai_clk , ddrc_core_clk , vpu_clk support odd divider clock, other clock only support even divider clock.

2.3 Power Management and Low-Power Mode

2.3.1 overview

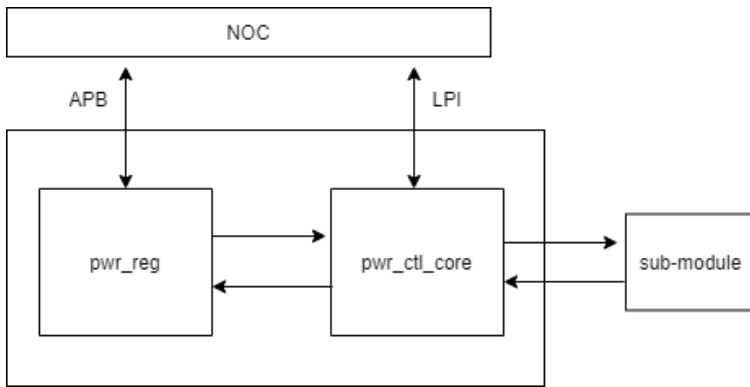
Power control module (PWR) is used for power control of SOC and sub-module. There are five power mode : Power on, sleep0, sleep1, standby, powerdown. Power on: The normal function mode. Sleep0 : all power domain except for PD_Normal power domain are powered off. PLLs remain power on. Sleep1 : all power domain except for PD_Normal power domain are powered off. PLLs are powered off. standby: PD_Normal is also power off. All power domain except PMU are powered off Power down: the whole soc are power off There are three kinds of wakeup source : GPIO, PMU and timer The power domain are divided as :

Table 2-3-1 power domain

power domain	module	power mode
PD_PMU	pmu subsystem	always on
PD_AI	ai subsystem	power gating
PD_Display	display subsystem 2.5D engine	power gating
PD_VPU	video subsystem	power gating
PD_3D	3D structure subsystem	power gating
PD_CPU1	cpu1 subsystem	power gating
PD_DDR	DDR subsystem	PMIC
PD_Normal	other module	PMIC

2.3.2 block diagram

Figure 2-3-1 power control diagram



2.3.3 function description

pwr_reg : CPU can control power control module through apb interface. **pwr_ctl_core**: power gating unit is used to control soc and sub-module power gating. **xxx_PWR_LPI_CTL.xxx_pwr_mode** is used to select power mode. The default value is 0. When **xxx_pwr_mode=0** power will be gated. When **xxx_pwr_mode=1** power will remain on while clock is gated. Before module enters low power mode, bus operation should be finished. This can be obtained through NOC LPI handshake. NOC LPI low power handshake can be triggered by **xxx_PWR_LPI_CTL.xxx_pwr_off_req** automatically or by **xxx_PWR_LPI_CTL.xxx_lpi_go_idle** explicitly. SOC low power mode can be trigger automatically when all sub-modules enter low power mode or by **SOC_SLEEP_CTL.core_sleep_req** explicitly. **SOC_SLEEP_CTL.auto_slp_en** is used to control automatic low power mode. When **SOC_SLEEP_CTL.auto_slp_en = 1**, SOC will enter low power mode when all modules enter low power mode. **SOC_SLEEP_CTL.core_sleep_mode** is used to control whether to power down PLL during SLEEP mode. The default value is 0 which means PLL will be powerd off.

2.3.3.1 sub-module power control

sub-module low power flow:

- Configure **xxx_PWR_LPI_CTL.xxx_pwr_mode** to select power off mode or clock off mode. The default mode is power off.
- Configure **xxx_PWR_LPI_CTL.xxx_lpi_go_idle** to start NOC LPI handshake. This step is optional because it will be trigger by next step.
- Configure **xxx_PWR_LPI_CTL.xxx_pwr_off_req** to start low power sequence.
- Module power status can be obtained through **xxx_LPI_STAT**.

2.3.3.2 CPU power control

Because the particularity of CPUs, low power sequence is little different with normal operation.

CPUx_PWR_CTL.cpux_auto_pwr_en is used to select power control mode : automatic and non-automatic mode.

Automatic power off mode

When **CPUx_PWR_CTL.cpux_auto_pwr_en = 1**, automatical power off mode is used.

- CPU enter WFI state.
- WFI counter start to work.
- Power off sequence start when WFI counter is full.

When **CPUx_PWR_CTL.cpux_auto_pwr_en = 0**, non-automatical power off mode is used. The power off sequence is the same as [2.3.3.1 sub-module power control](#)

2.3.3.3 DDR power control

K230 DDR has 6 AXI low power interface, one for ddrc core and another for 5 ddrc axi ports. These 6 AXI low power interface can be controlled independly. When DDR is needed to enter low power state, NOC LPI should be processed at first.

power sequence

The following power sequence must be met:

- power on : Supply normal domain power at first, and then DDR power.
- power off: Remove DDR power at first, an then normal domain power.

MCTL_PWR_LPI_CTL.mctl_hw_pwr_mode is used to select DDR power mode. When MCTL_PWR_LPI_CTL.mctl_hw_pwr_mode equals 1 , hardware mode is selected. Otherwise software mode is selected.

enter low power mode using hardware mode

- Set MCTL_PWR_LPI_CTL.mctl_go_idle to 1. Then DDR will enter self-refresh mode through hardware operation automatically. This step is optional. This step can be triggered by next step.
- Set MCTL_PWR_LPI_CTL.mctl_pwr_off_req to 1 to start low power sequence.
- CPU check DDR status until self-refresh state is obtained. Change Pstate of DDR to LP3.
- CPU set PwrOkIn to 0.
- Hardware will process NOC, clock, reset automatically.
- CPU control PMIC to cutoff VDD and VAA of DDR.

exit low power mode using hardware mode

- Supply VDD and VAA.
- Hareware will process NOC, clock, reset automatically.
- CPU check MCTL_LPI_STAT.mctl_lpi_work untill its value equals 1.
- CPU start to configure DDRC. When configuration of DDRC is finished, configure MCTL_PWR_LPI_CTL.mctl_ddrc_init_done to 1.
- Software begin to configure DDR PHY. When configuration of DDR PHY is finished, set MCTL_LPI_STAT.mctl_dfi_init_done to 1.

enter low power mode using software mode

- Power remove
 - Write '0' to PCTRL_n.port_en. Blocks AXI ports from taking anymore transactions
 - Poll PSTAT.rd_port_busy_n = 0, Poll PSTAT.wr_port_busy_n = 0. Waits unit all AXI ports are idle.
 - Write '0' to SBRCTL.scrub_en. Disables SBR, only required if SBR instantiated.
 - Poll SBRSTAT.scrub_busy = 0. Indicates that there are no outstanding SBR read commands, only required if SBR instantiated.
 - Program DFILPCF0.dfi_lp_en_sr = 0.Poll DFISTAT.dfi_lp_ack = 0.
 - Wait until STAT.operating_mode[1:0]!=11. indicating that the DDR controller is not in self-refresh mode.
 - Write '1' to PWRCTL.selfref_sw. Causes system to move to Self Refresh state. For LPDDR4, PWRCTL stay_in_selfref must be set to '0' to enter Self Refresh Power dow.
 - Poll STAT.selfref_type= 2'b10. Poll STAT.selfref_state = 2'b10 (LPDDR4 only). Waits until Self Refresh state is entered. If DDR4 retry is enabled by CRCPARCTL1.crc_parity_retry_enable and software intervention is enabled by CRCPARCTL1.alert_wait_for_sw. Also monitor CRCPARSTAT.dfi_alert_err_int and CRCPARSTAT .dfi_alert_err_fatl_intduring the polling STAT.selfref_type. If one or more of them are asserted before the polling is done, complete the retry procedure prior to the subsequent steps. For LPDDR4, wait until Self Refresh Power Down state is entered.
 - Program DFIMISC.dfi_frequency as required. 0x1f for PHY Deep Sleep/Retention.
 - Set DFIMISC.dfi_init_start to '1'.

- The PHY de-asserts dfi_init_complete. The controller polls DFISTAT.dfi_init_complete=0.
- Set DFIMISC.dfi_init_start to '0'.
- The PHY asserts dfi_init_complete. The controller polls DFISTAT.dfi_init_complete=1.
- Place IOs in retention mode.
- Set MCTL_PWR_LPI_CTL.mctl_pwr_off_req to 1 to start low power sequence.
- Remove power VDD and VAA.

exit low power mode using software mode

- Supply VDD and VAA.
- Hardware will process NOC, clock, reset automatically.
- CPU check MCTL_LPI_STAT.mctl_lpi_work until its value equals 1.
 - Program INIT0.skip_dram_init = 2'b11. Skips the DRAM init routine and starts up in self-refresh mode.
 - Programs PWRCTL.selfref_sw = 1'b1 Keeps the controller in self-refresh mode.
 - Program DFIMISC.dfi_init_complete_en to 1'b0. PHY initialization needs to be rerun so set to '0' until initialization complete.
- Hardware detect ddrc_init_complete equals 1, deassert DDRC reset.
- Run PHY initialization/training as required, including removing the IOs from retention mode, including steps below.
- Program DFIMISC.dfi_frequency as required.
- Set DFIMISC.dfi_init_start to '1'.
- The PHY asserts dfi_init_complete. The controller polls DFISTAT.dfi_init_complete=1.
- Set DFIMISC.dfi_init_start to '0'.
- Program DFIMISC.dfi_init_complete_en to 1'b1. Indicates to controller that PHY has completed re-training/initialization
- Program PWRCTL.selfref_sw = 1'b0 . Trigger self-refresh exit.
- Poll STAT.selfref_type = 2'b00. Wait until self-refresh state is exited.
- Poll STAT.operating_mode for Normal Mode entry.
- Restore value DFILPCF0.dfi_lp_en_sr prior to self-refresh entry.
- Write PCTRL.port_en = 1. AXI ports are no longer blocked from taking transactions.
- Write '1' to SBRCTL.scrub_en Enable SBR if desired, instantiated only required if SBR instantiated.

Power on flow

- Power supply on
- Clock and reset are provided by hardware automatically.
- CPU check status of MCTL_LPI_STAT.ddrc_init until its value equals 1.
- CPU start to configure DDRC. When configuration of DDRC is finished, configure MCTL_PWR_LPI_CTL.mctl_ddrc_init_done to 1.
- Software begin to configure DDR PHY. When configuration of DDR PHY is finished, set MCTL_LPI_STAT.mctl_dfi_init_done to 1.

2.3.3.4 SOC power control

When all sub-module are powered off SOC can start to enter low power mode. SOC_SLEEP_MASK is used to mask module power status.

SOC_SLEEP_CTL.auto_slp_en is used to select SOC power mode. When SOC_SLEEP_CTL.auto_slp_en =1 automatic mode is selected. Otherwise software mode is selected.

automatic mode

- Set SOC_SLEEP_CTL.auto_slp_en to 1.

- Set xxx_PWR_LPI_CTL.xxx_pwr_off_req to 1, control each module enter low power state.
- Perform CPU powr off flow. CPUs will enter WFI state.
- When all module are powered off, SOC will enter sleep mode.

software mode

- Set SOC_SLEEP_CTL.auto_slp_en to 1.
- Set xxx_PWR_LPI_CTL.xxx_pwr_off_req to 1, control each module enter low power state.
- CPU check xxx_PWR_LPI_STAT.xxx_pcu_pwroff to ensure each module are powered off.
- Set SOC_SLEEP_CTL.core_sleep_req to 1.
- Perform CPU powr off flow. CPUs will enter WFI state.
- When all module are powered off, SOC will enter sleep mode.

SOC_SLEEP_CTL.core_sleep_mode is used to control whether power down PLLs. When its value equals 0 and SOC is at sleep0 mode, PLLs will be powered off. And SOC will enter sleep1 mode.

When SOC is at sleep mode, PD_Normal can be powered off through external PMIC.

There are three kinds of wakeupsources:

- PMU.
- GPIO.
- Timer.

CPU can configure which GPIO is used as wakeup source. Wakeup event can be level, pulse, rising edge, falling edge, double edge.

2.3.4 register summary

Global power control register base address : refer to [BOOT base address](#)

Table 2-3-2 global power control register summary

Offset	Register	Description
0x6c	SOC_SLEEP_CTL	soc power control
0x78	SOC_WAKUP_SRC	soc wakeup source
0x90	SYS_CTL_INT0_RAW	module powerdown raw status
0x94	SYS_CTL_INT0_EN	module powerdown interrupt enable
0x98	SYS_CTL_INT0_STAT	module powerdown status
0xA0	SYS_CTL_INT1_RAW	module powerup raw status
0xA4	SYS_CTL_INT1_EN	module powerup interrupt enable
0xA8	SYS_CTL_INT1_STAT	module powerup status
0xB0	SYS_CTL_INT2_RAW	sleep mode raw status
0xB4	SYS_CTL_INT2_EN	sleep mode interrupt enable
0xB8	SYS_CTL_INT2_STAT	sleep mode status
0x118	SOC_SLEEP_MASK	module sleep mask

Module power control register base address : refer to [PWR base address](#)

Table 2-3-3 module power control register summary

Offset	Register	Description
0x00	CPU0_PWR_TIM	cpu0 power control timer parameter
0x04	CPU0_LPI_TIM	cpu0 NOC LPI control timer parameter
0x08	CPU0_PWR_LPI_CTL	cpu0 power control
0x0c	CPU0_PWR_LPI_STAT	cpu0 power status
0x10	CPU1_PWR_TIM	cpu1 power control timer parameter
0x14	CPU1_LPI_TIM	cpu1 NOC LPI control timer parameter
0x18	CPU1_PWR_LPI_CTL	cpu1 power control
0x1c	CPU1_PWR_LPI_STAT	cpu1 power status
0x20	ai_PWR_TIM	ai power control timer parameter
0x24	ai_LPI_TIM	ai NOC LPI control timer parameter
0x28	ai_PWR_LPI_CTL	ai power control
0x2c	ai_PWR_LPI_STAT	ai power status
0x30	disp_PWR_TIM	disp power control timer parameter
0x34	disp_LPI_TIM	disp NOC LPI control timer parameter
0x38	disp_GPU_TIM	gpu LPI control timer parameter
0x3c	disp_LPI_CTL	disp power control
0x40	disp_LPI_STAT	disp power status
0x60	shrm_PWR_TIM	shrm power control timer parameter
0x64	shrm_LPI_TIM	shrm NOC LPI control timer parameter
0x68	shrm_PWR_LPI_CTL	shrm power control
0x6c	shrm_PWR_LPI_STAT	shrm power status
0x70	vpu_PWR_TIM	vpu power control timer parameter
0x74	vpu_LPI_TIM	vpu NOC LPI control timer parameter
0x78	vpu_QCH_TIM	vpu qchannel control timer parameter
0x7c	vpu_PWR_LPI_CTL	vpu power control
0x80	vpu_LPI_STAT	vpu power status
0x90	MCTL_PWR_TIM	MCTRL power control timer parameter
0x94	MCTL_NOC_LPI_TIM	MCTRL NOC LPI control timer parameter
0x98	MCTL_AXI_LPI_TIM	MCTRL ACI LPI control timer parameter
0x9c	MCTL_PWR_LPI_CTL	MCTRL power control
0xa0	MCTL_CLOCK_SWITCH	MCTRL clock switch
0xa4	MCTL_LPI_STAT	MCTRL power status
0x100	dpu_PWR_TIM	dpu power control timer parameter

Offset	Register	Description
0x104	dpu_LPI_TIM	dpu NOC LPI control timer parameter
0x108	dpu_PWR_LPI_CTL	dpu power control
0x10c	dpu_PWR_LPI_STAT	dpu power status
0x110	hi_PWR_TIM	hi power control timer parameter
0x114	hi_LPI_TIM	hi NOC LPI control timer parameter
0x118	hi_PWR_LPI_CTL	hi power control
0x11c	hi_LPI_STAT	hi power status
0x120	ls_PWR_TIM	ls power control timer parameter
0x124	ls_LPI_TIM	ls NOC LPI control timer parameter
0x128	ls_PWR_LPI_CTL	ls power control
0x12c	ls_LPI_STAT	ls power status
0x130	sec_PWR_TIM	sec power control timer parameter
0x134	sec_LPI_TIM	sec NOC LPI control timer parameter
0x138	sec_PWR_LPI_CTL	sec power control
0x13c	sec_PWR_LPI_STAT	sec power status
0x140	ISP_PWR_TIM	isp power control timer parameter
0x144	ISP_LPI_TIM	isp NOC LPI control timer parameter
0x148	ISP_PWR_LPI_CTL	isp power control
0x14c	ISP_PWR_LPI_STAT	isp power status
0x150	PMU_PWR_TIM	PMU power control timer parameter
0x154	PMU_LPI_TIM	PMU NOC LPI control timer parameter
0x158	PMU_PWR_LPI_CTL	PMU power control
0x15c	PMU_PWR_LPI_STAT	PMU power status
0x160	SRAM0_REPAIR_TIM	sram repair timer
0x164	SSYS_CTL_GPIO_CTL	gpio wakeup control
0x170	SSYS_CTL_GPIO_EN0	gpio wakeup enable
0x174	SSYS_CTL_GPIO_EN1	gpio wakeup enable
0x178	CPU_REPAIR_TIM	cpu wait repaire time

2.3.5 register description

global power control register description:

SOC_SLEEP_CTL

Offset=0x6c, Total Reset Value: 0x0000_0000, Write_enable=yes

Bits	Access	Name	Descriptions	Reset
4	RW	auto_slp_en	SoC core auto sleep enable control bit. If this field is set to '1', when all the power domains are in Power-Down mode, SoC core will automatically starts entering SLEEP mode flow. 0x0: disable auto sleep entering feature; 0x1: enable auto sleep entering feature.	0x0
1	RW	core_sleep_mode	0: powerdown pll when sleep0 time out 1: remain PLL stable. Stay in sleep0 until wakeup int	0x0
0	W1T	core_sleep_req	SoC core go sleep request bit. Writing '1' to this field will request SOC core to enter SLEEP mode. SoC core starts sleep-entering flow when all the power domains are in Power-Down mode. 0x0: no operation; 0x1: Request SoC core to enter SLEEP mode.	0x0

SOC_WAKUP_SRC

Offset=0x78, Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
7:4	RO	slp1_wakeup_cause	The cause of SoC waking up from SLEEP1 mode. There can be three source to waking up SoC in SLEEP1 mode, and they are GPIO,VAD,TIMER, and RTC event. This field will be auto-cleared when SoC enter SLEEP1 mode next time. Bit 0: if asserted, SoC is woken up by PMU generated event; Bit 1: if asserted, SoC is woken up by TIMER generate event; Bit 2: if asserted, SoC is woken up by GPIO generate event; Bit 3: not used	0x0
3:0	RO	slp0_wakeup_cause	The cause of SoC waking up from SLEEP0 mode. There can be three source to waking up SoC in SLEEP0 mode, and they are GPIO,VAD,TIMER, and RTC event. This field will be auto-cleared when SoC enter SLEEP0 mode next time. Bit 0: if asserted, SoC is woken up by PMU generated event; Bit 1: if asserted, SoC is woken up by TIMER generate event; Bit 2: if asserted, SoC is woken up by GPIO generate event; Bit 3: not used	0x0

SYS_CTL_INT0_RAW

Offset=0x90, Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
12	W1C	pd_pmu_PD_raw	PMU power domain go power-off interrupt raw status bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: no interrupt happens; 0x1: PD_ISP enter Power-Off state.	0x0

Bits	Access	Name	Descriptions	Reset
11	W1C	pd_shrm_PD_raw	shrm power domain go power-off interrupt raw status bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: no interrupt happens; 0x1: PD_ISP enter Power-Off state.	0x0
10	W1C	pd_sec_PD_raw	sec power domain go power-off interrupt raw status bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: no interrupt happens; 0x1: PD_ISP enter Power-Off state.	0x0
9	W1C	pd_ls_PD_raw	low sys power domain go power-off interrupt raw status bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: no interrupt happens; 0x1: PD_MCTL enter Power-Off state.	0x0
8	W1C	pd_hi_PD_raw	high sys power domain go power-off interrupt raw status bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: no interrupt happens; 0x1: PD_RAM1 enter Power-Off state.	0x0
7	W1C	pd_ddr_PD_raw	ddr power domain go power-off interrupt raw status bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: no interrupt happens; 0x1: PD_RAM0 enter Power-Off state.	0x0
6	W1C	pd_isp_PD_raw	isp power domain go power-off interrupt raw status bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: no interrupt happens; 0x1: PD_shrm enter Power-Off state.	0x0
5	W1C	pd_dpu_PD_raw	dpu power domain go power-off interrupt raw status bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: no interrupt happens; 0x1: PD_PERI enter Power-Off state.	0x0
4	W1C	pd_vpu_PD_raw	VPU power domain go power-off interrupt raw status bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: no interrupt happens; 0x1: PD_SEC enter Power-Off state.	0x0
3	W1C	pd_disp_PD_raw	disp power domgnnen go power-off interrupt raw status bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: no interrupt happens; 0x1: PD_GNNE enter Power-Off state.	0x0
2	W1C	pd_ai_PD_raw	AI power domain go power-off interrupt raw status bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: no interrupt happens; 0x1: PD_AI enter Power-Off state.	0x0
1	W1C	pd_cpu1_PD_raw	CPU1 power domain go power-off interrupt raw status bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: no interrupt happens; 0x1: PD_CPUup enter Power-Off state.	0x0

Bits	Access	Name	Descriptions	Reset
0	W1C	pd_cpu0_PD_raw	CPU0 power domain go power-off interrupt raw status bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: no interrupt happens; 0x1: PD_CPUm enter Power-Off state.	0x0

SYS_CTL_INT0_EN

Offset=0x94, Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
12	RW	pd_pmu_PD_ien	PMU power domain go power-off interrupt enable bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: disable interrupt; 0x1: enable interrupt.	0x0
11	RW	pd_shrm_PD_ien	shrm power domain go power-off interrupt enable bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: disable interrupt; 0x1: enable interrupt.	0x0
10	RW	pd_sec_PD_ien	sec power domain go power-off interrupt enable bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: disable interrupt; 0x1: enable interrupt.	0x0
9	RW	pd_ls_PD_ien	low sys Controller power domain go power-off interrupt enable bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: disable interrupt; 0x1: enable interrupt.	0x0
8	RW	pd_hi_PD_ien	hi sys power domain go power-off interrupt enable bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: disable interrupt; 0x1: enable interrupt.	0x0
7	RW	pd_ddr_PD_ien	ddr power domain go power-off interrupt enable bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: disable interrupt; 0x1: enable interrupt.	0x0
6	RW	pd_isp_PD_ien	isp power domain go power-off interrupt enable bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: disable interrupt; 0x1: enable interrupt.	0x0
5	RW	pd_dpu_PD_ien	Dpu power domain go power-off interrupt enable bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: disable interrupt; 0x1: enable interrupt.	0x0
4	RW	pd_vpu_PD_ien	vpu power domain go power-off interrupt enable bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: disable interrupt; 0x1: enable interrupt.	0x0
3	RW	pd_disp_PD_ien	disp power domain go power-off interrupt enable bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: disable interrupt; 0x1: enable interrupt.	0x0
2	RW	pd_ai_PD_ien	AI power domain go power-off interrupt enable bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: disable interrupt; 0x1: enable interrupt.	0x0

Bits	Access	Name	Descriptions	Reset
1	RW	pd_cpu1_PD_ien	CPU1 power domain go power-off interrupt enable bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: disable interrupt; 0x1: enable interrupt.	0x0
0	RW	pd_cpu0_PD_ien	CPU0 power domain go power-off interrupt enable bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: disable interrupt; 0x1: enable interrupt.	0x0

SYS_CTL_INT0_STAT

Offset=0x98, Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
12	RO	pd_pmu_PD_stat	PMU power domain go power-off interrupt status bit. 0x0: no interrupt; 0x1: interrupt happens.	0x0
11	RO	pd_shrm_PD_stat	shrm power domain go power-off interrupt status bit. 0x0: no interrupt; 0x1: interrupt happens.	0x0
10	RO	pd_sec_PD_stat	sec power domain go power-off interrupt status bit. 0x0: no interrupt; 0x1: interrupt happens.	0x0
9	RO	pd_ls_PD_stat	low sys power domain go power-off interrupt status bit. 0x0: no interrupt; 0x1: interrupt happens.	0x0
8	RO	pd_hi_PD_stat	hi sys power domain go power-off interrupt status bit. 0x0: no interrupt; 0x1: interrupt happens.	0x0
7	RO	pd_ddr_PD_stat	ddr power domain go power-off interrupt status bit. 0x0: no interrupt; 0x1: interrupt happens.	0x0
6	RO	pd_isp_PD_stat	isp power domain go power-off interrupt status bit. 0x0: no interrupt; 0x1: interrupt happens.	0x0
5	RO	pd_dpu_PD_stat	dpu power domain go power-off interrupt status bit. 0x0: no interrupt; 0x1: interrupt happens.	0x0
4	RO	pd_vpu_PD_stat	vpu power domain go power-off interrupt status bit 0x0: no interrupt; 0x1: interrupt happens.	0x0
3	RO	pd_disp_PD_stat	disp power domain go power-off interrupt status bit. 0x0: no interrupt; 0x1: interrupt happens.	0x0
2	RO	pd_ai_PD_stat	AI power domain go power-off interrupt status bit. 0x0: no interrupt; 0x1: interrupt happens.	0x0
1	RO	pd_cpu1_PD_stat	CPU1 power domain go power-off interrupt status bit. 0x0: no interrupt; 0x1: interrupt happens.	0x0
0	RO	pd_cpu0_PD_stat	CPU0 power domain go power-off interrupt status bit. 0x0: no interrupt; 0x1: interrupt happens.	0x0

SYS_CTL_INT1_RAW

Offset=0xA0, Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
12	W1C	pd_pmu_PU_raw	PMU power domain go power-up interrupt raw status bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: no interrupt happens; 0x1: PD_DISP enter Power-On state.	0x0
11	W1C	pd_shrm_PU_raw	shrm power domain go power-up interrupt raw status bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: no interrupt happens; 0x1: PD_DISP enter Power-On state.	0x0
10	W1C	pd_sec_PU_raw	sec power domain go power-up interrupt raw status bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: no interrupt happens; 0x1: PD_ISP enter Power-On state.	0x0
9	W1C	pd_ls_PU_raw	Low sys power domain go power-up interrupt raw status bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: no interrupt happens; 0x1: PD_MCTL enter Power-On state.	0x0
8	W1C	pd_hi_PU_raw	hi sys power domain go power-up interrupt raw status bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: no interrupt happens; 0x1: PD_RAM1 enter Power-On state.	0x0
7	W1C	pd_ddr_PU_raw	ddr power domain go power-up interrupt raw status bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: no interrupt happens; 0x1: PD_RAM0 enter Power-On state.	0x0
6	W1C	pd_isp_PU_raw	isp power domain go power-up interrupt raw status bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: no interrupt happens; 0x1: PD_shrm enter Power-On state.	0x0
5	W1C	pd_dpu_PU_raw	Dpu power domain go power-up interrupt raw status bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: no interrupt happens; 0x1: PD_PERI enter Power-On state.	0x0
4	W1C	pd_vpu_PU_raw	vpu power domain go power-up interrupt raw status bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: no interrupt happens; 0x1: PD_SEC enter Power-On state.	0x0
3	W1C	pd_disp_PU_raw	disp power domain go power-up interrupt raw status bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: no interrupt happens; 0x1: PD_GNNE enter Power-On state.	0x0
2	W1C	pd_ai_PU_raw	AI power domain go power-up interrupt raw status bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: no interrupt happens; 0x1: PD_AI enter Power-On state.	0x0
1	W1C	pd_cpu1_PU_raw	CPU1 power domain go power-up interrupt raw status bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: no interrupt happens; 0x1: PD_CPUp enter Power-On state.	0x0
0	W1C	pd_cpu0_PU_raw	CPU0 power domain go power-up interrupt raw status bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: no interrupt happens; 0x1: PD_CPUm enter Power-On state.	0x0

[SYS_CTL_INT1_EN](#)

Offset=0xA4, Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
12	RW	pd_pmu_PU_iен	PMU power domain go power-up interrupt enable bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: disable interrupt; 0x1: enable interrupt.	0x0
11	RW	pd_shrm_PU_iен	shrm power domain go power-up interrupt enable bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: disable interrupt; 0x1: enable interrupt.	0x0
10	RW	pd_sec_PU_iен	sec power domain go power-up interrupt enable bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: disable interrupt; 0x1: enable interrupt.	0x0
9	RW	pd_ls_PU_iен	low sys power domain go power-up interrupt enable bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: disable interrupt; 0x1: enable interrupt.	0x0
8	RW	pd_hi_PU_iен	hi sys power domain go power-up interrupt enable bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: disable interrupt; 0x1: enable interrupt.	0x0
7	RW	pd_ddr_PU_iен	ddr power domain go power-up interrupt enable bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: disable interrupt; 0x1: enable interrupt.	0x0
6	RW	pd_isp_PU_iен	isp power domain go power-up interrupt enable bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: disable interrupt; 0x1: enable interrupt.	0x0
5	RW	pd_dpu_PU_iен	Dpu power domain go power-up interrupt enable bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: disable interrupt; 0x1: enable interrupt.	0x0
4	RW	pd_vpu_PU_iен	vpu power domain go power-up interrupt enable bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: disable interrupt; 0x1: enable interrupt.	0x0
3	RW	pd_disp_PU_iен	disp power domain go power-up interrupt enable bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: disable interrupt; 0x1: enable interrupt.	0x0
2	RW	pd_ai_PU_iен	AI power domain go power-up interrupt enable bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: disable interrupt; 0x1: enable interrupt.	0x0
1	RW	pd_cpu1_PU_iен	CPU1 power domain go power-up interrupt enable bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: disable interrupt; 0x1: enable interrupt.	0x0
0	RW	pd_cpu0_PU_iен	CPU0 power domain go power-up interrupt enable bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: disable interrupt; 0x1: enable interrupt.	0x0

SYS_CTL_INT1_STAT

Offset=0xA8, Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
12	RO	pd_pmu_PU_stat	PMU power domain go power-up interrupt status bit. 0x0: no interrupt; 0x1: interrupt happens.	0x0
11	RO	pd_shrm_PU_stat	Shrm power domain go power-up interrupt status bit. 0x0: no interrupt; 0x1: interrupt happens.	0x0
10	RO	pd_sec_PU_stat	sec power domain go power-up interrupt status bit. 0x0: no interrupt; 0x1: interrupt happens.	0x0
9	RO	pd_ddr_PU_stat	Ddr power domain go power-up interrupt status bit. 0x0: no interrupt; 0x1: interrupt happens.	0x0
8	RO	pd_ls_PU_stat	low sys power domain go power-up interrupt status bit. 0x0: no interrupt; 0x1: interrupt happens.	0x0
7	RO	pd_hi_PU_stat	Hi sys power domain go power-up interrupt status bit. 0x0: no interrupt; 0x1: interrupt happens.	0x0
6	RO	pd_isp_PU_stat	isp power domain go power-up interrupt status bit. 0x0: no interrupt; 0x1: interrupt happens.	0x0
5	RO	pd_dpu_PU_stat	Dpu power domain go power-up interrupt status bit. 0x0: no interrupt; 0x1: interrupt happens.	0x0
4	RO	pd_vpu_PU_stat	vpu power domain go power-up interrupt status bit 0x0: no interrupt; 0x1: interrupt happens.	0x0
3	RO	pd_disp_PU_stat	Disp power domain go power-up interrupt status bit. 0x0: no interrupt; 0x1: interrupt happens.	0x0
2	RO	pd_ai_PU_stat	AI power domain go power-up interrupt status bit. 0x0: no interrupt; 0x1: interrupt happens.	0x0
1	RO	pd_cpu1_PU_stat	CPU1 power domain go power-up interrupt status bit. 0x0: no interrupt; 0x1: interrupt happens.	0x0
0	RO	pd_cpu0_PU_stat	CPU0 power domain go power-up interrupt status bit. 0x0: no interrupt; 0x1: interrupt happens.	0x0

SYS_CTL_INT2_RAW

Offset=0xB0, Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
4	RO	soc_slp1_wakup_raw	SoC core wake up from SLEEP1 mode interrupt raw status bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: no interrupt happens; 0x1: SoC core waking up from SLEEP1 mode.	0x0
3	RO	soc_slp0_wakup_raw	SoC core wake up from SLEEP0 mode interrupt raw status bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: no interrupt happens; 0x1: SoC core waking up from SLEEP0 mode.	0x0

Bits	Access	Name	Descriptions	Reset
1	RO	soc_go_slp1_raw	SoC core entering SLEEP1 mode interrupt raw status bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: no interrupt happens; 0x1: SoC core enters SLEEP1 mode.	0x0
0	RO	soc_go_slp0_raw	SoC core entering SLEEP0 mode interrupt raw status bit. Writing '1' to this bit will clear the raw interrupt status bit. 0x0: no interrupt happens; 0x1: SoC core enters SLEEP0 mode.	0x0

SYS_CTL_INT2_EN

Offset=0xB4, Total Reset Value: 0x0000_0000, Write_enable=yes

Bits	Access	Name	Descriptions	Reset
12	W1T	clr_slp1_wakeup_cause	clear INT2_RAW[4]	0x0
11	W1T	clr_slp0_wakeup_cause	clear INT2_RAW[3]	0x0
9	W1T	clr_soc_go_slp1_ack	clear INT2_RAW[1]	0x0
8	W1T	clr_soc_go_slp0_ack	clear INT2_RAW[0]	0x0
4	RW	soc_slp1_wakeup_iен	SoC core waking up from SLEEP1 mode interrupt enable bit. 0x0: disable interrupt; 0x1: enable interrupt.	0x0
3	RW	soc_slp0_wakeup_iен	SoC core waking up from SLEEP0 mode interrupt enable bit. 0x0: disable interrupt; 0x1: enable interrupt.	0x0
1	RW	soc_go_slp1_iен	SoC core entering SLEEP1 mode interrupt enable bit. 0x0: disable interrupt; 0x1: enable interrupt.	0x0
0	RW	soc_go_slp0_iен	SoC core entering SLEEP0 mode interrupt enable bit. 0x0: disable interrupt; 0x1: enable interrupt.	0x0

SYS_CTL_INT2_STAT

Offset=0xB8, Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
4	RO	soc_slp1_wakeup_stat	SoC core waking up from SLEEP1 mode interrupt status bit. 0x0: no interrupt; 0x1: interrupt happens.	0x0
3	RO	soc_slp0_wakeup_stat	SoC core waking up from SLEEP0 mode interrupt status bit 0x0: no interrupt; 0x1: interrupt happens.	0x0
1	RO	soc_go_slp1_stat	SoC core entering SLEEP1 mode interrupt status bit. 0x0: no interrupt; 0x1: interrupt happens.	0x0
0	RO	soc_go_slp0_stat	SoC core entering SLEEP0 mode interrupt status bit. 0x0: no interrupt; 0x1: interrupt happens.	0x0

SOC_SLEEP_MASK

Offset=0x118, Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
12	RW	pmu_slp_mask	PMU sleep mask bit. If this bit is set to '1', when SoC core enter to sleep mode, it will ignore PMU state. 0X0: unmask 0x1:mask	0x0
11	RW	shrm_slp_mask	shrm sleep mask bit. If this bit is set to '1', when SoC core enter to sleep mode, it will ignore shrm state. 0X0: unmask 0x1:mask	0x0
10	RW	sec_slp_mask	sec sleep mask bit. If this bit is set to '1', when SoC core enter to sleep mode, it will ignore sec state. 0X0: unmask 0x1:mask	0x0
9	RW	ls_slp_mask	low sys sleep mask bit. If this bit is set to '1', when SoC core enter to sleep mode, it will ignore low sys state. 0x0: 0X0: unmask 0x1:mask	0x0
8	RW	hs_slp_mask	hi sys sleep mask bit. If this bit is set to '1', when SoC core enter to sleep mode, it will ignore high sys state. 0X0: unmask 0x1:mask	0x0
7	RW	ddr_slp_mask	ddr sleep mask bit. If this bit is set to '1', when SoC core enter to sleep mode, it will ignore ddr state. 0X0: unmask 0x1:mask	0x0
6	RW	isp_slp_mask	Isp sleep mask bit. If this bit is set to '1', when SoC core enter to sleep mode, it will ignore ISP state. 0X0: unmask 0x1:mask	0x0
5	RW	dpu_slp_mask	dpu sleep mask bit. If this bit is set to '1', when SoC core enter to sleep mode, it will ignore dpu state. 0X0: unmask 0x1:mask	0x0
4	RW	vpu_slp_mask	vpu sleep mask bit. If this bit is set to '1', when SoC core enter to sleep mode, it will ignore vpu state. 0x0: unmask PD_sec sleep 0x1:mask PD_sec sleep	0x0
3	RW	disp_slp_mask	disp sleep mask bit. If this bit is set to '1', when SoC core enter to sleep mode, it will ignore disp state. 0X0: unmask 0x1:mask	0x0
2	RW	ai_slp_mask	AI sleep mask bit. If this bit is set to '1', when SoC core enter to sleep mode, it will ignore AI state. 0X0: unmask 0x1:mask	0x0

Bits	Access	Name	Descriptions	Reset
1	RW	cpu1_slp_mask	CPU1 sleep mask bit. If this bit is set to '1', when SoC core enter to sleep mode, it will ignore CPU1 state. 0X0: unmask 0x1:mask	0x0
0	RW	cpu0_slp_mask	CPU0 sleep mask bit. If this bit is set to '1', when SoC core enter to sleep mode, it will ignore CPU0 state. 0X0: unmask 0x1:mask	0x0

module power control register description

CPU0_PWR_TIM

Offset=0x0, Total Reset Value: 0x4008_0c05, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31:20	RW	cpu0_wfi_to_tim	cpu0 WFI signal waiting timeout value. After cpu0 wfi signal is asserted, a down-count timer start running. when the timer expires, PD_CPUm power-down sequence is started if cpu0_PWR_CTL.cpu0_auto_pwr_en is set to 1. N: $N * 0.25\mu s$, 0x0: forbidden.	0x400
19:12	RW	cpu0_pwr_su_tim	PD_CPU0 power switch control setup time. This is the time interval from chaning power switch control signal to power status reach stable. 0x1: 1 $0.25\mu s$; 0x2: 2 $0.25\mu s$; 0x3: 3 $0.25\mu s$; 0xFF: 255 $0.25\mu s$; 0x0: forbidden.	0x80
11:4	RW	cpu0_pd_hd_tim	PD_CPU0 power-off mode least resting time. This is the least time that the power domain need to stay after entering power-off mode. 0x1: 1 $0.25\mu s$; 0x2: 2 $0.25\mu s$; 0x3: 3 $0.25\mu s$; 0xFF: 255 $0.25\mu s$; 0x0: forbidden.	0xC0
3:0	RW	cpu0_iso_su_tim	PD_CPU0 ISO cell control setup time. This is the time interval from changing ISO cell control signal to ISO cell truly active or deactive. 0x1: 1 $0.25\mu s$; 0x2: 2 $0.25\mu s$; 0x3: 3 $0.25\mu s$; 0xE: 14 $0.25\mu s$; 0xF: 15 * $0.25\mu s$; 0x0: forbidden.	0x5

CPU0_LPI_TIM

Offset=0x4, Total Reset Value: 0x0005_0810, Write_enable=no

Bits	Access	Name	Descriptions	Reset
21:16	RW	cpu0_idle_hd_tim	cpu0 NOC power controller in idle min time. This is the min time that NOC power controller should stay when it's brought into idle mode. 0x1: 1 $0.25\mu s$; 0x2: 2 $0.25\mu s$; 0x3: 3 $0.25\mu s$; 0x3F: 63 $0.25\mu s$; other value: forbidden.	0x5

Bits	Access	Name	Descriptions	Reset
12:8	RW	cpu0_idle_to_tim	cpu0 NOC power controller idle timeout value. This is the max allowed time from assertion of idleAck to assertion of idle. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0x1F: 31 0.25us; other value: forbidden.	0x8
4:0	RW	cpu0_ack_to_tim	cpu0 NOC power controller idleAck timeout value. This is the max allowed time from assertion of idleReq to assertion of idleAck. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0x1F: 31 0.25us; other values: forbidden.	0x10

CPU0_PWR_LPI_CTL

Offset=0x8, Total Reset Value: 0x0000_0000, Write_enable=yes

Bits	Access	Name	Descriptions	Reset
7	RW	cpu0_tdt_dm_core_unavail	1: core is not available	0x0
6	WO	cpu0_pwr_mode	0: power off, 1: clock off	0x0
5	RWSC	cpu0_lpi_exit_idle	cpu0 NOC power controller exit IDLE trigger. This is a write-only self-clearing bit. 0x1: request cpu0 NOC domain exiting idle state; 0x0: no operation.	0x0
4	RWSC	cpu0_lpi_go_idle	cpu0 NOC power controller go IDLE trigger. This is a write-only self-clearing bit. 0x1: request cpu0 NOC domain entering idle state; 0x0: no operation.	0x0
3	RW	cpu0_hold_reset	whether to keep reset when cpu0 exit power-off mode. 0x1: keep applying reset when exit power-off mode; 0x0: remove reset when exit power-off mode.	0x0
2	RW	cpu0_auto_pwr_en	cpu0 power domain power controller auto-PD enable. 0x1: enable power control unit auto control mode; 0x0: disable power control unit auto control mode;	0x0
1	RWSC	cpu0_pwr_up_req	cpu0 power domain go POWER-ON trigger. This bit will be cleared by a hardware ack signal automatically. 0x1: request cpu0 NOC domain entering PowerOn state; 0x0: no operation.	0x0
0	RWSC	cpu0_pwr_off_req	cpu0 power domain go POWER-OFF trigger. This bit will be cleared by a hardware ack signal automatically. 0x1: request cpu0 NOC domain entering idle state; 0x0: no operation.	0x0

CPU0_PWR_LPI_STAT

Offset=0xc, Total Reset Value: 0x0000_000a, Write_enable=no

Bits	Access	Name	Descriptions	Reset

Bits	Access	Name	Descriptions	Reset
3	RO	cpu0_lpi_work	cpu0 NOC power controller work status bit. 0x1: NOC logics in this power domain is in mission state; 0x0: not in work state.	0x1
2	RO	cpu0_lpi_idle	cpu0 NOC power controller idle status bit. 0x1: NOC logics in this power domain is in idle state; 0x0: not in idle state.	0x0
1	RO	cpu0_pcu_pwrup	cpu0 power domain current POWER-UP state bit. 0x1: The power domain is in power-up state; 0x0: not in power-up state.	0x1
0	RO	cpu0_pcu_pwroff	cpu0 power domain current POWER-OFF state bit. 0x1: The power domain is in power-off state; 0x0: not in power-off state.	0x0

CPU1_PWR_TIM

Offset=0x10, Total Reset Value: 0x4008_0c05, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31:20	RW	cpu1_wfi_to_tim	cpu1 WFI signal waiting timeout value. After cpu1 wfi signal is asserted, a down-count timer start running. when the timer expires, PD_CPUUp power-down sequence is started if cpu1_PWR_CTL.cpu1_auto_pwr_en is set to 1. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0xFFE: 254 0.25us; 0xFFF: 255 * 0.25us; 0x0: forbidden.	0x400
19:12	RW	cpu1_pwr_su_tim	PD_CPU1 power switch control setup time. This is the time interval from changing power switch control signal to power status reach stable. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0xFF: 255 0.25us; 0x0: forbidden.	0x80
11:4	RW	cpu1_pd_hd_tim	PD_CPU1 power-off mode least resting time. This is the least time that the power domain need to stay after entering power-off mode. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0xFF: 255 0.25us; 0x0: forbidden.	0xC0
3:0	RW	cpu1_iso_su_tim	PD_CPU1 ISO cell control setup time. This is the time interval from changing ISO cell control signal to ISO cell truly active or deactive. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0xE: 14 0.25us; 0xF: 15 * 0.25us; 0x0: forbidden.	0x5

CPU1_LPI_TIM

Offset=0x14, Total Reset Value: 0x0005_0810, Write_enable=no

Bits	Access	Name	Descriptions	Reset

Bits	Access	Name	Descriptions	Reset
21:16	RW	cpu1_idle_hd_tim	cpu1 NOC power controller in idle min time. This is the min time that NOC power controller should stay when it's brought into idle mode. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0x3F: 63 0.25us; other value: forbidden.	0x5
12:8	RW	cpu1_idle_to_tim	cpu1 NOC power controller idle timeout value. This is the max allowed time from assertion of idleAck to assertion of idle. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0x1F: 31 0.25us; other value: forbidden.	0x8
4:0	RW	cpu1_ack_to_tim	cpu1 NOC power controller idleAck timeout value. This is the max allowed time from assertion of idleReq to assertion of idleAck. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0x1F: 31 0.25us; other values: forbidden.	0x10

CPU1_PWR_LPI_CTL

Offset=0x18, Total Reset Value: 0x0000_0000, Write_enable=yes

Bits	Access	Name	Descriptions	Reset
7	RW	cpu1_tdt_dm_core_unavail	1: core is not available	0x0
6	WO	cpu1_pwr_mode	0: power off, 1: clock off	0x0
5	RWSC	cpu1_lpi_exit_idle	cpu1 NOC power controller exit IDLE trigger. This is a write-only self-clearing bit. 0x1: request cpu1 NOC logic exiting idle state; 0x0: no operation.	0x0
4	RWSC	cpu1_lpi_go_idle	cpu1 NOC power controller go IDLE trigger. This is a write-only self-clearing bit. 0x1: request cpu1 NOC logic entering idle state; 0x0: no operation.	0x0
3	RW	cpu1_hold_reset	whether to keep reset when cpu1 exit power-off mode. 0x1: keep applying reset when exit power-off mode; 0x0: remove reset when exit power-off mode.	0x0
2	RW	cpu1_auto_pwr_en	cpu1 domain power controller auto-PD enable. 0x1: enable power control unit auto control mode; 0x0: disable power control unit auto control mode;	0x0
1	RWSC	cpu1_pwr_up_req	cpu1 power domain go POWER-ON trigger. This bit will be cleared by a hardware ack signal automatically. 0x1: request this power domain entering PowerOn state; 0x0: no operation.	0x0
0	RWSC	cpu1_pwr_off_req	cpu1 power domain go POWER-OFF trigger. This bit will be cleared by a hardware ack signal automatically. 0x1: request this power domain entering idle state; 0x0: no operation.	0x0

CPU1_PWR_LPI_STAT

Offset=0x1c, Total Reset Value: 0x0000_000a, Write_enable=no

Bits	Access	Name	Descriptions	Reset
3	RO	cpu1_lpi_work	cpu1 NOC power controller current WORK status bit. 0x1: NOC logics in this power domain is in mission state; 0x0: no operation.	0x1
2	RO	cpu1_lpi_idle	cpu1 NOC power controller current IDLE status bit. 0x1: NOC logics in this power domain is in idle state; 0x0: no operation.	0x0
1	RO	cpu1_pcu_pwrup	cpu1 power domain current POWER-UP status bit. 0x1: The power domain is in power-up state; 0x0: not in power-up state.	0x1
0	RO	cpu1_pcu_pwroff	cpu1 power domain current POWER-OFF status bit. 0x1: The power domain is in power-off state; 0x0: not in power-off state.	0x0

ai_PWR_TIM

Offset=0x20, Total Reset Value: 0x0008_0c05, Write_enable=no

Bits	Access	Name	Descriptions	Reset
19:12	RW	ai_pwr_su_tim	PD_AI power switch control setup time. This is the time interval from changing power switch control signal to power status reach stable. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0xFF: 255 0.25us; 0x0: forbidden.	0x80
11:4	RW	ai_pd_hd_tim	PD_AI power-off mode least resting time. This is the least time that the power domain need to stay after entering power-off mode. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0xFF: 255 0.25us; 0x0: forbidden.	0xC0
3:0	RW	ai_iso_su_tim	PD_AI ISO cell control setup time. This is the time interval from changing ISO cell control signal to ISO cell truly active or deactive. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0xE: 14 0.25us; 0xF: 15 * 0.25us; 0x0: forbidden.	0x5

ai_LPI_TIM

Offset=0x24, Total Reset Value: 0x0005_0810, Write_enable=no

Bits	Access	Name	Descriptions	Reset
21:16	RW	ai_idle_hd_tim	AI subsys NOC power controller in idle min time. This is the min time that NOC power controller should stay when it's brought into idle mode. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0x3F: 63 0.25us; other value: forbidden.	0x5

Bits	Access	Name	Descriptions	Reset
12:8	RW	ai_idle_to_tim	AI subsys NOC power controller idle timeout value. This is the max allowed time from assertion of idleAck to assertion of idle. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0x1F: 31 0.25us; other value: forbidden.	0x8
4:0	RW	ai_ack_to_tim	AI subsys NOC power controller idleAck timeout value. This is the max allowed time from assertion of idleReq to assertion of idleAck. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0x1F: 31 0.25us; other values: forbidden.	0x10

ai_PWR_LPI_CTL

Offset=0x28, Total Reset Value: 0x0000_0010, Write_enable=yes

Bits	Access	Name	Descriptions	Reset
5	WO	ai_pwr_mode	0: power off, 1: clock off	0x0
4	RWSC	ai_repair_grp_en	1: ai power domain need sram repair,	0x1
3	RWSC	ai_lpi_exit_idle	AI subsys NOC power controller exit IDLE trigger. This is a write-only self-clearing bit. 0x1: request NOC logic in this domain exiting idle state; 0x0: no operation.	0x0
2	RWSC	ai_lpi_go_idle	AI subsys NOC power controller go IDLE trigger. This is a write-only self-clearing bit. 0x1: request NOC logic in this domain entering idle state; 0x0: no operation.	0x0
1	RWSC	ai_pwr_up_req	AI subsystem power domain go POWER-ON trigger. This bit will be cleared by a hardware ack signal automatically. 0x1: request AI subsystem power domain entering PowerOn state; 0x0: no operation.	0x0
0	RWSC	ai_pwr_off_req	AI subsystem power domain go POWER-OFF trigger. This bit will be cleared by a hardware ack signal automatically. 0x1: request AI subsystem power domain entering PowerOff state; 0x0: no operation.	0x0

ai_PWR_LPI_STAT

Offset=0x2c, Total Reset Value: 0x0000_000a, Write_enable=no

Bits	Access	Name	Descriptions	Reset
3	RO	ai_lpi_work	AI subsystem NOC power controller WORK status bit. 0x1: NOC logics in AI subsystem power domain is in mission state; 0x0: not in mission state.	0x1
2	RO	ai_lpi_idle	AI subsystem NOC power controller IDLE status bit. 0x1: NOC logics in AI subsystem power domain is in idle state; 0x0: not in idle state.	0x0

Bits	Access	Name	Descriptions	Reset
1	RO	ai_pcu_pwrup	AI subsystem power domain current POWER-UP status bit. 0x1: AI subsystem power domain is in power-up state; 0x0: not in power-up state.	0x1
0	RO	ai_pcu_pwroff	AI subsystem power domain current POWER-OFF status bit. 0x1: AI subsystem power domain is in power-off state; 0x0: not in power-off state.	0x0

disp_PWR_TIM

Offset=0x30, Total Reset Value: 0x0008_0c05, Write_enable=no

Bits	Access	Name	Descriptions	Reset
19:12	RW	disp_pwr_su_tim	PD_disp power switch control setup time. This is the time interval from changing power switch control signal to power status reach stable. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0xFF: 255 0.25us; 0x0: forbidden.	0x80
11:4	RW	disp_pd_hd_tim	PD_disp power-off mode least resting time. This is the least time that the power domain need to stay after entering power-off mode. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0xFF: 255 0.25us; 0x0: forbidden.	0xC0
3:0	RW	disp_iso_su_tim	PD_disp ISO cell control setup time. This is the time interval from changing ISO cell control signal to ISO cell truly active or deactive. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0xE: 14 0.25us; 0xF: 15 * 0.25us; 0x0: forbidden.	0x5

disp_LPI_TIM

Offset=0x34, Total Reset Value: 0x0005_0810, Write_enable=no

Bits	Access	Name	Descriptions	Reset
21:16	RW	disp_idle_hd_tim	disp subsys NOC power controller in idle min time. This is the min time that NOC power controller should stay when it's brought into idle mode. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0x3F: 63 0.25us; other value: forbidden.	0x5
12:8	RW	disp_idle_to_tim	disp subsys NOC power controller idle timeout value. This is the max allowed time from assertion of idleAck to assertion of idle. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0x1F: 31 0.25us; other value: forbidden.	0x8
4:0	RW	disp_ack_to_tim	disp subsys NOC power controller idleAck timeout value. This is the max allowed time from assertion of idleReq to assertion of idleAck. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0x1F: 31 0.25us; other values: forbidden.	0x10

disp_GPU_TIM

Offset=0x38, Total Reset Value: 0x0000_1010, Write_enable=no

Bits	Access	Name	Descriptions	Reset
15:8	RW	disp_gpu_idle_hd_tim	gpu axi low power interface stay in idle time	0x10
7:0	RW	disp_gpu_idle_su_tim	gpu axi low power interface wait for active/ack time	0x10

disp_LPI_CTL

Offset=0x3c, Total Reset Value: 0x0000_0000, Write_enable=yes

Bits	Access	Name	Descriptions	Reset
6	WO	disp_pwr_mode	0: power off, 1: clock off	0x0
5	RWSC	gpu_exit_idle	1: gpu axi exit idle request.	0x0
4	RWSC	gpu_go_idle	1: gpu axi go idle request.	0x0
3	RWSC	disp_lpi_exit_idle	disp subsys NOC power controller exit IDLE trigger. This is a write-only self-clearing bit. 0x1: request NOC logic in this domain exiting idle state; 0x0: no operation.	0x0
2	RWSC	disp_lpi_go_idle	disp subsys NOC power controller go IDLE trigger. This is a write-only self-clearing bit. 0x1: request NOC logic in this domain entering idle state; 0x0: no operation.	0x0
1	RWSC	disp_pwr_up_req	disp subsystem power domain go POWER-ON trigger. This bit will be cleared by a hardware ack signal automatically. 0x1: request disp subsystem power domain entering PowerOn state; 0x0: no operation.	0x0
0	RWSC	disp_pwr_off_req	disp subsystem power domain go POWER-OFF trigger. This bit will be cleared by a hardware ack signal automatically. 0x1: request disp subsystem power domain entering PowerOff state; 0x0: no operation.	0x0

disp_LPI_STAT

Offset=0x40, Total Reset Value: 0x0000_002a, Write_enable=no

Bits	Access	Name	Descriptions	Reset
5	RO	disp_axi_work	Display gpu module axi low power interface information. 1: disp gpu axi work state 0: disp gpu axi not work statae	0x1
4	RO	disp_axi_idle	Display gpu module axi low power interface information. 1: disp gpu axi idle state 0: disp gpu axi not idle state	0x0

Bits	Access	Name	Descriptions	Reset
3	RO	disp_lpi_work	disp subsystem NOC power controller WORK status bit. 0x1: NOC logics in disp subsystem power domain is in mission state; 0x0: not in mission state.	0x1
2	RO	disp_lpi_idle	disp subsystem NOC power controller IDLE status bit. 0x1: NOC logics in disp subsystem power domain is in idle state; 0x0: not in idle state.	0x0
1	RO	disp_pcu_pwrup	disp subsystem power domain current POWER-UP status bit. 0x1: disp subsystem power domain is in power-up state; 0x0: not in power-up state.	0x1
0	RO	disp_pcu_pwroff	disp subsystem power domain current POWER-OFF status bit. 0x1: disp subsystem power domain is in power-off state; 0x0: not in power-off state.	0x0

shrm_PWR_TIM

Offset=0x60, Total Reset Value: 0x0008_0c05, Write_enable=no

Bits	Access	Name	Descriptions	Reset
19:12	RW	shrm_pwr_su_tim	Shrm power switch control setup time. This is the time interval from changing power switch control signal to power status reach stable. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0xFF: 255 0.25us; 0x0: forbidden.	0x80
11:4	RW	shrm_pd_hd_tim	Shrm power-off mode least resting time. This is the least time that the power domain need to stay after entering power-off mode. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0xFF: 255 0.25us; 0x0: forbidden.	0xC0
3:0	RW	shrm_iso_su_tim	shrm ISO cell control setup time. This is the time interval from changing ISO cell control signal to ISO cell truly active or deactive. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0xE: 14 0.25us; 0xF: 15 * 0.25us; 0x0: forbidden.	0x5

shrm_LPI_TIM

Offset=0x64, Total Reset Value: 0x0005_0810, Write_enable=no

Bits	Access	Name	Descriptions	Reset
21:16	RW	shrm_idle_hd_tim	shrm subsys NOC power controller in idle min time. This is the min time that NOC power controller should stay when it's brought into idle mode. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0x3F: 63 0.25us; other value: forbidden.	0x5

Bits	Access	Name	Descriptions	Reset
12:8	RW	shrm_idle_to_tim	shrm subsys NOC power controller idle timeout value. This is the max allowed time from assertion of idleAck to assertion of idle. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0x1F: 31 0.25us; other value: forbidden.	0x8
4:0	RW	shrm_ack_to_tim	shrm subsys NOC power controller idleAck timeout value. This is the max allowed time from assertion of idleReq to assertion of idleAck. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0x1F: 31 0.25us; other values: forbidden.	0x10

shrm_PWR_LPI_CTL

Offset=0x68, Total Reset Value: 0x0000_0010, Write_enable=yes

Bits	Access	Name	Descriptions	Reset
5	WO	shrm_pwr_mode	0: power off, 1: clock off	0x0
4	RWSC	shrm_repair_grp_en	1: shrm power domain need sram repair,	0x1
3	RWSC	shrm_lpi_exit_idle	shrm subsystem NOC power controller exit IDLE trigger. This is a write-only self-clearing bit. 0x1: request NOC logic in this domain exiting idle state; 0x0: no operation.	0x0
2	RWSC	shrm_lpi_go_idle	shrm subsystem NOC power controller go IDLE trigger. This is a write-only self-clearing bit. 0x1: request NOC logic in this domain entering idle state; 0x0: no operation.	0x0
1	RWSC	shrm_pwr_up_req	shrm subsystem power domain go POWER-ON trigger. This bit will be cleared by a hardware ack signal automatically. 0x1: request this power domain entering PowerOn state; 0x0: no operation.	0x0
0	RWSC	shrm_pwr_off_req	shrm subsystem power domain go POWER-OFF trigger. This bit will be cleared by a hardware ack signal automatically. 0x1: request this power domain entering PowerOff state; 0x0: no operation.	0x0

shrm_PWR_LPI_STAT

Offset=0x6c, Total Reset Value: 0x0000_000a, Write_enable=no

Bits	Access	Name	Descriptions	Reset
3	RO	shrm_lpi_work	shrm subsystem NOC power controller WORK status bit. 0x1: NOC logics in this power domain is in mission state; 0x0: not in work status.	0x1
2	RO	shrm_lpi_idle	shrm subsystem NOC power controller IDLE status bit. 0x1: NOC logics in this power domain is in idle state; 0x0: not in idle status.	0x0

Bits	Access	Name	Descriptions	Reset
1	RO	shrm_pcu_pwrup	shrm subsystem power domain current POWER-UP status bit. 0x1: the power domain is in power-up state; 0x0: not in power-up status.	0x1
0	RO	shrm_pcu_pwroff	shrm subsystem power domain current POWER-OFF status bit. 0x1: the power domain is in power-off state; 0x0: not in power-off status.	0x0

vpu_PWR_TIM

Offset=0x70, Total Reset Value: 0x0008_0c05, Write_enable=no

Bits	Access	Name	Descriptions	Reset
20	RW	vpu_pwakeup	vpu pwakeup signal.	0x0
19:12	RW	vpu_pwr_su_tim	PD_vpu power switch control setup time. This is the time interval from changing power switch control signal to power status reach stable. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0xFF: 255 0.25us; 0x0: forbidden.	0x80
11:4	RW	vpu_pd_hd_tim	PD_vpu power-off mode least resting time. This is the least time that the power domain need to stay after entering power-off mode. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0xFF: 255 0.25us; 0x0: forbidden.	0xC0
3:0	RW	vpu_iso_su_tim	PD_vpu ISO cell control setup time. This is the time interval from changing ISO cell control signal to ISO cell truly active or deactive. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0xE: 14 0.25us; 0xF: 15 * 0.25us; 0x0: forbidden.	0x5

vpu_LPI_TIM

Offset=0x74, Total Reset Value: 0x0005_0810, Write_enable=no

Bits	Access	Name	Descriptions	Reset
21:16	RW	vpu_idle_hd_tim	vpu subsys NOC power controller in idle min time. This is the min time that NOC power controller should stay when it's brought into idle mode. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0x3F: 63 0.25us; other value: forbidden.	0x5
12:8	RW	vpu_idle_to_tim	vpu subsys NOC power controller idle timeout value. This is the max allowed time from assertion of idleAck to assertion of idle. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0x1F: 31 0.25us; other value: forbidden.	0x8
4:0	RW	vpu_ack_to_tim	vpu subsys NOC power controller idleAck timeout value. This is the max allowed time from assertion of idleReq to assertion of idleAck. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0x1F: 31 0.25us; other values: forbidden.	0x10

vpu_QCH_TIM

Offset=0x78, Total Reset Value: 0x0000_1010, Write_enable=no

Bits	Access	Name	Descriptions	Reset
15:8	RW	vpu_qch_hd_tim	vpu qchannel stay in idle time	0x10
7:0	RW	vpu_qch_su_tim	vpu qchannel stay in QRUN, QCONTINUE, QEXIT time	0x10

vpu_PWR_LPI_CTL

Offset=0x7c, Total Reset Value: 0x0000_0000, Write_enable=yes

Bits	Access	Name	Descriptions	Reset
6	WO	vpu_pwr_mode	0: power off, 1: clock off	0x0
5	RWSC	vpu_qch_exit_idle	Vpu qchannel exit IDLE trigger. This is a write-only self-clearing bit. 0x1: request NOC logic in this domain exiting idle state; 0x0: no operation.	0x0
4	RWSC	vpu_qch_go_idle	Vpu qchannel go IDLE trigger. This is a write-only self-clearing bit. 0x1: request NOC logic in this domain entering idle state; 0x0: no operation.	0x0
3	RWSC	vpu_lpi_exit_idle	Vpu subsys NOC power controller exit IDLE trigger. This is a write-only self-clearing bit. 0x1: request NOC logic in this domain exiting idle state; 0x0: no operation.	0x0
2	RWSC	vpu_lpi_go_idle	Vpu subsystem NOC power controller go IDLE trigger. This is a write-only self-clearing bit. 0x1: request NOC logic in this domain entering idle state; 0x0: no operation.	0x0
1	RWSC	vpu_pwr_up_req	Vpu subsystem power domain go POWER-ON trigger. This bit will be cleared by a hardware ack signal automatically. 0x1: request this power domain entering PowerOn state; 0x0: no operation.	0x0
0	RWSC	vpu_pwr_off_req	Vpu subsystem power domain go POWER-OFF trigger. This bit will be cleared by a hardware ack signal automatically. 0x1: request this power domain entering idle state; 0x0: no operation.	0x0

vpu_LPI_STAT

Offset=0x80, Total Reset Value: 0x0000_002a, Write_enable=no

Bits	Access	Name	Descriptions	Reset
5	RO	vpu_qch_state_work	1: vpu qchannel work	0x1
4	RO	vpu_qch_state_idle	1: vpu qchannel idle	0x0

Bits	Access	Name	Descriptions	Reset
3	RO	vpu_lpi_work	Vpu subsystem NOC power controller WORK status bit. 0x1: NOC logics in vpu subsys power domain is in mission state; 0x0: no operation.	0x1
2	RO	vpu_lpi_idle	Vpu subsys NOC power controller IDLE status bit. 0x1: NOC logics in vpu subsys power domain is in idle state; 0x0: no operation.	0x0
1	RO	vpu_pcu_pwrup	Vpu subsystem power domain current POWER-UP status bit. 0x1: vpupheral subsys power domain is in power-up state; 0x0: no operation.	0x1
0	RO	vpu_pcu_pwroff	Vpu subsystem power domain current POWER-OFF status bit. 0x1: vpu subsys power domain is in power-off state; 0x0: no operation.	0x0

MCTL_PWR_TIM

Offset=0x90, Total Reset Value: 0x0008_0c05, Write_enable=no

Bits	Access	Name	Descriptions	Reset
19:12	RW	mctl_pwr_su_tim	PD_MCTL power switch control setup time. This is the time interval from chaning power switch control signal to power status reach stable. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0xFF: 255 0.25us; 0x0: forbidden.	0x80
11:4	RW	mctl_pd_hd_tim	PD_MCTL power-off mode least resting time. This is the least time that the power domain need to stay after entering power-off mode. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0xFF: 255 0.25us; 0x0: forbidden.	0xC0
3:0	RW	mctl_iso_su_tim	PD_MCTL ISO cell control setup time. This is the time interval from changing ISO cell control signal to ISO cell truly active or deactive. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0xE: 14 0.25us; 0xF: 15 * 0.25us; 0x0: forbidden.	0x5

MCTL_NOC_LPI_TIM

Offset=0x94, Total Reset Value: 0x0005_0810, Write_enable=no

Bits	Access	Name	Descriptions	Reset
21:16	RW	mctl_idle_hd_tim	Memory Controller NOC power controller in idle min time. This is the min time that NOC power controller should stay when it's brought into idle mode. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0x3F: 63 0.25us; other value: forbidden.	0x5
12:8	RW	mctl_idle_to_tim	Memory Controller NOC power controller idle timeout value. This is the max allowed time from assertion of idleAck to assertion of idle. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0x1F: 31 0.25us; other value: forbidden.	0x8

Bits	Access	Name	Descriptions	Reset
4:0	RW	mctl_ack_to_tim	Memory Controller NOC power controller idleAck timeout value. This is the max allowed time from assertion of idleReq to assertion of idleAck. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0x1F: 31 0.25us; other values: forbidden.	0x10

MCTL_AXI_LPI_TIM

Offset=0x98, Total Reset Value: 0x0000_1010, Write_enable=no

Bits	Access	Name	Descriptions	Reset
15:8	RW	mctl_axi_hd_tim	ddrc axi port or ddrc core stay in idle time	0x10
7:0	RW	mctl_axi_su_tim	ddrc axi port stay in ACK, ACKN, ACTIVE state time	0x10

MCTL_PWR_LPI_CTL

Offset=0x9c, Total Reset Value: 0x0021_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
21	RW	mctl_hw_pwr_mode	1: hardware auto trigger self refresh through ddrc low power interface 0: software control ddrc self refresh	0x1
20	RW	mctl_ddrc_core_clken	1: use ddrc core low power interface to control ddrc core clock During ddr low power process this bit should be set to 0, otherwise ddrc core clock will be gated. This doesn't fulfill ddrc requirement. During ddrc bring up clock should be stable.	0x0
19	RW	mctl_pwr_mode	0: power off, 1: clock off	0x0
18	RW	mctl_dfi_init_done	after initialize ddr phy, disable ddrc low power interface	0x0
17	RW	mctl_ddrc_init_done	when power up, first initialize ddrc, after initialize dessert core reset	0x0
16	RWSC	mctl_pwr_ok_in	rmu set to 1 during reset phase after power up, when need to sleep, soft set this bit to 0	0x0
15	RWSC	mctl_axi4_exit_idle	ddrc axi port4 clock domain exit low power request	0x0
14	RWSC	mctl_axi3_exit_idle	ddrc axi port3 clock domain exit low power request	0x0
13	RWSC	mctl_axi2_exit_idle	ddrc axi port2 clock domain exit low power request	0x0
12	RWSC	mctl_axi1_exit_idle	ddrc axi port1 clock domain exit low power request	0x0
11	RWSC	mctl_axi0_exit_idle	ddrc axi port0 clock domain exit low power request	0x0
10	RWSC	mctl_exit_idle	ddrc core exit low power request	0x0
9	RWSC	mctl_axi4_go_idle	ddrc axi port4 clock domain low power request	0x0
8	RWSC	mctl_axi3_go_idle	ddrc axi port3 clock domain low power request	0x0
7	RWSC	mctl_axi2_go_idle	ddrc axi port2 clock domain low power request	0x0

Bits	Access	Name	Descriptions	Reset
6	RWSC	mctl_axi1_go_idle	ddrc axi port1 clock domain low power request	0x0
5	RWSC	mctl_axi0_go_idle	ddrc axi port0 clock domain low power request	0x0
4	RWSC	mctl_go_idle	ddrc core low power request	0x0
3	RWSC	mctl_lpi_exit_idle	Memory Controller NOC power controller exit IDLE trigger. This is a write-only self-clearing bit. 0x1: request NOC logic in this power domain exiting idle state; 0x0: no operation.	0x0
2	RWSC	mctl_lpi_go_idle	Memory Controller NOC power controller go IDLE trigger. This is a write-only self-clearing bit. 0x1: request NOC logic in this power domain entering idle state; 0x0: no operation.	0x0
1	RWSC	mctl_pwr_up_req	Memory Controller power domain go POWER-ON trigger. This bit will be cleared by a hardware ack signal automatically. 0x1: request Memory Controller power domain entering PowerOn state; 0x0: no operation.	0x0
0	RWSC	mctl_pwr_off_req	Memory Controller power domain go POWER-OFF trigger. This bit will be cleared by a hardware ack signal automatically. 0x1: request Memory Controller power domain entering idle state; 0x0: no operation.	0x0

MCTL_CLOCK_SWITCH

Offset=0xa0, Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
4	RW	dfi1_enabled	dfi1_enabled	0x0
3	RW	csysdiscamdrain_ddrc	csysdiscamdrain_ddrc	0x0
2:1	RW	csysfrequency_ddrc	csysfrequency_ddrc	0x0
0	RW	csysmode	csysmode	0x0

MCTL_LPI_STAT

Offset=0xa4, Total Reset Value: 0x000a_aac0, Write_enable=no

Bits	Access	Name	Descriptions	Reset
19	RO	ddrc_init	1: when power up, enter ddrc init state.	0x0
18	RO	mctl_axi4_idle	1: ddrc port4 idle	0x0
17	RO	mctl_axi4_work	1: ddrc port4 work	0x0
16	RO	mctl_axi3_idle	1: ddrc port3 idle	0x0
15	RO	mctl_axi3_work	1: ddrc port3 work	0x0

Bits	Access	Name	Descriptions	Reset
14	RO	mctl_axi2_idle	1: ddrc port2 idle	0x0
13	RO	mctl_axi2_work	1: ddrc port2 work	0x0
12	RO	mctl_axi1_idle	1: ddrc port1 idle	0x0
11	RO	mctl_axi1_work	1: ddrc port1 work	0x0
10	RO	mctl_axi0_idle	1: ddrc port0 idle	0x0
9	RO	mctl_axi0_work	1: ddrc port0 work	0x0
8	RO	mctl_ddrc_idle	1: ddrc core idle	0x0
7	RO	mctl_ddrc_work	1: ddrc core work	0x0
6	RO	mctl_lpi_work	Memory Controller NOC power controller is in WORK state. 0x1: NOC logics in this power domain is in mission state; 0x0: no change.	0x1
5	RO	mctl_lpi_idle	Memory Controller NOC power controller is in IDLE state. 0x1: NOC logics in this power domain is in idle state; 0x0: no change.	0x0
4 :0	RO	mctl_pwr_stat	Memory Controller Power Control State: 0x0: Work; 0x1:DisconnectNoC; 0x2:SelfRefresh; 0x3:Retention; 0x4:ResetOn; 0x5:EnableISO; 0x6:PowerOff; 0x7:BringUpPower; 0x8:DisableISO; 0x9:ResetClkOn ;0xa:ResetRecover; 0xb:ResetRemove; 0xc:MemctlInit; 0xd:ConnectNoC.	0x0

dpu_PWR_TIM

Offset=0x100, Total Reset Value: 0x0008_0c05, Write_enable=no

Bits	Access	Name	Descriptions	Reset
19:12	RW	dpu_pwr_su_tim	DPU power switch control setup time. This is the time interval from changing power switch control signal to power status reach stable. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0xFF: 255 0.25us; 0x0: forbidden.	0x80
11:4	RW	dpu_pd_hd_tim	DPU power-off mode least resting time. This is the least time that the power domain need to stay after entering power-off mode. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0xFF: 255 0.25us; 0x0: forbidden.	0xC0
3:0	RW	dpu_iso_su_tim	DPU ISO cell control setup time. This is the time interval from changing ISO cell control signal to ISO cell truly active or deactive. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0xE: 14 0.25us; 0xF: 15 * 0.25us; 0x0: forbidden.	0x5

dpu_LPI_TIM

Offset=0x104, Total Reset Value: 0x0005_0810, Write_enable=no

Bits	Access	Name	Descriptions	Reset

Bits	Access	Name	Descriptions	Reset
21:16	RW	dpu_idle_hd_tim	DPU NOC power controller in idle min time. This is the min time that NOC power controller should stay when it's brought into idle mode. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0x3F: 63 0.25us; other value: forbidden.	0x5
12:8	RW	dpu_idle_to_tim	DPU NOC power controller idle timeout value. This is the max allowed time from assertion of idleAck to assertion of idle. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0x1F: 31 0.25us; other value: forbidden.	0x8
4:0	RW	dpu_ack_to_tim	DPU NOC power controller idleAck timeout value. This is the max allowed time from assertion of idleReq to assertion of idleAck. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0x1F: 31 0.25us; other values: forbidden.	0x10

dpu_PWR_LPI_CTL

Offset=0x108, Total Reset Value: 0x0000_0000, Write_enable=yes

Bits	Access	Name	Descriptions	Reset
4	WO	dpu_pwr_mode	0: power off, 1: clock off	0x0
3	RWSC	dpu_lpi_exit_idle	DPU NOC power controller exit IDLE trigger. This is a write-only self-clearing bit. 0x1: request NOC logic in this power domain exiting idle state; 0x0: no operation.	0x0
2	RWSC	dpu_lpi_go_idle	DPU NOC power controller go IDLE trigger. This is a write-only self-clearing bit. 0x1: request NOC logic in this power domain entering idle state; 0x0: no operation.	0x0
1	RWSC	dpu_pwr_up_req	DPU power domain go POWER-ON trigger. This bit will be cleared by a hardware ack signal automatically. 0x1: request this power domain entering PowerOn state; 0x0: no operation.	0x0
0	RWSC	dpu_pwr_off_req	DPU power domain go POWER-OFF trigger. This bit will be cleared by a hardware ack signal automatically. 0x1: request the power domain entering idle state; 0x0: no operation.	0x0

dpu_PWR_LPI_STAT

Offset=0x10c, Total Reset Value: 0x0000_000a, Write_enable=no

Bits	Access	Name	Descriptions	Reset
3	RO	dpu_lpi_work	SRAM block 0 NOC power controller is in WORK state. 0x1: NOC logics in this power domain is in mission state; 0x0: no change.	0x1

Bits	Access	Name	Descriptions	Reset
2	RO	dpu_lpi_idle	SRAM block 0 NOC power controller is in IDLE state. 0x1: NOC logics in this power domain is in idle state; 0x0: no change.	0x0
1	RO	dpu_pcu_pwrup	SRAM block 0 power domain is currently in POWER-UP state. 0x1: the power domain is in power-up state; 0x0: no change.	0x1
0	RO	dpu_pcu_pwroff	SRAM block 0 power domain is currently in POWER-OFF state. 0x1: the power domain is in power-off state; 0x0: no change.	0x0

hi_PWR_TIM

Offset=0x110, Total Reset Value: 0x0008_0c05, Write_enable=no

Bits	Access	Name	Descriptions	Reset
19:12	RW	hi_pwr_su_tim	hi sys power switch control setup time. This is the time interval from changing power switch control signal to power status reach stable. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0xFF: 255 0.25us; 0x0: forbidden.	0x80
11:4	RW	hi_pd_hd_tim	hi sys power-off mode least resting time. This is the least time that the power domain need to stay after entering power-off mode. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0xFF: 255 0.25us; 0x0: forbidden.	0xC0
3:0	RW	hi_iso_su_tim	hi sys ISO cell control setup time. This is the time interval from changing ISO cell control signal to ISO cell truly active or deactive. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0xE: 14 0.25us; 0xF: 15 * 0.25us; 0x0: forbidden.	0x5

hi_LPI_TIM

Offset=0x114, Total Reset Value: 0x0005_0810, Write_enable=no

Bits	Access	Name	Descriptions	Reset
21:16	RW	hi_idle_hd_tim	hi sys NOC power controller in idle min time. This is the min time that NOC power controller should stay when it's brought into idle mode. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0x3F: 63 0.25us; other value: forbidden.	0x5
12:8	RW	hi_idle_to_tim	hi sys NOC power controller idle timeout value. This is the max allowed time from assertion of idleAck to assertion of idle. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0x1F: 31 0.25us; other value: forbidden.	0x8
4:0	RW	hi_ack_to_tim	hi sys NOC power controller idleAck timeout value. This is the max allowed time from assertion of idleReq to assertion of idleAck. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0x1F: 31 0.25us; other values: forbidden.	0x10

hi_PWR_LPI_CTL

Offset=0x118, Total Reset Value: 0x0000_0000, Write_enable=yes

Bits	Access	Name	Descriptions	Reset
4	WO	hi_pwr_mode	0: power off, 1: clock off	0x0
3	RWSC	hi_lpi_exit_idle	hi sys NOC power controller exit IDLE trigger. This is a write-only self-clearing bit. 0x1: request NOC logic in this power domain exiting idle state; 0x0: no operation.	0x0
2	RWSC	hi_lpi_go_idle	hi sys NOC power controller go IDLE trigger. This is a write-only self-clearing bit. 0x1: request NOC logic in this power domain entering idle state; 0x0: no operation.	0x0
1	RWSC	hi_pwr_up_req	Hi sys power domain go POWER-ON trigger. This bit will be cleared by a hardware ack signal automatically. 0x1: request the power domain entering PowerOn state; 0x0: no operation.	0x0
0	RWSC	hi_pwr_off_req	hi sys power domain go POWER-OFF trigger. This bit will be cleared by a hardware ack signal automatically. 0x1: request the power domain entering idle state; 0x0: no operation.	0x0

hi_LPI_STAT

Offset=0x11c, Total Reset Value: 0x0000_000a, Write_enable=no

Bits	Access	Name	Descriptions	Reset
3	RO	hi_lpi_work	hi sys NOC power controller is in WORK state. 0x1: NOC logics in this power domain is in mission state; 0x0: no change.	0x1
2	RO	hi_lpi_idle	hi sys NOC power controller is in IDLE state. 0x1: NOC logics in this power domain is in idle state; 0x0: no change.	0x0
1	RO	hi_pcu_pwrup	hi sys power domain is currently in POWER-UP state. 0x1: the power domain is in power-up state; 0x0: no change.	0x1
0	RO	hi_pcu_pwroff	hi sys power domain is currently in POWER-OFF state. 0x1: the power domain is in power-off state; 0x0: no change.	0x0

Is_PWR_TIM

Offset=0x120, Total Reset Value: 0x0008_0c05, Write_enable=no

Bits	Access	Name	Descriptions	Reset

Bits	Access	Name	Descriptions	Reset
19:12	RW	ls_pwr_su_tim	PD_Is power switch control setup time. This is the time interval from changing power switch control signal to power status reach stable. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0xFF: 255 0.25us; 0x0: forbidden.	0x80
11:4	RW	ls_pd_hd_tim	PD_Is power-off mode least resting time. This is the least time that the power domain need to stay after entering power-off mode. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0xFF: 255 0.25us; 0x0: forbidden.	0xC0
3:0	RW	ls_iso_su_tim	PD_Is ISO cell control setup time. This is the time interval from changing ISO cell control signal to ISO cell truly active or deactive. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0xE: 14 0.25us; 0xF: 15 * 0.25us; 0x0: forbidden.	0x5

Is_LPI_TIM

Offset=0x124, Total Reset Value: 0x0005_0810, Write_enable=no

Bits	Access	Name	Descriptions	Reset
21:16	RW	ls_idle_hd_tim	low sys NOC power controller in idle min time. This is the min time that NOC power controller should stay when it's brought into idle mode. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0x3F: 63 0.25us; other value: forbidden.	0x5
12:8	RW	ls_idle_to_tim	low sys NOC power controller idle timeout value. This is the max allowed time from assertion of idleAck to assertion of idle. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0x1F: 31 0.25us; other value: forbidden.	0x8
4:0	RW	ls_ack_to_tim	low sys NOC power controller idleAck timeout value. This is the max allowed time from assertion of idleReq to assertion of idleAck. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0x1F: 31 0.25us; other values: forbidden.	0x10

Is_PWR_LPI_CTL

Offset=0x128, Total Reset Value: 0x0000_0000, Write_enable=yes

Bits	Access	Name	Descriptions	Reset
4	WO	ls_pwr_mode	0: power off, 1: clock off	0x0
3	RWSC	ls_lpi_exit_idle	Is NOC power controller exit IDLE trigger. This is a write-only self-clearing bit. 0x1: request NOC logic in this power domain exiting idle state; 0x0: no operation.	0x0
2	RWSC	ls_lpi_go_idle	Is NOC power controller go IDLE trigger. This is a write-only self-clearing bit. 0x1: request NOC logic in this power domain entering idle state; 0x0: no operation.	0x0

Bits	Access	Name	Descriptions	Reset
1	RWSC	ls_pwr_up_req	Is power domain go POWER-ON trigger. This bit will be cleared by a hardware ack signal automatically. 0x1: request the power domain entering PowerOn state; 0x0: no operation.	0x0
0	RWSC	ls_pwr_off_req	Is power domain go POWER-OFF trigger. This bit will be cleared by a hardware ack signal automatically. 0x1: request the power domain entering idle state; 0x0: no operation.	0x0

ls_LPI_STAT

Offset=0x12c, Total Reset Value: 0x0000_000a, Write_enable=no

Bits	Access	Name	Descriptions	Reset
3	RO	ls_lpi_work	Is NOC power controller is in WORK state. 0x1: NOC logics in this power domain is in mission state; 0x0: no change.	0x1
2	RO	ls_lpi_idle	Is NOC power controller is in IDLE state. 0x1: NOC logics in this power domain is in idle state; 0x0: no change.	0x0
1	RO	ls_pcu_pwrup	Is power domain is currently in POWER-UP state. 0x1: the power domain is in power-up state; 0x0: no change.	0x1
0	RO	ls_pcu_pwroff	Is power domain is currently in POWER-OFF state. 0x1: the power domain is in power-off state; 0x0: no change.	0x0

sec_PWR_TIM

Offset=0x130, Total Reset Value: 0x0008_0c05, Write_enable=no

Bits	Access	Name	Descriptions	Reset
19:12	RW	sec_pwr_su_tim	PD_sec power switch control setup time. This is the time interval from changing power switch control signal to power status reach stable. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0xFF: 255 0.25us; 0x0: forbidden.	0x80
11:4	RW	sec_pd_hd_tim	PD_sec power-off mode least resting time. This is the least time that the power domain need to stay after entering power-off mode. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0xFF: 255 0.25us; 0x0: forbidden.	0xC0
3:0	RW	sec_iso_su_tim	PD_sec ISO cell control setup time. This is the time interval from changing ISO cell control signal to ISO cell truly active or deactive. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0xE: 14 0.25us; 0xF: 15 * 0.25us; 0x0: forbidden.	0x5

sec_LPI_TIM

Offset=0x134, Total Reset Value: 0x0005_0810, Write_enable=no

Bits	Access	Name	Descriptions	Reset
21:16	RW	sec_idle_hd_tim	sec NOC power controller in idle min time. This is the min time that NOC power controller should stay when it's brought into idle mode. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0x3F: 63 0.25us; other value: forbidden.	0x5
12:8	RW	sec_idle_to_tim	sec NOC power controller idle timeout value. This is the max allowed time from assertion of idleAck to assertion of idle. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0x1F: 31 0.25us; other value: forbidden.	0x8
4:0	RW	sec_ack_to_tim	sec NOC power controller idleAck timeout value. This is the max allowed time from assertion of idleReq to assertion of idleAck. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0x1F: 31 0.25us; other values: forbidden.	0x10

sec_PWR_LPI_CTL

Offset=0x138, Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
4	WO	sec_pwr_mode	0: power off, 1: clock off	0x0
3	RWSC	sec_lpi_exit_idle	sec NOC power controller exit IDLE trigger. This is a write-only self-clearing bit. 0x1: request NOC logic in this power domain exiting idle state; 0x0: no operation.	0x0
2	RWSC	sec_lpi_go_idle	sec NOC power controller go IDLE trigger. This is a write-only self-clearing bit. 0x1: request NOC logic in this power domain entering idle state; 0x0: no operation.	0x0
1	RWSC	sec_pwr_up_req	sec power domain go POWER-ON trigger. This bit will be cleared by a hardware ack signal automatically. 0x1: request the power domain entering PowerOn state; 0x0: no operation.	0x0
0	RWSC	sec_pwr_off_req	sec power domain go POWER-OFF trigger. This bit will be cleared by a hardware ack signal automatically. 0x1: request the power domain entering idle state; 0x0: no operation.	0x0

sec_PWR_LPI_STAT

Offset=0x13c, Total Reset Value: 0x0000_000a, Write_enable=no

Bits	Access	Name	Descriptions	Reset

Bits	Access	Name	Descriptions	Reset
3	RO	sec_lpi_work	sec NOC power controller is in WORK state. 0x1: NOC logics in this power domain is in mission state; 0x0: no change.	0x1
2	RO	sec_lpi_idle	sec NOC power controller is in IDLE state. 0x1: NOC logics in this power domain is in idle state; 0x0: no change.	0x0
1	RO	sec_pcu_pwrup	sec power domain is currently in POWER-UP state. 0x1: the power domain is in power-up state; 0x0: no change.	0x1
0	RO	sec_pcu_pwroff	sec power domain is currently in POWER-OFF state. 0x1: the power domain is in power-off state; 0x0: no change.	0x0

ISP_PWR_TIM

Offset=0x140, Total Reset Value: 0x0008_0c05, Write_enable=no

Bits	Access	Name	Descriptions	Reset
19:12	RW	isp_pwr_su_tim	PD_ISP power switch control setup time. This is the time interval from changing power switch control signal to power status reach stable. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0xFF: 255 0.25us; 0x0: forbidden.	0x80
11:4	RW	isp_pd_hd_tim	PD_ISP power-off mode least resting time. This is the least time that the power domain need to stay after entering power-off mode. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0xFF: 255 0.25us; 0x0: forbidden.	0xC0
3:0	RW	isp_iso_su_tim	PD_ISP ISO cell control setup time. This is the time interval from changing ISO cell control signal to ISO cell truly active or deactive. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0xE: 14 0.25us; 0xF: 15 * 0.25us; 0x0: forbidden.	0x5

ISP_LPI_TIM

Offset=0x144, Total Reset Value: 0x0005_0810, Write_enable=no

Bits	Access	Name	Descriptions	Reset
21:16	RW	isp_idle_hd_tim	ISP NOC power controller in idle min time. This is the min time that NOC power controller should stay when it's brought into idle mode. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0x3F: 63 0.25us; other value: forbidden.	0x5
12:8	RW	isp_idle_to_tim	ISP NOC power controller idle timeout value. This is the max allowed time from assertion of idleAck to assertion of idle. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0x1F: 31 0.25us; other value: forbidden.	0x8

Bits	Access	Name	Descriptions	Reset
4:0	RW	isp_ack_to_tim	ISP NOC power controller idleAck timeout value. This is the max allowed time from assertion of idleReq to assertion of idleAck. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0x1F: 31 0.25us; other values: forbidden.	0x10

ISP_PWR_LPI_CTL

Offset=0x148, Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
4	WO	isp_pwr_mode	0: power off, 1: clock off	0x0
3	RWSC	isp_lpi_exit_idle	ISP NOC power controller exit IDLE trigger. This is a write-only self-clearing bit. 0x1: request NOC logic in this power domain exiting idle state; 0x0: no operation.	0x0
2	RWSC	isp_lpi_go_idle	ISP NOC power controller go IDLE trigger. This is a write-only self-clearing bit. 0x1: request NOC logic in this power domain entering idle state; 0x0: no operation.	0x0
1	RWSC	isp_pwr_up_req	ISP power domain go POWER-ON trigger. This bit will be cleared by a hardware ack signal automatically. 0x1: request the power domain entering PowerOn state; 0x0: no operation.	0x0
0	RWSC	isp_pwr_off_req	ISP power domain go POWER-OFF trigger. This bit will be cleared by a hardware ack signal automatically. 0x1: request the power domain entering idle state; 0x0: no operation.	0x0

ISP_PWR_LPI_STAT

Offset=0x14c, Total Reset Value: 0x0000_000a, Write_enable=no

Bits	Access	Name	Descriptions	Reset
3	RO	isp_lpi_work	ISP NOC power controller is in WORK state. 0x1: NOC logics in this power domain is in mission state; 0x0: no change.	0x1
2	RO	isp_lpi_idle	ISP NOC power controller is in IDLE state. 0x1: NOC logics in this power domain is in idle state; 0x0: no change.	0x0
1	RO	isp_pcu_pwrup	ISP power domain is currently in POWER-UP state. 0x1: the power domain is in power-up state; 0x0: no change.	0x1
0	RO	isp_pcu_pwroff	ISP power domain is currently in POWER-OFF state. 0x1: the power domain is in power-off state; 0x0: no change.	0x0

PMU_PWR_TIM

Offset=0x150, Total Reset Value: 0x0008_0c05, Write_enable=no

Bits	Access	Name	Descriptions	Reset
19:12	RW	pmu_pwr_su_tim	PMU power switch control setup time. This is the time interval from changing power switch control signal to power status reach stable. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0xFF: 255 0.25us; 0x0: forbidden.	0x80
11:4	RW	pmu_pd_hd_tim	PMU power-off mode least resting time. This is the least time that the power domain need to stay after entering power-off mode. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0xFF: 255 0.25us; 0x0: forbidden.	0xC0
3:0	RW	pmu_iso_su_tim	PMU ISO cell control setup time. This is the time interval from changing ISO cell control signal to ISO cell truly active or deactive. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0xE: 14 0.25us; 0xF: 15 * 0.25us; 0x0: forbidden.	0x5

PMU_LPI_TIM

Offset=0x154, Total Reset Value: 0x0005_0810, Write_enable=no

Bits	Access	Name	Descriptions	Reset
21:16	RW	pmu_idle_hd_tim	PMU NOC power controller in idle min time. This is the min time that NOC power controller should stay when it's brought into idle mode. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0x3F: 63 0.25us; other value: forbidden.	0x5
12:8	RW	pmu_idle_to_tim	PMU NOC power controller idle timeout value. This is the max allowed time from assertion of idleAck to assertion of idle. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0x1F: 31 0.25us; other value: forbidden.	0x8
4:0	RW	pmu_ack_to_tim	PMU NOC power controller idleAck timeout value. This is the max allowed time from assertion of idleReq to assertion of idleAck. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0x1F: 31 0.25us; other values: forbidden.	0x10

PMU_PWR_LPI_CTL

Offset=0x158, Total Reset Value: 0x0000_0020, Write_enable=no

Bits	Access	Name	Descriptions	Reset
5	RW	pmu_iso_en	1: pmu isolation enable.	0x1
4	WO	pmu_pwr_mode	0: power off, 1: clock off	0x0
3	RWSC	pmu_lpi_exit_idle	PMU NOC power controller exit IDLE trigger. This is a write-only self-clearing bit. 0x1: request NOC logic in this power domain exiting idle state; 0x0: no operation.	0x0

Bits	Access	Name	Descriptions	Reset
2	RWSC	pmu_lpi_go_idle	PMU NOC power controller go IDLE trigger. This is a write-only self-clearing bit. 0x1: request NOC logic in this power domain entering idle state; 0x0: no operation.	0x0
1	RWSC	pmu_pwr_up_req	PMU power domain go POWER-ON trigger. This bit will be cleared by a hardware ack signal automatically. 0x1: request the power domain entering PowerOn state; 0x0: no operation.	0x0
0	RWSC	pmu_pwr_off_req	PMU power domain go POWER-OFF trigger. This bit will be cleared by a hardware ack signal automatically. 0x1: request the power domain entering idle state; 0x0: no operation.	0x0

PMU_PWR_LPI_STAT

Offset=0x15c, Total Reset Value: 0x0000_000a, Write_enable=no

Bits	Access	Name	Descriptions	Reset
3	RO	pmu_lpi_work	PMU NOC power controller is in WORK state. 0x1: NOC logics in this power domain is in mission state; 0x0: no change.	0x1
2	RO	pmu_lpi_idle	PMU NOC power controller is in IDLE state. 0x1: NOC logics in this power domain is in idle state; 0x0: no change.	0x0
1	RO	pmu_pcu_pwrup	PMU power domain is currently in POWER-UP state. 0x1: the power domain is in power-up state; 0x0: no change.	0x1
0	RO	pmu_pcu_pwroff	PMU power domain is currently in POWER-OFF state. 0x1: the power domain is in power-off state; 0x0: no change.	0x0

REPAIR_STATUS

Offset=0x160, Total Reset Value: 0x0000_0007, Write_enable=no

Bits	Access	Name	Descriptions	Reset
2	RO	glb_repair_done	1: global memory repair done	0x0
1	RO	ai_repair_done	AI power domain repair done. 0x1: the power domain repair done; 0x0: no change.	0x0
0	RO	shrm_repair_done	Shrm power domain repair done. 0x1: the power domain repair done; 0x0: no change.	0x0

SRAM0_REPAIR_TIM

Offset=0x164, Total Reset Value: 0x01fa_01fa, Write_enable=no

Bits	Access	Name	Descriptions	Reset
24	RW	para_use_shrm_repair_tim	0x1: wait shrm_repair_tim; 0x0: wait shrm_repair_done.	0x1
23:16	RW	para_shrm_repair_tim	shrm power controller wait repair time. This is the min time that power controller should stay before remove reset. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0xFFF: 4095 0.25us; other value: forbidden.	0xFA
8	RW	para_use_ai_repair_tim	0x1: wait ai_repair_tim; 0x0: wait ai_repair_done.	0x1
7:0	RW	para_ai_repair_tim	AI power controller wait repair time. This is the min time that power controller should stay before remove reset. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0xFFF: 4095 0.25us; other value: forbidden.	0xFA

SSYS_CTL_GPIO_CTL

Offset=0x168, Total Reset Value: 0x0000_00e1, Write_enable=no

Bits	Access	Name	Descriptions	Reset
7	RW	gpio_int_bothedge	0: single edge sensitive, depend on gpio_inttype_level and gpio_int_polarity 1: both edge sensitive	0x1
6	RW	gpio_debounce	0: no debounce 1: debounce	0x1
5	RW	gpio_int_polarity	0: falling edge or low level sensitive 1: rising edge or high level sensitive	0x1
4	RW	gpio_ls_sync	1: interrupt sync to pclk 0: interrupt dont sync to pclk	0x0
3	RW	gpio_porta_eoi	1: clear interrupt 0: No interrupt clear	0x0
2	RW	gpio_inttype_level	0: interrupt is level sensitive 1: interrupt is edge sensitive	0x0
1	RW	gpio_intmask	1: interrupt int mask	0x0
0	RW	gpio_inten	1: interrupt enable	0x1

SSYS_CTL_GPIO_EN0

Offset=0x170, Total Reset Value: 0x0000_0001, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31:0	RW	gpio_enL	bit map with gpio. 1: enable to bring power up	0x1

SSYS_CTL_GPIO_EN1

Offset=0x174, Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31:0	RW	gpio_enH	bit map with gpio. 1: enable to bring power up	0x0

CPU_REPAIR_TIM

Offset=0x178, Total Reset Value: 0x01fa_01fa, Write_enable=no

Bits	Access	Name	Descriptions	Reset
24	RW	para_use_cpu1_repair_tim	0x1: wait cpu1_repair_tim; 0x0: wait cpu1_repair_done.	0x1
23:16	RW	para_cpu1_repair_tim	CPU1 power controller wait repair time. This is the min time that power controller should stay before remove reset. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0xFFF: 4095 0.25us; other value: forbidden.	0xFA
8	RW	para_use_cpu0_repair_tim	0x1: wait cpu0_repair_tim; 0x0: wait cpu0_repair_done	0x1
7:0	RW	para_cpu0_repair_tim	CPU0 power controller wait repair time. This is the min time that power controller should stay before remove reset. 0x1: 1 0.25us; 0x2: 2 0.25us; 0x3: 3 0.25us; 0xFFF: 4095 0.25us; other value: forbidden.	0xFA

2.4 Interrupt System

K230 supports 208 interrupt sources. Table describes the mapping between interrupt bits and interrupt sources.

Table Mapping between the interrupt bits and interrupt sources

Interrupt Bit	Interrupt Source	Interrupt Bit	Interrupt Source
0	UART0	95	MB_CPU02CPU1_0
1	UART1	96	MB_CPU02CPU1_1
2	UART2	97	SYSCTL
3	UART3	98	ISP_VI
4	UART4	99	IPI_END_FRAME0
5	I2C0	100	IPI_END_FRAME_2IF0
6	I2C1	101	IPI_END_FRAME_3IF0
7	I2C2	102	IPI_END_FRAME1
8	I2C3	103	IPI_END_FRAME_2IF1
9	I2C4	104	IPI_END_FRAME_3IF1

Interrupt Bit	Interrupt Source	Interrupt Bit	Interrupt Source
10	PWM0	105	IPI_END_FRAME2
11	PWM1	106	IPI_END_FRAME_2IF2
12	PWM2	107	IPI_END_FRAME_3IF2
13	PWM3	108	ISP0
14	PWM4	109	ISP1
15	PWM5	110	ISP2
16	GPIO0_0	111	ISP_MI0
17	GPIO0_1	112	ISP_FE0
18	GPIO0_2	113	ISP_IRQ0
19	GPIO0_3	114	ISP_DWE
20	GPIO0_4	115	ISP_FE
21	GPIO0_5	116	VIDEO
22	GPIO0_6	117	DISP_VO
23	GPIO0_7	118	DISP_DSI
24	GPIO0_8	119	DISP_XAQ2
25	GPIO0_9	120	DWC_DDRPHY
26	GPIO0_10	121	DFI_ALERT_ERR
27	GPIO0_11	122	DECOMP_CTRL
28	GPIO0_12	123	PDMA
29	GPIO0_13	124	GSDMA
30	GPIO0_14	125	NONAI_2D
31	GPIO0_15	126	SD0
32	GPIO0_16	127	SD0_WAKEUP
33	GPIO0_17	128	SD1
34	GPIO0_18	129	SD1_WAKEUP
35	GPIO0_19	130	SSI0_TXE
36	GPIO0_20	131	SSI0_TXO
37	GPIO0_21	132	SSI0_RXF
38	GPIO0_22	133	SSI0_RXO
39	GPIO0_23	134	SSI0_TXU
40	GPIO0_24	135	SSI0_RXU
41	GPIO0_25	136	SSI0_MST
42	GPIO0_26	137	SSI0_DONE
43	GPIO0_27	138	SSI0_AXIE

Interrupt Bit	Interrupt Source	Interrupt Bit	Interrupt Source
44	GPIO0_28	139	SSI1_TXE
45	GPIO0_29	140	SSI1_TXO
46	GPIO0_30	141	SSI1_RXF
47	GPIO0_31	142	SSI1_RXO
48	GPIO1_0	143	SSI1_TXU
49	GPIO1_1	144	SSI1_RXU
50	GPIO1_2	145	SSI1_MST
51	GPIO1_3	146	SSI1_DONE
52	GPIO1_4	147	SSI1_AXIE
53	GPIO1_5	148	SSI2_TXE
54	GPIO1_6	149	SSI2_TXO
55	GPIO1_7	150	SSI2_RXF
56	GPIO1_8	151	SSI2_RXO
57	GPIO1_9	152	SSI2_TXU
58	GPIO1_10	153	SSI2_RXU
59	GPIO1_11	154	SSI2_MST
60	GPIO1_12	155	SSI2_DONE
61	GPIO1_13	156	SSI2_AXIE
62	GPIO1_14	157	OTG0
63	GPIO1_15	158	OTG1
64	GPIO1_16	159	PMU
65	GPIO1_17	160	OBS_MAINFAULT0
66	GPIO1_18	161	OBS_MAINFAULT1
67	GPIO1_19	162	MCTL_PROBE0
68	GPIO1_20	163	MCTL_PROBE1
69	GPIO1_21	164	MCTL_PROBE2
70	GPIO1_22	165	MCTL_PROBE3
71	GPIO1_23	166	MCTL_PROBE4
72	GPIO1_24	167	SRAM_PROBE0
73	GPIO1_25	168	SRAM_PROBE1
74	GPIO1_26	169	SEC
75	GPIO1_27	170	DPU
76	GPIO1_28	171	DPU_INT_TYPE0
77	GPIO1_29	172	DPU_INT_TYPE1

Interrupt Bit	Interrupt Source	Interrupt Bit	Interrupt Source
78	GPIO1_30	173	GNNE
79	GPIO1_31	174	FFT
80	AUDIO	175	AI_2D
81	JAMLINK0	176	CPU0/CPU1_PAR_VIO
82	JAMLINK1	177	ADC0
83	JAMLINK2	178	ADC1
84	JAMLINK3	179	ADC2
85	TIMER0	180	PDMA_CHANNEL1
86	TIMER1	181	PDMA_CHANNEL2
87	TIMER2	182	PDMA_CHANNEL3
88	TIMER3	183	PDMA_CHANNEL4
89	TIMER4	184	PDMA_CHANNEL5
90	TIMER5	185	PDMA_CHANNEL6
91	WDT0	186	PDMA_CHANNEL7
92	WDT1	187	PDMA_ALL_CHANNEL
93	MB_CPU12CPU0_0	188	ISP_VSE
94	MB_CPU12CPU0_1	189~207	Reserved

2.5 DMA Controller

2.5.1 PDMA

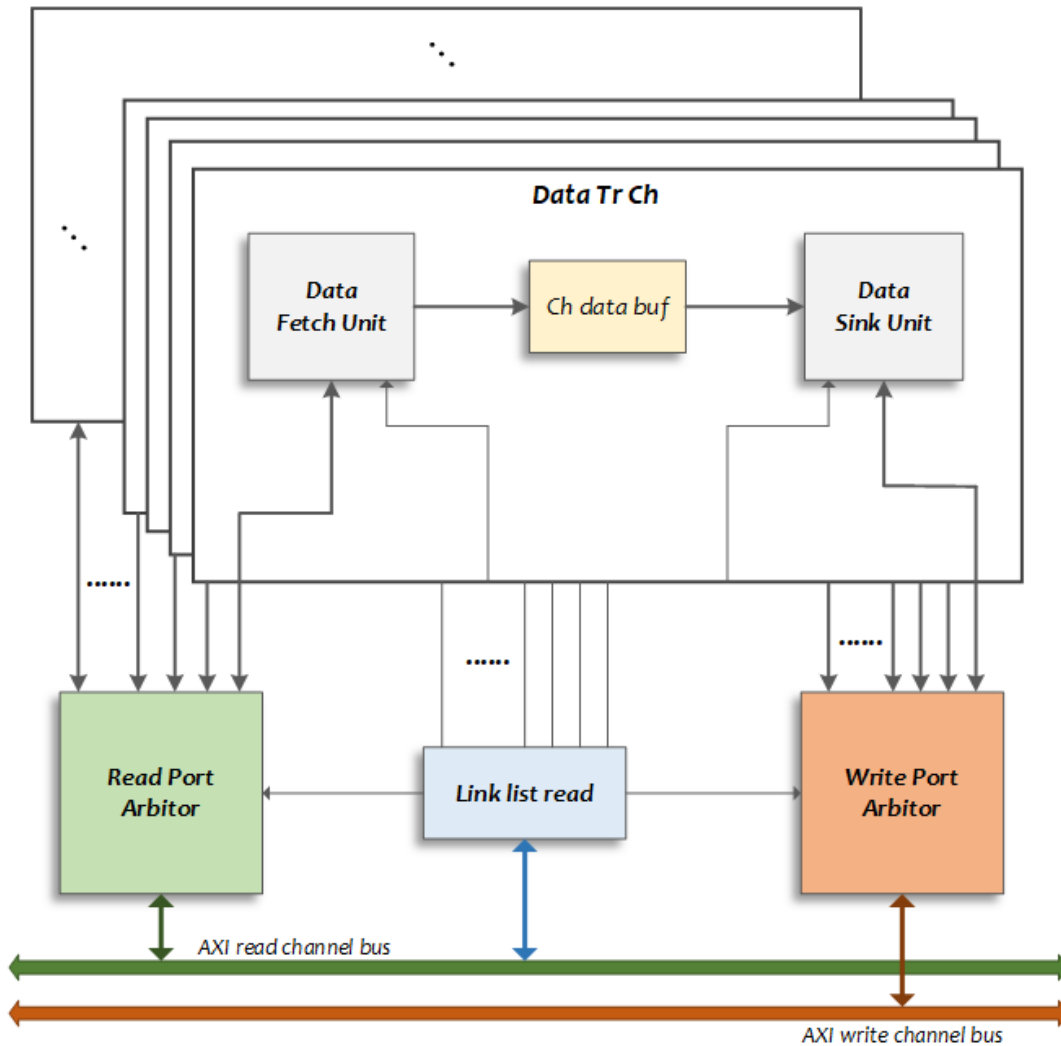
2.5.1.1 Overview

PDMA (Peripheral Direct Memory Access) supports the data transfer between peripheral ports and DDR/SRAM. Key features:

- PDMA provides 8 channels to transfer different transactions, with 64-byte data buffer per channel.
- It supports up to 35 peripheral ports through channel configuration.
- Low power IDLE state of each channel can be set by shutting down corresponding clock.
- A 64-bit AXI4 master is used to transfer different channel transactions.
- The priority of each transaction is configured with each channel. Transactions with high priority gets high chance to be transferred.
- Peripheral data address can only be accessed in 4 bytes aligned with strobe signal to indicate the lower 1/2/4 byte(s) are used, and only fixed address is supported.

2.5.1.2 Block Diagram

Figure 2-5-1 pdma block diagram



2.5.1.3 Function Description

- Peri_dma_tr_ch is a data transferring module. It consists of 4 sub-modules, which are peri_dma_ch_reg, peri_dma_data_fetch, peri_dma_data_sink, and peri_dma_data_buf. Besides it has synchronizers and clock gating units.
- peri_dma_data_fetch sends read requests of different channels to an arbiter. The arbiter sends out read transactions of the selected channel through AXI master to get data from the source address. The read data is directly written to the peri_dma_ch_buf module inside the channel selected by arbiter.
- peri_dma_data_sink sends out the reading data to the destination device.
- peri_dma_data_buf sends out requests to the arbiter. After getting acknowledgement, it reads out the buffered data from peri_dma_ch_buf of the selected channel, and writes them to the destination device through AXI.
- peri_dma_llr is the read from memory configuration interface module of the peripheral DMA controller.
- peri_dma_rd_arbitor is an arbitration unit to handle different channel requests from the DMA controller. 16 data transfer channels can request to fetch data at the same time. This arbiter determines which channel can get the permission of AXI master to fetch data. Software can set the request priority for the 16 data channels. Higher request priority channel gets higher probability to get the AXI master reading permission.
- peri_dma_wr_arbitor is the arbitration unit to handle different channel requests from the DMA controller. 16 data transfer channels request to store data at the same time. This arbiter determines which channel can get the permission of AXI master to store data. Software can set the request priority for the 16 data channels. Higher request priority channel gets higher probability to get the AXI master writing permission.

2.5.1.4 Programming Guideline

- Query pdma_channel_enable register, see which channel is not occupied, and then write 1.

- Query register CHx_Chx of STAT_Busy status bit, to ensure that the current DMA channel is in IDLE status.
- Write access register CHx_PERI_DEV_SEL, select which peripheral interface module the current DMA transmission channel wants to use for data transmission.
- Configuration register CHx_CFG0, set the transmission mode, source device type, channel priority, effective data width of peripheral modules and other parameters.
- Write the linked list information to the memory, and write the address of the linked list memory to CHx_LLT_SADDR register.
- If the selected linked list is paused after completion, set pause to the current descriptor.
- Write register DMA_INT_MASK, configure the interrupt enable bit corresponding to the current DMA channel as '1'.
- Configuration CHx_CTL register's to chx_start parameter writes '1' to start the transmission of the current DMA channel.
- Wait for the data transmission of the current channel to complete. When an interrupt is detected, immediately change the CHx_CTL register's chx_start parameter writes '1' to start the next DMA channel data transmission.
- After transmission, pdma enable register's corresponding bit is cleared.

2.5.1.5 register summary

Offset Address	Register	Description
0x0	PDMA_CH_EN	channel enable register
0x4	PDMA_INT_MASK	Interrupt bit mask register
0x8	PDMA_INT_STATUS	Interrupt status register
0x20	CH0_CTL	pdma control register
0x24	CH0_STATUS	pdma status register
0x28	CH0_CFG	pdma config register
0x2c	CH0_LLT_SADDR	dma link list address register
0x30	CH0_CURRENT_SIZE	transmitted length of the current channel register
0x34	CH0_CURRENT_LLT	Linked list address being processed register
0x38	CH0_CURRENT_SADDR	source address of the current transfer register
0x3c	CH0_CURRENT_DADDR	destination address of the current transfer register
0x40	CH1_CTL	pdma control register
0x44	CH1_STATUS	pdma status register
0x48	CH1_CFG	pdma config register
0x4c	CH1_LLT_SADDR	dma link list address register
0x50	CH1_CURRENT_SIZE	transmitted length of the current channel register
0x54	CH1_CURRENT_LLT	Linked list address being processed register
0x58	CH1_CURRENT_SADDR	source address of the current transfer register
0x5c	CH1_CURRENT_DADDR	destination address of the current transfer register
0x60	CH2_CTL	pdma control register
0x64	CH2_STATUS	pdma status register
0x68	CH2_CFG	pdma config register

Offset Address	Register	Description
0x6c	CH2_LLT_SADDR	dma link list address register
0x70	CH2_CURRENT_SIZE	transmitted length of the current channel register
0x74	CH2_CURRENT_LLT	Linked list address being processed register
0x78	CH2_CURRENT_SADDR	source address of the current transfer register
0x7c	CH2_CURRENT_DADDR	destination address of the current transfer register
0x80	[CH3_CTL] (#CH3_CTL)	pdma control register
0x84	CH3_STATUS	pdma status register
0x88	CH3_CFG	pdma config register
0x8c	CH3_LLT_SADDR	dma link list address register
0x90	CH3_CURRENT_SIZE	transmitted length of the current channel register
0x94	CH3_CURRENT_LLT	Linked list address being processed register
0x98	CH3_CURRENT_SADDR	source address of the current transfer register
0x9c	CH3_CURRENT_DADDR	destination address of the current transfer register
0xa0	CH4_CTL	pdma control register
0xa4	CH4_STATUS	pdma status register
0xa8	CH4_CFG	pdma config register
0xac	CH4_LLT_SADDR	dma link list address register
0xb0	CH4_CURRENT_SIZE	transmitted length of the current channel register
0xb4	CH4_CURRENT_LLT	Linked list address being processed register
0xb8	CH4_CURRENT_SADDR	source address of the current transfer register
0xbc	CH4_CURRENT_DADDR	destination address of the current transfer register
0xc0	CH5_CTL	pdma control register
0xc4	CH5_STATUS	pdma status register
0xc8	CH5_CFG	pdma config register
0xcc	CH5_LLT_SADDR	dma link list address register
0xd0	CH5_CURRENT_SIZE	transmitted length of the current channel register
0xd4	CH5_CURRENT_LLT	Linked list address being processed register
0xd8	CH5_CURRENT_SADDR	source address of the current transfer register
0xdc	CH5_CURRENT_DADDR	destination address of the current transfer register
0xe0	CH6_CTL	pdma control register
0xe4	CH6_STATUS	pdma status register
0xe8	CH6_CFG	pdma config register
0xec	CH6_LLT_SADDR	dma link list address register
0xf0	CH6_CURRENT_SIZE	transmitted length of the current channel register

Offset Address	Register	Description
0xf4	CH6_CURRENT_LLT	Linked list address being processed register
0xf8	CH6_CURRENT_SADDR	source address of the current transfer register
0xfc	CH6_CURRENT_DADDR	destination address of the current transfer register
0x100	CH7_CTL	pdma control register
0x104	CH7_STATUS	pdma status register
0x108	CH7_CFG	pdma config register
0x10c	CH7_LLT_SADDR	dma link list address register
0x110	CH7_CURRENT_SIZE	transmitted length of the current channel register
0x114	CH7_CURRENT_LLT	Linked list address being processed register
0x118	CH7_CURRENT_SADDR	source address of the current transfer register
0x11c	CH7_CURRENT_DADDR	destination address of the current transfer register

2.5.1.6 Register Description

PDMA_CH_EN

PDMA_CH_EN is a channel enable register .

Offset Address : 0x0 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
7:0	pdma_channel_enable	0x0	RW	0x1:Open channel 0x0: close channel

PDMA_INT_MASK

PDMA_INT_MASK is a Interrupt bit mask register.

Offset Address : 0x4 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
31:24	chx_dev_tout_mask	0x0	RW	Shield pdma timeout interrupt
23:16	chx_llt_pause_mask	0x0	RW	Shield pdma transfer pause interrupt
15:8	chx_llt_item_int_mask	0x0	RW	Shield pdma node transmission completed interrupt
7:0	chx_tr_done_mask	0x0	RW	Shield pdma transmission completed interrupt bit0

PDMA_INT_STATUS PDMA_INT_STATUS is a Interrupt status register.

Offset Address : 0x8 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
31:24	chx_dev_tout_int	0x0	RW	pdma timeout interrupt
23:16	chx_llt_pause_int	0x0	RW	pdma transfer pause interrupt
15:8	chx_llt_item_int	0x0	RW	pdma node transmission completed interrupt
7:0	chx_tr_done_int	0x0	RW	pdma transmission completed interrupt

CH0_CTL CH0_CTL is a pdma control register.

Offset Address : 0x20 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
2	ch0_resume	0x0	WO	Control the pdma channel to resume transmission
1	ch0_stop	0x0	WO	Control the pdma channel to stop transmission
0	ch0_start	0x0	WO	Control the pdma channel to start transmission

CH0_STATUS CH0_STATUS is a pdma status register.

Offset Address : 0x24 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
1	ch0_pause	0x0	RO	Whether the pdma is paused
0	ch0_busy	0x0	RO	Whether the pdma is in the transmission state

CH0_CFG CH0_CFG is a pdma config register.

Offset Address : 0x28 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
27:16	ch0_dev_tout	0x0	RW	DMA channel 0 peripheral device DataReq timeout counter initial value. When reading from or writing to peripheral device, a timeout counter runs while channel waiting for device DataReq. If DataReq keep deasserting when this counter down counts to 0, current data transfer session will be stopped. 0x8: 8 cycles; 0x9: 9 cycles; 0xA: 10 cycles; 0xB: 11 cycles; 0xFFF: 4095 timer cycles; others: forbidden.
15:12	ch0_priority	0x0	RW	Transfer channel 0 priority configuration bits. Valid value can be 1 to 15. A greater value means higher priority.
11:8	ch0_dev_blen	0x0	RW	The burst length of the transaction used to read data from or write data to peripheral devices. 0x0: single burst; 0x1: burst 2; 0x2: burst 3 ; 0xF: burst 16.
5:4	ch0_dat_endian	0x0	RW	This bit field is used to specify the byte order of transmitted data. 0x0: Default configuration, no adjustment; 0x1: The transmission length is 2 bytes,and endian swap; 0x2: The transmission length is 4 bytes,and endian swap; 0x3: The transmission length is 8 bytes,and endian swap;

bits	Name	Reset	Access	Descriptions
2:1	ch0_dev_hsize	0x0	RW	The bit width of a single data word provided by or sent to peripheral device. 0x0: one byte; 0x1: two bytes; 0x2: four bytes.
0	ch0_src_type	0x0	RW	Transfer channel data source side device type. 0x0: system memory or SRAM area with continous access address; 0x1: peripheral device FIFO mapped to a fixed register address.

CH0_LL_T_SADDR CH0_LL_T_SADDR is a channel link list address register

Offset Address : 0x2c Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
31:0	ch0_llt_saddr	0x0	RW	This parameter is used to specify the starting entry address of the linked list block allocated to the DMA channel.

CH0_CURRENT_LL_T CH0_CURRENT_LL_T is a Linked list address being processed register .

Offset Address : 0x30 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
31:0	ch0_next_linklist	0x0	RW	The next linked list address of the current working pdma, used for debugging.

CH0_CURRENT_SIZE CH0_CURRENT_SIZE is a The transmitted length of the current channel register .

Offset Address : 0x34 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
31:0	ch0_current_size	0x0	RW	The total length of data transported by pdma, used for debugging.

CH0_CURRENT_SADDR CH0_CURRENT_SADDR is a source address of the current transfer register .

Offset Address : 0x38 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
31:0	ch0_current_saddr	0x0	RW	The source address of the current read request sent by the pdma, used for debugging

CH0_CURRENT_DADDR CH0_CURRENT_DADDR is a destination address of the current transfer register .

Offset Address : 0x3c Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
31:0	ch0_current_daddr	0x0	RW	The target address of the current write request sent by the pdma, which is used for debugging.

CH1_CTL CH1_CTL is a pdma control register.

Offset Address : 0x40 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions

bits	Name	Reset	Access	Descriptions
2	ch1_resume	0x0	WO	Control the pdma channel to resume transmission
1	ch1_stop	0x0	WO	Control the pdma channel to stop transmission
0	ch1_start	0x0	WO	Control the pdma channel to start transmission

CH1_STATUS CH1_STATUS is a pdma status register.

Offset Address : 0x44 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
1	ch1_pause	0x0	RO	Whether the pdma is paused
0	ch1_busy	0x0	RO	Whether the pdma is in the transmission state

CH1_CFG |CH1_CFG is a pdma config register.

|Offset Address : 0x48 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
27:16	ch1_dev_tout	0x0	RW	DMA channel 0 peripheral device DataReq timeout counter initial value. When reading from or writing to peripheral device, a timeout counter runs while channel waiting for device DataReq. If DataReq keep deasserting when this counter down counts to 0, current data transfer session will be stopped. 0x8: 8 cycles; 0x9: 9 cycles; 0xA: 10 cycles; 0xB: 11 cycles; 0xFFF: 4095 timer cycles; others: forbidden.
15:12	ch1_priority	0x0	RW	Transfer channel 0 priority configuration bits. Valid value can be 1 to 15. A greater value means higher priority.
11:8	ch1_dev_blen	0x0	RW	The burst length of the transaction used to read data from or write data to peripheral devices. 0x0: single burst; 0x1: burst 2; 0x2: burst 3 ; 0xF: burst 16.
5:4	ch1_dat_endian	0x0	RW	This bit field is used to specify the byte order of transmitted data. 0x0: Default configuration, no adjustment; 0x1: The transmission length is 2 bytes,and endian swap; 0x2: The transmission length is 4 bytes,and endian swap; 0x3: The transmission length is 8 bytes,and endian swap;

bits	Name	Reset	Access	Descriptions
2:1	ch1_dev_hsize	0x0	RW	The bit width of a single data word provided by or sent to peripheral device. 0x0: one byte; 0x1: two bytes; 0x2: four bytes.
0	ch1_src_type	0x0	RW	Transfer channel data source side device type. 0x0: system memory or SRAM area with continous access address; 0x1: peripheral device FIFO mapped to a fixed register address.

CH1_LL_T_SADDR CH1_LL_T_SADDR is a channel link list address register .

Offset Address : 0x4c Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
31:0	ch1_llt_saddr	0x0	RW	This parameter is used to specify the starting entry address of the linked list block allocated to the DMA channel.

CH1_CURRENT_LL_T CH1_CURRENT_LL_T is a Linked list address being processed register .

Offset Address : 0x50 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
31:0	ch1_next_linklist	0x0	RW	The next linked list address of the current working pdma, used for debugging.

CH1_CURRENT_SIZE CH1_CURRENT_SIZE is a The transmitted length of the current channel register .

Offset Address : 0x54 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
31:0	ch1_current_size	0x0	RW	The total length of data transported by pdma, used for debugging.

CH1_CURRENT_SADDR CH1_CURRENT_SADDR is a source address of the current transfer register .

Offset Address : 0x58 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
31:0	ch1_current_saddr	0x0	RW	The source address of the current read request sent by the pdma, used for debugging

CH1_CURRENT_DADDR CH1_CURRENT_DADDR is a destination address of the current transfer register .

Offset Address : 0x5c Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
31:0	ch1_current_daddr	0x0	RW	The target address of the current write request sent by the pdma, which is used for debugging.

CH2_CTL CH2_CTL is a pdma control register.

Offset Address : 0x60 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions

bits	Name	Reset	Access	Descriptions
2	ch2_resume	0x0	WO	Control the pdma channel to resume transmission
1	ch2_stop	0x0	WO	Control the pdma channel to stop transmission
0	ch2_start	0x0	WO	Control the pdma channel to start transmission

CH2_STATUS CH2_STATUS is a pdma status register.

Offset Address : 0x64 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
1	ch2_pause	0x0	RO	Whether the pdma is paused
0	ch2_busy	0x0	RO	Whether the pdma is in the transmission state

CH2_CFG CH2_CFG is a pdma config register.

Offset Address : 0x68 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
27:16	ch2_dev_tout	0x0	RW	DMA channel 0 peripheral device DataReq timeout counter initial value. When reading from or writing to peripheral device, a timeout counter runs while channel waiting for device DataReq. If DataReq keep deasserting when this counter down counts to 0, current data transfer session will be stopped. 0x8: 8 cycles; 0x9: 9 cycles; 0xA: 10 cycles; 0xB: 11 cycles; 0xFFF: 4095 timer cycles; others: forbidd.
15:12	ch2_priority	0x0	RW	Transfer channel 0 priority configuration bits. Valid value can be 1 to 15. A greater value means higher priority.
11:8	ch2_dev_blen	0x0	RW	The burst length of the transaction used to read data from or write data to peripheral devices. 0x0: single burst; 0x1: burst 2; 0x2: burst 3 ; 0xF: burst 16.
5:4	ch2_dat_endian	0x0	RW	This bit field is used to specify the byte order of transmitted data. 0x0: Default configuration, no adjustment; 0x1: The transmission length is 2 bytes,and endian swap; 0x2: The transmission length is 4 bytes,and endian swap; 0x3: The transmission length is 8 bytes,and endian swap;

bits	Name	Reset	Access	Descriptions
2:1	ch2_dev_hsize	0x0	RW	The bit width of a single data word provided by or sent to peripheral device. 0x0: one byte; 0x1: two bytes; 0x2: four bytes.
0	ch2_src_type	0x0	RW	Transfer channel data source side device type. 0x0: system memory or SRAM area with continous access address; 0x1: peripheral device FIFO mapped to a fixed register address.

CH2_LL_T_SADDR CH2_LL_T_SADDR is a channel link list address register .

Offset Address : 0x6c Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
31:0	ch2_llt_saddr	0x0	RW	This parameter is used to specify the starting entry address of the linked list block allocated to the DMA channel.

CH2_CURRENT_LL_T CH2_CURRENT_LL_T is a Linked list address being processed register .

Offset Address : 0x70 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
31:0	ch2_next_linklist	0x0	RW	The next linked list address of the current working pdma, used for debugging.

CH2_CURRENT_SIZE CH2_CURRENT_SIZE is a The transmitted length of the current channel register .

Offset Address : 0x74 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
31:0	ch2_current_size	0x0	RW	The total length of data transported by pdma, used for debugging.

CH2_CURRENT_SADDR CH2_CURRENT_SADDR is a source address of the current transfer register .

Offset Address : 0x78 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
31:0	ch2_current_saddr	0x0	RW	The source address of the current read request sent by the pdma, used for debugging

CH2_CURRENT_DADDR CH2_CURRENT_DADDR is a destination address of the current transfer register .

Offset Address : 0x7c Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
31:0	ch2_current_daddr	0x0	RW	The target address of the current write request sent by the pdma, which is used for debugging.

CH3_CTL CH3_CTL is a pdma control register.

Offset Address : 0x80 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions

bits	Name	Reset	Access	Descriptions
2	ch3_resume	0x0	WO	Control the pdma channel to resume transmission
1	ch3_stop	0x0	WO	Control the pdma channel to stop transmission
0	ch3_start	0x0	WO	Control the pdma channel to start transmission

CH3_STATUS CH3_STATUS is a pdma status register.

Offset Address : 0x84 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
1	ch3_pause	0x0	RO	Whether the pdma is paused
0	ch3_busy	0x0	RO	Whether the pdma is in the transmission state

CH3_CFG CH3_CFG is a pdma config register.

Offset Address : 0x84 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
27:16	ch3_dev_tout	0x0	RW	DMA channel 0 peripheral device DataReq timeout counter initial value. When reading from or writing to peripheral device, a timeout counter runs while channel waiting for device DataReq. If DataReq keep deasserting when this counter down counts to 0, current data transfer session will be stopped. 0x8: 8 cycles; 0x9: 9 cycles; 0xA: 10 cycles; 0xB: 11 cycles; 0xFFFF: 4095 timer cycles; others: forbidden.
15:12	ch3_priority	0x0	RW	Transfer channel 0 priority configuration bits. Valid value can be 1 to 15. A greater value means higher priority.
11:8	ch3_dev_blen	0x0	RW	The burst length of the transaction used to read data from or write data to peripheral devices. 0x0: single burst; 0x1: burst 2; 0x2: burst 3 ; 0xF: burst 16.
5:4	ch3_dat_endian	0x0	RW	This bit field is used to specify the byte order of transmitted data. 0x0: Default configuration, no adjustment; 0x1: The transmission length is 2 bytes,and endian swap; 0x2: The transmission length is 4 bytes,and endian swap; 0x3: The transmission length is 8 bytes,and endian swap;

bits	Name	Reset	Access	Descriptions
2:1	ch3_dev_hsize	0x0	RW	The bit width of a single data word provided by or sent to peripheral device. 0x0: one byte; 0x1: two bytes; 0x2: four bytes.
0	ch3_src_type	0x0	RW	Transfer channel data source side device type. 0x0: system memory or SRAM area with continous access address; 0x1: peripheral device FIFO mapped to a fixed register address.

CH3_LL_T_SADDR CH3_LL_T_SADDR is a channel link list address register .

Offset Address : 0x88 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
31:0	ch3_llt_saddr	0x0	RW	This parameter is used to specify the starting entry address of the linked list block allocated to the DMA channel.

CH3_CURRENT_LL_T CH3_CURRENT_LL_T is a Linked list address being processed register .

Offset Address : 0x8c Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
31:0	ch3_next_linklist	0x0	RW	The next linked list address of the current working pdma, used for debugging.

CH3_CURRENT_SIZE CH3_CURRENT_SIZE is a The transmitted length of the current channel register .

Offset Address : 0x90 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
31:0	ch3_current_size	0x0	RW	The total length of data transported by pdma, used for debugging.

CH3_CURRENT_SADDR CH3_CURRENT_SADDR is a source address of the current transfer register .

Offset Address : 0x94 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
31:0	ch3_current_saddr	0x0	RW	The source address of the current read request sent by the pdma, used for debugging

CH3_CURRENT_DADDR CH3_CURRENT_DADDR is a destination address of the current transfer register .

Offset Address : 0x98 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
31:0	ch3_current_daddr	0x0	RW	The target address of the current write request sent by the pdma, which is used for debugging.

CH4_CTL CH4_CTL is a pdma control register.

Offset Address : 0xa0 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions

bits	Name	Reset	Access	Descriptions
2	ch4_resume	0x0	WO	Control the pdma channel to resume transmission
1	ch4_stop	0x0	WO	Control the pdma channel to stop transmission
0	ch4_start	0x0	WO	Control the pdma channel to start transmission

CH4_STATUS CH4_STATUS is a pdma status register.

Offset Address : 0xa4 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
1	ch4_pause	0x0	RO	Whether the pdma is paused
0	ch4_busy	0x0	RO	Whether the pdma is in the transmission state

CH4_CFG CH4_CFG is a pdma config register.

Offset Address : 0xa8 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
27:16	ch4_dev_tout	0x0	RW	DMA channel 0 peripheral device DataReq timeout counter initial value. When reading from or writing to peripheral device, a timeout counter runs while channel waiting for device DataReq. If DataReq keep deasserting when this counter down counts to 0, current data transfer session will be stopped. 0x8: 8 cycles; 0x9: 9 cycles; 0xA: 10 cycles; 0xB: 11 cycles; 0xFFF: 4095 timer cycles; others: forbidden.
15:12	ch0_priority	0x0	RW	Transfer channel 0 priority configuration bits. Valid value can be 1 to 15. A greater value means higher priority.
11:8	ch4_dev_blen	0x0	RW	The burst length of the transaction used to read data from or write data to peripheral devices. 0x0: single burst; 0x1: burst 2; 0x2: burst 3 ; 0xF: burst 16.
5:4	ch4_dat_endian	0x0	RW	This bit field is used to specify the byte order of transmitted data. 0x0: Default configuration, no adjustment; 0x1: The transmission length is 2 bytes,and endian swap; 0x2: The transmission length is 4 bytes,and endian swap; 0x3: The transmission length is 8 bytes,and endian swap;

bits	Name	Reset	Access	Descriptions
2:1	ch4_dev_hsize	0x0	RW	The bit width of a single data word provided by or sent to peripheral device. 0x0: one byte; 0x1: two bytes; 0x2: four bytes.
0	ch4_src_type	0x0	RW	Transfer channel data source side device type. 0x0: system memory or SRAM area with continous access address; 0x1: peripheral device FIFO mapped to a fixed register address.

CH4_LL_T_SADDR CH4_LL_T_SADDR is a channel link list address register .

Offset Address : 0xac Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
31:0	ch4_llt_saddr	0x0	RW	This parameter is used to specify the starting entry address of the linked list block allocated to the DMA channel.

CH4_CURRENT_LL_T CH4_CURRENT_LL_T is a Linked list address being processed register .

Offset Address : 0xb0 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
31:0	ch4_next_linklist	0x0	RW	The next linked list address of the current working pdma, used for debugging.

CH4_CURRENT_SIZE CH4_CURRENT_SIZE is a The transmitted length of the current channel register .

Offset Address : 0xb4 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
31:0	ch4_current_size	0x0	RW	The total length of data transported by pdma, used for debugging.

CH4_CURRENT_SADDR CH4_CURRENT_SADDR is a source address of the current transfer register .

Offset Address : 0xb8 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
31:0	ch4_current_saddr	0x0	RW	The source address of the current read request sent by the pdma, used for debugging

CH4_CURRENT_DADDR CH4_CURRENT_DADDR is a destination address of the current transfer register .

Offset Address : 0xbc Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
31:0	ch4_current_daddr	0x0	RW	The target address of the current write request sent by the pdma, which is used for debugging.

CH5_CTL CH5_CTL is a pdma control register.

Offset Address : 0xc0 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions

bits	Name	Reset	Access	Descriptions
2	ch5_resume	0x0	WO	Control the pdma channel to resume transmission
1	ch5_stop	0x0	WO	Control the pdma channel to stop transmission
0	ch5_start	0x0	WO	Control the pdma channel to start transmission

CH5_STATUS CH5_STATUS is a pdma status register.

Offset Address : 0xc4 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
1	ch5_pause	0x0	RO	Whether the pdma is paused
0	ch5_busy	0x0	RO	Whether the pdma is in the transmission state

CH5_CFG CH5_CFG is a pdma config register.

Offset Address : 0xc8 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
27:16	ch5_dev_tout	0x0	RW	DMA channel 0 peripheral device DataReq timeout counter initial value. When reading from or writing to peripheral device, a timeout counter runs while channel waiting for device DataReq. If DataReq keep deasserting when this counter down counts to 0, current data transfer session will be stopped. 0x8: 8 cycles; 0x9: 9 cycles; 0xA: 10 cycles; 0xB: 11 cycles; 0xFFF: 4095 timer cycles; others: forbidden.
15:12	ch5_priority	0x0	RW	Transfer channel 0 priority configuration bits. Valid value can be 1 to 15. A greater value means higher priority.
11:8	ch5_dev_blen	0x0	RW	The burst length of the transaction used to read data from or write data to peripheral devices. 0x0: single burst; 0x1: burst 2; 0x2: burst 3 ; 0xF: burst 16.
5:4	ch5_dat_endian	0x0	RW	This bit field is used to specify the byte order of transmitted data. 0x0: Default configuration, no adjustment; 0x1: The transmission length is 2 bytes,and endian swap; 0x2: The transmission length is 4 bytes,and endian swap; 0x3: The transmission length is 8 bytes,and endian swap;

bits	Name	Reset	Access	Descriptions
2:1	ch5_dev_hsize	0x0	RW	The bit width of a single data word provided by or sent to peripheral device. 0x0: one byte; 0x1: two bytes; 0x2: four bytes.
0	ch5_src_type	0x0	RW	Transfer channel data source side device type. 0x0: system memory or SRAM area with continous access address; 0x1: peripheral device FIFO mapped to a fixed register address.

CH5_LL_T_SADDR CH5_LL_T_SADDR is a channel link list address register .

Offset Address : 0xcc Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
31:0	ch5_llt_saddr	0x0	RW	This parameter is used to specify the starting entry address of the linked list block allocated to the DMA channel.

CH5_CURRENT_LL_T CH5_CURRENT_LL_T is a Linked list address being processed register .

Offset Address : 0xd0 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
31:0	ch5_next_linklist	0x0	RW	The next linked list address of the current working pdma, used for debugging.

CH5_CURRENT_SIZE CH5_CURRENT_SIZE is a The transmitted length of the current channel register .

Offset Address : 0xd4 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
31:0	ch5_current_size	0x0	RW	The total length of data transported by pdma, used for debugging.

CH5_CURRENT_SADDR CH5_CURRENT_SADDR is a source address of the current transfer register .

Offset Address : 0xd8 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
31:0	ch5_current_saddr	0x0	RW	The source address of the current read request sent by the pdma, used for debugging

CH5_CURRENT_DADDR CH5_CURRENT_DADDR is a destination address of the current transfer register .

Offset Address : 0xdc Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
31:0	ch5_current_daddr	0x0	RW	The target address of the current write request sent by the pdma, which is used for debugging.

CH6_CTL CH6_CTL is a pdma control register.

Offset Address : 0xe0 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions

bits	Name	Reset	Access	Descriptions
2	ch6_resume	0x0	WO	Control the pdma channel to resume transmission
1	ch6_stop	0x0	WO	Control the pdma channel to stop transmission
0	ch6_start	0x0	WO	Control the pdma channel to start transmission

CH6_STATUS CH6_STATUS is a pdma status register.

Offset Address : 0xe4 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
1	ch6_pause	0x0	RO	Whether the pdma is paused
0	ch6_busy	0x0	RO	Whether the pdma is in the transmission state

CH6_CFG CH6_CFG is a pdma config register.

Offset Address : 0xe8 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
27:16	ch6_dev_tout	0x0	RW	DMA channel 0 peripheral device DataReq timeout counter initial value. When reading from or writing to peripheral device, a timeout counter runs while channel waiting for device DataReq. If DataReq keep deasserting when this counter down counts to 0, current data transfer session will be stopped. 0x8: 8 cycles; 0x9: 9 cycles; 0xA: 10 cycles; 0xB: 11 cycles; 0xFFF: 4095 timer cycles; others: forbidden.
15:12	ch0_priority	0x0	RW	Transfer channel 0 priority configuration bits. Valid value can be 1 to 15. A greater value means higher priority.
11:8	ch6_dev_blen	0x0	RW	The burst length of the transaction used to read data from or write data to peripheral devices. 0x0: single burst; 0x1: burst 2; 0x2: burst 3 ; 0xF: burst 16.
5:4	ch6_dat_endian	0x0	RW	This bit field is used to specify the byte order of transmitted data. 0x0: Default configuration, no adjustment; 0x1: The transmission length is 2 bytes,and endian swap; 0x2: The transmission length is 4 bytes,and endian swap; 0x3: The transmission length is 8 bytes,and endian swap;

bits	Name	Reset	Access	Descriptions
2:1	ch6_dev_hsize	0x0	RW	The bit width of a single data word provided by or sent to peripheral device. 0x0: one byte; 0x1: two bytes; 0x2: four bytes.
0	ch6_src_type	0x0	RW	Transfer channel data source side device type. 0x0: system memory or SRAM area with continous access address; 0x1: peripheral device FIFO mapped to a fixed register address.

CH6_LL_T_SADDR CH6_LL_T_SADDR is a channel link list address register .

Offset Address : 0xec Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
31:0	ch6_llt_saddr	0x0	RW	This parameter is used to specify the starting entry address of the linked list block allocated to the DMA channel.

CH6_CURRENT_LL_T CH6_CURRENT_LL_T is a Linked list address being processed register .

Offset Address : 0xf0 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
31:0	ch6_next_linklist	0x0	RW	The next linked list address of the current working pdma, used for debugging.

CH6_CURRENT_SIZE CH6_CURRENT_SIZE is a The transmitted length of the current channel register .

Offset Address : 0xf4 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
31:0	ch6_current_size	0x0	RW	The total length of data transported by pdma, used for debugging.

CH6_CURRENT_SADDR CH6_CURRENT_SADDR is a source address of the current transfer register .

Offset Address : 0xf8 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
31:0	ch6_current_saddr	0x0	RW	The source address of the current read request sent by the pdma, used for debugging

CH6_CURRENT_DADDR CH0_CURRENT_DADDR is a destination address of the current transfer register .

Offset Address : 0xfc Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
31:0	ch6_current_daddr	0x0	RW	The target address of the current write request sent by the pdma, which is used for debugging.

CH7_CTL CH7_CTL is a pdma control register.

Offset Address : 0x100 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions

bits	Name	Reset	Access	Descriptions
2	ch7_resume	0x0	WO	Control the pdma channel to resume transmission
1	ch7_stop	0x0	WO	Control the pdma channel to stop transmission
0	ch7_start	0x0	WO	Control the pdma channel to start transmission

CH7_STATUS CH7_STATUS is a pdma status register.

Offset Address : 0x104 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
1	ch7_pause	0x0	RO	Whether the pdma is paused
0	ch7_busy	0x0	RO	Whether the pdma is in the transmission state

CH7_CFG CH7_CFG is a pdma config register.

Offset Address : 0x108 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
27:16	ch7_dev_tout	0x0	RW	DMA channel 0 peripheral device DataReq timeout counter initial value. When reading from or writing to peripheral device, a timeout counter runs while channel waiting for device DataReq. If DataReq keep deasserting when this counter down counts to 0, current data transfer session will be stopped. 0x8: 8 cycles; 0x9: 9 cycles; 0xA: 10 cycles; 0xB: 11 cycles; 0xFFF: 4095 timer cycles; others: forbidden.
15:12	ch7_priority	0x0	RW	Transfer channel 0 priority configuration bits. Valid value can be 1 to 15. A greater value means higher priority.
11:8	ch7_dev_blen	0x0	RW	The burst length of the transaction used to read data from or write data to peripheral devices. 0x0: single burst; 0x1: burst 2; 0x2: burst 3 ; 0xF: burst 16.
5:4	ch7_dat_endian	0x0	RW	This bit field is used to specify the byte order of transmitted data. 0x0: Default configuration, no adjustment; 0x1: The transmission length is 2 bytes,and endian swap; 0x2: The transmission length is 4 bytes,and endian swap; 0x3: The transmission length is 8 bytes,and endian swap;

bits	Name	Reset	Access	Descriptions
2:1	ch7_dev_hsize	0x0	RW	The bit width of a single data word provided by or sent to peripheral device. 0x0: one byte; 0x1: two bytes; 0x2: four bytes.
0	ch7_src_type	0x0	RW	Transfer channel data source side device type. 0x0: system memory or SRAM area with continuous access address; 0x1: peripheral device FIFO mapped to a fixed register address.

CH7_LLT_SADDR CH7_LLT_SADDR is a channel link list address register .

Offset Address : 0x10c Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
31:0	ch7_llt_saddr	0x0	RW	This parameter is used to specify the starting entry address of the linked list block allocated to the DMA channel.

CH7_CURRENT_LLT CH7_CURRENT_LLT is a Linked list address being processed register .

Offset Address : 0x110 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
31:0	ch7_next_linklist	0x0	RW	The next linked list address of the current working pdma, used for debugging.

CH7_CURRENT_SIZE CH7_CURRENT_SIZE is a The transmitted length of the current channel register .

Offset Address : 0x114 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
31:0	ch7_current_size	0x0	RW	The total length of data transported by pdma, used for debugging.

CH7_CURRENT_SADDR CH7_CURRENT_SADDR is a source address of the current transfer register .

Offset Address : 0x118 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
31:0	ch7_current_saddr	0x0	RW	The source address of the current read request sent by the pdma, used for debugging

CH7_CURRENT_DADDR CH7_CURRENT_DADDR is a destination address of the current transfer register .

Offset Address : 0x11c Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
31:0	ch7_current_daddr	0x0	RW	The target address of the current write request sent by the pdma, which is used for debugging.

2.5.2 SDMA

2.5.2.1 Overview

SDMA (System Direct Memory Access) has the capability of high-speed data transmission without using CPU, it supports data transfer within DDR or SRAM, or between DDR and SRAM.

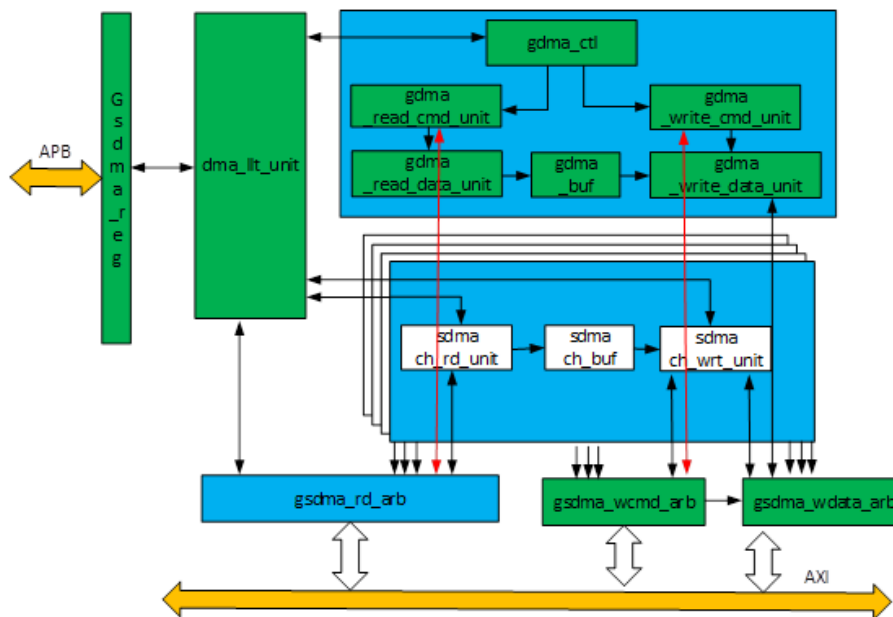
2.5.2.2 Features

The SDMA has the following features:

- Provides four DMA channels : ch0,ch1,ch2 and ch3
- The axi data width is 64-bit
- The max axi burst length is 16
- Supports configurable axi transaction outstanding numbers: the max outstanding number of ch0,ch2,ch3 is 2, the max outstanding number of ch1 is 16
- Supports configurable data endian
- Supports data transfer with the linked list
- Supports configurable weight-based channel priority arbitration
- Supports lowpower mode : each channel supports auto clock gate
- Supports specified data transfer for decompression module using ch0 and ch1

2.5.2.3 Block Diagram

Figure 2-5-2 Gsdma Block Diagram



The sdma and gdma share one

dma architecture, names as gsdma. the gsdma has 1 master data transfer interface and 1 slave config interface, the master interface is axi4 and the slave interface is apb2.0. Both the sdma and gdma access to the master interface through the arbiter and they share the base address of the register space.

2.5.2.4 Functional Description

channel priority arbitration since sdma and gdma are based on the gsdma architecture, they both follow the same arbitration mechanism. A weight-based channel arbitration mechanism is used here. The arbitration for reading/writing the master port are both implemented based on this method. The mechanism is described below. The GSDMA_WEIGHT register can be divided into eight 3-bit fields, and each field corresponds to an arbitration code. These eight arbitration codes are connected in series base on a circular manner, and a shift is performed each time an arbitration action is completed. Every 8 consecutive requests for arbitration actions constitute an arbitration cycle (there may be only one channel in a certain arbitration requesting to use the AXI master port, but this situation is still regarded as a valid arbitration action), and the current status is updated every time an arbitration action is completed. The arbitration code has 5 values: 0, 1, 2, 3, 4. When the arbitration code is '0', the channel priority is defined as: ch0 > ch1

> ch2 > ch3>gdma; When the arbitration code is '1', the channel priority is defined as: ch1 > ch2 > ch3 > gdma>ch0; When the arbitration code is '2', the channel priority is defined as: ch2 > ch3 >gdma> ch0 > ch1; When the arbitration code is '3', the channel priority is defined as: ch3 >gdma> ch0 > ch1 > ch2; When the arbitration code is '4', the channel priority is defined as: gdma > ch0 > ch1 > ch2 > ch3. Assuming that the value of the GSDMA_WEIGHTH register is configured as 0_1_2_4_4_4_3_0, these 8 values correspond to 8 arbitration actions. From this configuration value, it can be seen that channel 4 (gdma) has the largest weight, followed by channel 0, and finally channels 1, 2, and 3.

Linked List Transfer The SDMA linked list takes 24bytes as a node. The content of the linked list is placed in the memory space and The register CHx_LLT_SADDR is used to specify the entry address of the starting linked list. The linked list address needs to be 8bytes aligned. When next_ll_i_addr is 0, the linked list ends. The structure of the SDMA linked list is as follows:

Table 2-5-2-1 sdma Linked List

Bits	Description	Remarks
28	SDMA mode	0: line mode 1: 2D mode
29	pause_enable	1: dma enters the pause state after the current node is completed
30	Interrupt_enable	1: generate the node interrupt after the current node is completed
63:32	src_addr	source address
95:64	line_size	data length of a row
111:96	line_num	number of lines (required for 2D mode)
127:112	line_space	space between two lines(required for 2D mode) , Line_space=Stride-line_size
159:128	dst_addr	destination address
191:160	next_ll_i_addr	0: means the last node

1. line mode: In line mode, the total length of the current data is line_size, line_space and line_num do not need to be configured.
2. 2D mode: 2D mode can move 2D format data(such as image)to the target memory for continuous storage. The source data is stored as follows:

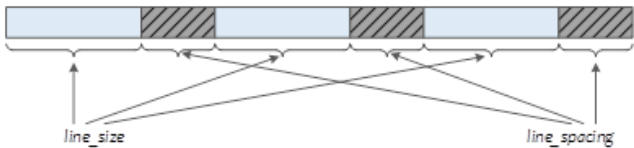


Figure 2.5.2.2 sdma line structure

The destination data is stored continuously, there is no gap between lines so line_space will not be used.

3. Fixed value filling mode: When the register chx_dat_mode is set to 1, enters the fixed value filling mode, and the value of the register chx_usr_dat is the content sent to the destination address

Pause and stop When the pause function is set in the linked list, dma enters the pause state after the current dma transfer is completed, and a pause interrupt will be issued. When the register chx_resume is enabled, obtain the next linked list and continue the dma transmission. When the stop function is enabled during dma transmission, the current transmission is canceled, but the ongoing bus transaction will be completed. After the stop function is enabled, the software needs to wait at least 1ms before starting a new transmission.

2.5.2.5 Working Mode

linked-list work mode

1. Read the register DMA_CH_EN to find and enable an available idle channel.
2. Configure the register DMA_INT_STAT to clear the interrupt status of the selected channel.
3. Configure the linked list and write it into memory. Configure the physical address of the first linked list to register CHx_LLT_SADDR;
4. Configuration register DMA_WEIGHT, configure channel arbitration weight, make sure that DMA_CKG.bit0=1 when configuring this register.
5. Configuration register CHx_CFG0, configure the transmission mode, etc.
6. Configure the register CHx_CTL to start the current DMA channel;
7. Wait for the interrupt, according to the configuration, wait for the pull-up of the linked list node interrupt or DMA completion interrupt;
8. After receiving the DMA interrupt, the DMA transfer ends.
9. Clear the interrupt status and start the next transfer.
10. After all transfers are completed, configure the register DMA_CH_EN to close the channel.

Work with gzip

When used to GZIP transmission, the ch0 is used as data prefetching channel and needs Handshaking signal dma_write_req , the ch1 is used as data write back channel and needs Handshaking signal dma_read_req The process is as follows:

1. Configure channel 0

- The first linked list node: Configure the source address as the memory address, the destination address as 0xa0000 of SRAM, and the data length as 128KB.
- The second linked list node: Configure the source address as the memory address, the destination address as 0xc0000 of SRAM, and the data length as 128KB.
- The third linked list node: Configure the source address as the memory address, the destination address as 0xa0000 of SRAM, and the data length as 128KB.
- The fourth linked list node: Configure the source address as the memory address, the destination address as 0xc0000 of SRAM, and the data length as 128KB. The entire pre-compression data is configured.

2. Configure channel 1

- The first linked list node: Configure the destination address as the memory address, the source address as 0x00000 of SRAM, and the data length as 128KB.
- The second linked list node: Configure the destination address as the memory address, the source address as 0x20000 of SRAM, and the data length as 128KB.
- The third linked list node: Configure the destination address as the memory address, the source address as 0x40000 of SRAM, and the data length as 128KB.
- The fourth linked list node: Configure the destination address as the memory address, the source address as 0x60000 of SRAM, and the data length as 128KB.
- The fifth linked list node: Configure the destination address as the memory address, the source address as 0x00000 of SRAM, and the data length as 128KB.
- The sixth linked list node: Configure the destination address as the memory address, the source address as 0x20000 of SRAM, and the data length as 128KB.

.

3. Configure the source length and destination length of the decompression module
4. Start the SDMA controller and start the decompression module.
5. SDMA detects that the decomp_ctrl_en signal is pulled high.
6. The decompression module initiates the hardware request signal dma_write_req, and SDMA channel 0 starts to transmit the first node data. When the transfer is completed, the signal dma_write_ack is pulled high. The

decompression module starts to decompress after detecting that the dma_write_ack signal is high. Decompress the first 128KB data block first, and immediately start requesting the transfer of the second 128KB data block. The decompressed data is sent to the SRAM block with address 0x0~0x7FFFF, and SDMA channel 1 is requested every time a 128KB data block is transferred, polled in this way until the decompression is completed.

2.5.2.6 Register Summary

The base address of the SDMA Register reference chapter 1.

Table 2-5-2-2 SDMA registers

Offset	Register	Description
0x0	DMA_CH_EN	dma enable register
0xc	DMA_CFG	dma configure register
0x4	DMA_INT_MASK	dma interrupt mask register
0x8	DMA_INT_STAT	dma interrupt status register
0x48	DMA_WEIGHT	dma arbitration weight register
0x50	CH0_CTL	sdma channel0 control register
0x54	CH0_STATUS	sdma channel0 status register
0x58	CH0_CFG	sdma channel0 configure register
0x5c	CH0_USR_DATA	sdma channel0 user specified data register
0x60	CH0_LLT_SADDR	sdma channel0 linked list base address register
0x64	CH0_CURRENT_LLT	sdma channel0 current linked list address register
0x80	CH1_CTL	sdma channel1 control register
0x84	CH1_STATUS	sdma channel1 status register
0x88	CH1_CFG	sdma channel1 configure register
0x8c	CH1_USR_DATA	sdma channel1 user specified data register
0x90	CH1_LLT_SADDR	sdma channel1 linked list base address register
0x94	CH1_CURRENT_LLT	sdma channel1 current linked list address register
0xb0	CH2_CTL	sdma channel2 control register
0xb4	CH2_STATUS	sdma channel2 status register
0xb8	CH2_CFG	sdma channel2 configure register
0xbc	CH2_USR_DATA	sdma channel2 user specified data register
0xc0	CH2_LLT_SADDR	sdma channel2 linked list base address register
0xc4	CH2_CURRENT_LLT	sdma channel2 current linked list address register
0xe0	CH3_CTL	sdma channel3 control register
0xe4	CH3_STATUS	sdma channel3 status register
0xe8	CH3_CFG	sdma channel3 configure register
0xec	CH3_USR_DATA	sdma channel3 user specified data register

Offset	Register	Description
0xf0	CH3_LLT_SADDR	sdma channel3 linked list base address register
0xf4	CH3_CURRENT_LLT	sdma channel3 current linked list address register

2.5.2.7 Register Description

DMA_CH_EN DMA_CH_EN is a dma enable register Offset Address:0x0 , Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
4	RW	gdma_enable	gdma channel enable 0x1: enable channel 0x0: close the channel	0x0
3:0	RW	chx_channel_enable	sdma channel enable bit0: channel 0 bit1: channel 1 bit2: channel 2 bit3: channel 3 0x1: enable channel 0x0: close the channel	0x0

DMA_CFG

DMA_CFG is a dma configure register Offset Address:0xc,Total Reset Value:0x0000_07ff

Bits	Access	Name	Description	Reset
12	RW	gdma_burst_len4_en	0b1: gdma burst length is 4 0b0: gdma burst length is 8	0x0
10:8	RW	gdma_max_outstanding	Maximum outstanding transactions 0x0: maximum is 1 0x1: maximum is 2 0x2: maximum is 3 ... 0x7: maximum is 8	0x7
7:0	RW	ckg_bypass	bit0: 1 indicates gsdma read/write arbitration clock automatic gate bypass bit1: 1 indicates gdma control module clock automatic gate bypass bit2: 1 indicates sdma channel 0 automatic gate bypass bit3: 1 indicates sdma channel 1 automatic gate bypass bit4: 1 indicates sdma channel 2 automatic gate bypass bit5: 1 indicates sdma channel 3 automatic gate bypass bit6: 1 indicates gdma read channel automatic gate bypass bit7: 1 indicates gdma write channel automatic gate bypass	0xff

DMA_INT_MASK DMA_INT_MASK is a dma interrupt mask register Offset Address:0x4.Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset

Bits	Access	Name	Description	Reset
18	RW	gdma_tr_done_mask	1: Mask the gdma transmission complete interrupt	0x0
17	RW	gdma_lli_item_int_mask	1: Mask the gdma linked list node interrupt	0x0
16	RW	gdma_lli_pause_mask	1: Mask the gdma pause interrupt	0x0
11:8	RW	chx_llt_pause_mask	Mask the sdma pause interrupt bit8: channel 0 bit9: channel 1 bit10: channel 2 bit11: channel 3	0x0
7:4	RW	chx_llt_item_int_mask	Mask the sdma linked list node interrupt bit4: channel 0 bit5: channel 1 bit6: channel 2 bit7: channel 3	0x0
3:0	RW	chx_tr_done_mask	Mask the sdma transmission complete interrupt bit0: channel 0 bit1: channel 1 bit2: channel 2 bit3: channel 3	0x0

DMA_INT_STAT DMA_INT_STAT is a dma interrupt status register Offset Address:0x8,Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
18	W1C	gdma_tr_done_int	gdma transmission completed interrupt flag, write 1 to clear	0x0
17	W1C	gdma_lli_item_int	gdma linked list node interrupt flag, write 1 to clear	0x0
16	W1C	gdma_lli_pause_int	gdma pause interrupt flag, write 1 to clear	0x0
11:8	W1C	chx_llt_pause_int	sdma pause interrupt flag, write 1 to clear bit8: channel 0 bit9: channel 1 bit10: channel 2 bit11: channel 3	0x0
7:4	W1C	chx_llt_item_int	sdma linked list node interrupt flag, write 1 to clear bit4: channel 0 bit5: channel 1 bit6: channel 2 bit7: channel 3	0x0
3:0	W1C	chx_tr_done_int	sdma transmission completed interrupt flag, write 1 to clear bit0: channel 0 bit1: channel 1 bit2: channel 2 bit3: channel 3	0x0

DMA_WEIGHT DMA_WEIGHT is a dma arbitration weight register Offset Address:0x48,Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
23:0	RW	gsdma_weight	Used to control the access priority weight of the DMA channels. This register is divided into 8 fields, each field is 3-bit, and each 3-bit field can be used to set an arbitration code. This arbitration code is used to determine the priority of the DMA channel when requesting to use the master AXI port. The arbitration code has 5 valid values: 0, 1, 2, 3, 4 0: ch0 > ch1 > ch2 > ch3 > gdma ; 1: ch1 > ch2 > ch3 > gdma > ch0; 2: ch2 > ch3 > gdma > ch0 > ch1; 3: ch3 > gdma > ch0 > ch1 > ch2; 4: gdma > ch0 > ch1 > ch2 > ch3 ;	0x0

CH0_CTL CH0_CTL is a sdma channel0 control register Offset Address:0x50,Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
2	WO1	ch0_resume	sdma channel0 resume after pause	0x0
1	WO1	ch0_stop	sdma channel0 stop working	0x0
0	WO1	ch0_start	sdma channel0 start	0x0

CH0_STATUS CH0_STATUS is a sdma channel0 status register Offset Address:0x54 , Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
1	RO	ch0_pause	sdma channel0 pause flag	0x0
0	RO	ch0_busy	sdma channel0 working flag	0x0

CH0_CFG CH0_CFG is a sdma channel0 configure register Offset Address:0x58 , Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
23:20	RW	ch0_max_rd_outstanding	the max axi read transaction outstanding numbers 0: max outstanding 1 1: max outstanding 2	0x0
19:16	RW	ch0_max_wr_outstanding	the max axi write transaction outstanding numbers 0: max outstanding 1 1: max outstanding 2	0x0
10	RW	decomp_ctrl_en	decompression enable flag	0x0
9	RW	ch0_dst_fixed	Destination address fixed flag 0: Destination address increments continuously 1: The destination address is fixed	0x0
8	RW	ch0_src_fixed	Source address fixed flag 0: Source address increments continuously 1: The source address is fixed	0x0

Bits	Access	Name	Description	Reset
5:4	RW	ch0_dat_endian	This field is used to specify the byte order of the transmitted data 0: default configuration, do nothing 1: 2-byte, the transmission length is 2byte, and the endian conversion is performed 2: 4-byte, the transmission length is 4byte, and the endian conversion is performed 3: 8-byte, the transmission length is 8byte, and the endian conversion is performed	0x0
2:1	RW	ch0_usr_dat_siz	This field is used to specify the effective byte quantity of the data in the CHx_USR_DATA register. 0: 1-byte, the lowest 8-bit is valid; 1: 2-byte, the lowest 16-bit is valid; 2: 4-byte, the entire CHx_USR_DATA is valid	0x0
0	RW	ch0_dat_mode	This field is used to specify the data mode during DMA transfer. 0: normal mode, read data from the source and transfer to the destination; 1: Fixed value mode, transfer the data in the register CHx_USR_DATA to the destination.	0x0

CH0_USR_DATA CH0_USR_DATA is a sdma channel0 user specified data register Offset Address:0x5c,Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:0	RW	ch0_usr_data	When the chx_dat_mode field in the CHx_CFG register is set to '1', The data in this register will be repeatedly transferred to the destination	0x0

CH0_LLT_SADDR

CH0_LLT_SADDR is a sdma channel0 linked list base address register Offset Address:0x60,Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:0	RW	ch0_llt_saddr	This register is used to specify the entry address of the DMA linked list block	0x0

CH0_CURRENT_LLT CH0_CURRENT_LLT is a sdma channel0 current linked list address register Offset Address:0x64,Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:0	RO	ch0_current_linklist	Linked list address of current transaction	0x0

CH1_CTL CH1_CTL is a sdma channel1 control register Offset Address:0x80,Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
2	WO1	ch1_resume	sdma channel1 resume after pause	0x0

Bits	Access	Name	Description	Reset
1	WO1	ch1_stop	sdma channel1 stop working	0x0
0	WO1	ch1_start	sdma channel1 start	0x0

CH1_STATUS CH1_STATUS is a sdma channel1 status register Offset Address:0x84,Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
1	RO	ch1_pause	sdma channel1 pause flag	0x0
0	RO	ch1_busy	sdma channel1 working flag	0x0

CH1_CFG CH1_CFG is a sdma channel1 configure register Offset Address:0x88,Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
23:20	RW	ch1_max_rd_outstanding	the max axi read transaction outstanding numbers 0: max outstanding 1 1: max outstanding 2 2: max outstanding 3 ... 15: max outstanding 16	0x0
19:16	RW	ch1_max_wr_outstanding	the max axi write transaction outstanding numbers 0: max outstanding 1 1: max outstanding 2 2: max outstanding 3 ... 15: max outstanding 16	0x0
9	RW	ch1_dst_fixed	Destination address fixed flag 0: Destination address increments continuously 1: The destination address is fixed	0x0
8	RW	ch1_src_fixed	Source address fixed flag 0: Source address increments continuously 1: The source address is fixed	0x0
5:4	RW	ch1_dat_endian	This field is used to specify the byte order of the transmitted data 0: default configuration, do nothing 1: 2-byte, the transmission length is 2byte, and the endian conversion is performed 2: 4-byte, the transmission length is 4byte, and the endian conversion is performed 3: 8-byte, the transmission length is 8byte, and the endian conversion is performed	0x0
2:1	RW	ch1_usr_dat_siz	This field is used to specify the effective byte quantity of the data in the CHx_USR_DATA register. 0: 1-byte, the lowest 8-bit is valid; 1: 2-byte, the lowest 16-bit is valid; 2: 4-byte, the entire CHx_USR_DATA is valid	0x0

Bits	Access	Name	Description	Reset
0	RW	ch1_dat_mode	This field is used to specify the data mode during DMA transfer. 0: normal mode, read data from the source and transfer to the destination; 1: Fixed value mode, transfer the data in the register CHx_USR_DATA to the destination.	0x0

CH1_USR_DATA CH1_USR_DATA is a sdma channel1 user specified data register Offset Address:0x8c,Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:0	RW	ch1_usr_data	When the chx_dat_mode field in the CHx_CFG register is set to '1', The data in this register will be repeatedly transferred to the destination	0x0

CH1_LLT_SADDR

CH1_LLT_SADDR is a sdma channel1 linked list base address register Offset Address:0x90,Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:0	RW	ch1_llt_saddr	This register is used to specify the entry address of the DMA linked list block	0x0

CH1_CURRENT_LLT CH1_CURRENT_LLT is a sdma channel1 current linked list address register Offset Address:0x94,Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:0	RO	ch1_current_linklist	Linked list address of current transaction	0x0

CH2_CTL CH2_CTL is a sdma channel2 control register Offset Address:0xb0,Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
2	WO1	ch2_resume	sdma channel2 resume after pause	0x0
1	WO1	ch2_stop	sdma channel2 stop working	0x0
0	WO1	ch2_start	sdma channel2 start	0x0

CH2_STATUS CH2_STATUS is a sdma channel2 status register Offset Address:0xb4,Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
1	RO	ch2_pause	sdma channel2 pause flag	0x0
0	RO	ch2_busy	sdma channel2 working flag	0x0

CH2_CFG CH2_CFG is a sdma channel2 configure register Offset Address:0xb8,Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset

Bits	Access	Name	Description	Reset
23:20	RW	ch2_max_rd_outstanding	the max axi read transaction outstanding numbers 0: max outstanding 1 1: max outstanding 2	0x0
19:16	RW	ch2_max_wr_outstanding	the max axi write transaction outstanding numbers 0: max outstanding 1 1: max outstanding 2	0x0
9	RW	ch2_dst_fixed	Destination address fixed flag 0: Destination address increments continuously 1: The destination address is fixed	0x0
8	RW	ch2_src_fixed	Source address fixed flag 0: Source address increments continuously 1: The source address is fixed	0x0
5:4	RW	ch2_dat_endian	This field is used to specify the byte order of the transmitted data 0: default configuration, do nothing 1: 2-byte, the transmission length is 2byte, and the endian conversion is performed 2: 4-byte, the transmission length is 4byte, and the endian conversion is performed 3: 8-byte, the transmission length is 8byte, and the endian conversion is performed	0x0
2:1	RW	ch2_usr_dat_siz	This field is used to specify the effective byte quantity of the data in the CHx_USR_DATA register. 0: 1-byte, the lowest 8-bit is valid; 1: 2-byte, the lowest 16-bit is valid; 2: 4-byte, the entire CHx_USR_DATA is valid	0x0
0	RW	ch2_dat_mode	This field is used to specify the data mode during DMA transfer. 0: normal mode, read data from the source and transfer to the destination; 1: Fixed value mode, transfer the data in the register CHx_USR_DATA to the destination.	0x0

CH2_USR_DATA CH2_USR_DATA is a sdma channel2 user specified data register Offset Address:0xbc,Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:0	RW	ch2_usr_data	When the chx_dat_mode field in the CHx_CFG register is set to '1', The data in this register will be repeatedly transferred to the destination	0x0

CH2_LLT_SADDR

CH2_LLT_SADDR is a sdma channel2 linked list base address register Offset Address:0xc0,Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset

Bits	Access	Name	Description	Reset
31:0	RW	ch2_llt_saddr	This register is used to specify the entry address of the DMA linked list block	0x0

CH2_CURRENT_LLT CH2_CURRENT_LLT is a sdma channel2 current linked list address register Offset Address:0xc4,Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:0	RO	ch2_current_linklist	Linked list address of current transaction	0x0

CH3_CTL CH3_CTL is a sdma channel3 control register Offset Address:0xe0,Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
2	WO1	ch3_resume	sdma channel3 resume after pause	0x0
1	WO1	ch3_stop	sdma channel3 stop working	0x0
0	WO1	ch3_start	sdma channel3 start	0x0

CH3_STATUS CH3_STATUS is a sdma channel3 status register Offset Address:0xe4,Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
1	RO	ch3_pause	sdma channel3 pause flag	0x0
0	RO	ch3_busy	sdma channel3 working flag	0x0

CH3_CFG CH3_CFG is a sdma channel3 configure register Offset Address:0xe8,Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
23:20	RW	ch3_max_rd_outstanding	the max axi read transaction outstanding numbers 0: max outstanding 1 1: max outstanding 2	0x0
19:16	RW	ch3_max_wr_outstanding	the max axi write transaction outstanding numbers 0: max outstanding 1 1: max outstanding 2	0x0
9	RW	ch3_dst_fixed	Destination address fixed flag 0: Destination address increments continuously 1: The destination address is fixed	0x0
8	RW	ch3_src_fixed	Source address fixed flag 0: Source address increments continuously 1: The source address is fixed	0x0

Bits	Access	Name	Description	Reset
5:4	RW	ch3_dat_endian	This field is used to specify the byte order of the transmitted data 0: default configuration, do nothing 1: 2-byte, the transmission length is 2byte, and the endian conversion is performed 2: 4-byte, the transmission length is 4byte, and the endian conversion is performed 3: 8-byte, the transmission length is 8byte, and the endian conversion is performed	0x0
2:1	RW	ch3_usr_dat_siz	This field is used to specify the effective byte quantity of the data in the CHx_USR_DATA register. 0: 1-byte, the lowest 8-bit is valid; 1: 2-byte, the lowest 16-bit is valid; 2: 4-byte, the entire CHx_USR_DATA is valid	0x0
0	RW	ch3_dat_mode	This field is used to specify the data mode during DMA transfer. 0: normal mode, read data from the source and transfer to the destination; 1: Fixed value mode, transfer the data in the register CHx_USR_DATA to the destination.	0x0

CH3_USR_DATA CH3_USR_DATA is a sdma channel3 user specified data register Offset Address:0xec,Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:0	RW	ch3_usr_data	When the chx_dat_mode field in the CHx_CFG register is set to '1', The data in this register will be repeatedly transferred to the destination	0x0

CH3_LLT_SADDR

CH3_LLT_SADDR is a sdma channel3 linked list base address register Offset Address:0xf0,Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:0	RW	ch3_llt_saddr	This register is used to specify the entry address of the DMA linked list block	0x0

CH3_CURRENT_LLT CH3_CURRENT_LLT is a sdma channel3 current linked list address register Offset Address:0xf4,Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:0	RO	ch3_current_linklist	Linked list address of current transaction	0x0

2.6 Timer

2.6.1 overview

K230 provides 9 timers (six general timer and three stc timer). Six general-purpose timers can be used as system clock of operating system and used to count external signal input. Stc0 timer can be used for video-input/video-output and audio synchronization. STC2 timer is used for cpu0 and STC3 timer is used for cpu1.

2.6.2 features

general-purpose timers feature list:

- APB interface supports APB3
- Up to six programmable timers
- Configurable timer width: 8 to 32 bits
- Support for two operation modes: free-running and user-defined count
- Support for independent clocking of timers
- Configurable polarity for each individual interrupt
- Configurable option for a single or combined interrupt output flag
- Configurable option to have read/write coherency registers for each timer
- configurable option to include timer toggle output, which toggles whenever timer counter reloads
- Configurable option to enable programmable pulse-width modulation of timer toggle outputs
- Configurable option to include pulse width modulation of timer toggle output with 0% and 100% duty cycle.
- can be used to count external input signal with frequency less than 1MHz and duty cycle from 0~100%

stc timer feature list:

- 64 bit timer width
- 3 configurable timescale : 27MHz, 1MHz, 90KHz
- configurable enable, load, clear control register
- apb 3.0 interface
- One timer can be used for video-input/video-output and audio synchronization.
- One timer is used for cpu0 and one timer is used for cpu1.
- timer used for video/audio synchronization provides two sets of timer value, one is old value and the other is latest value.

2.6.3 function description

General Timers count down from a programmed value and generate an interrupt when the count reaches zero.

Each general timer has an independent clock input, timer_N_clk (where N is in the range 1 to 6). You can configure the width of a timer from 8 to 32 bits using the TIMER_WIDTH_N parameter (Width of Timer N), where N is in the range 1 to 6, the number of instantiated timers.

The initial value for each timer – that is, the value from which it counts down is loaded into the timer using the appropriate load count register (TimerNLoadCount). Two events can cause a timer to load the initial count from its TimerNLoadCount register :

- Timer is enabled after being reset or disabled
- Timer counts down to 0 All interrupt status registers and end-of-interrupt registers can be accessed at any time.

2.6.3.1 general timer usage flow

1. Initialize the timer through the TimerNControlReg register (where N is in the range 1 to 8):

- a. Disable the timer by writing a “0” to the timer enable bit (bit 0); accordingly, the timer_en output signal is de-asserted.

- b. Program the timer mode—user-defined or free-running—by writing a “0” or “1,” respectively, to the timer mode bit (bit 1).
 - c. Set the interrupt mask as either masked or not masked by writing a “1” or “0,” respectively, to the timer interrupt mask bit (bit 2).
2. Load the timer counter value into the TimerNLoadCount register (where N is in the range 1 to 8).
 3. Enable the timer by writing a “1” to bit 0 of TimerNControlReg.

2.6.3.2 general timers configuration

choosing the number of timers

Suppose N represent the Number of Timers (where N is from 1 to 6):

- TimerNLoadCount – TimerN load count register
- TimerNLoadCount2 (optional) – TimerN load count register for programming width of HIGH period of timer_N_toggle output
- TimerNCurrentValue – TimerN current value register
- TimerNControlReg – TimerN control register
- TimerNEOI – TimerN end-of-interrupt register
- TimerNIntStatus – TimerN interrupt status register

Thus there are five individual registers for each of the timers in soc design. All other registers control their respective functions for all active timers, rather than for individual timers.

Enabling and Disabling a Timer

You use bit 0 of the TimerNControlReg, where N is in the range 1 to 8, to either enable or disable a timer.

Enabling a Timer

If you want to enable a timer, you write a “1” to bit 0 of its TimerNControlReg register.

Disabling a Timer

To disable a timer, write a “0” to bit 0 of its TimerNControlReg register.

When a timer is enabled and running, its counter decrements on each rising edge of its clock signal, timer_N_clk. When a timer transitions from disabled to enabled, the current value of its TimerNLoadCount register is loaded into the timer counter on the next rising edge of timer_N_clk.

When the timer enable bit is de-asserted and the timer stops running, the timer counter and any associated registers in the timer clock domain, such as the toggle register, are asynchronously reset.

When the timer enable bit is asserted, then a rising edge on the timer_en signal is used to load the initial value into the timer counter. A “0” is always read back when the timer is not enabled; otherwise, the current value of the timer (TimerNCurrentValue register) is read back.

Configuring the Width of a Timer

You configure the width of a timer through the TIMER_WIDTH_N parameter; each timer can be from 8 bits to 32 bits.

Loading a Timer Countdown Value

When a timer counter is enabled after being reset or disabled, the count value is loaded from the TimerNLoadCount register; this occurs in both free-running and user-defined count modes.

When a timer counts down to 0, it loads one of two values, depending on the timer operating mode:

- User-defined count mode – Timer loads the current value of the TimerNLoadCount register. Use this mode if you want a fixed, timed interrupt. Designate this mode by writing a “1” to bit 1 of TimerNControlReg.
- Free-running mode – Timer loads the maximum value, which is dependent on the timer width; that is, the TimerNLoadCount register is comprised of $2^{\text{TIMER\_WIDTH\_N}} - 1$ bits, all of which are loaded with 1s. The timer counter wrapping to its maximum value allows time to reprogram or disable the timer before another interrupt occurs. Use this mode if you want a single timed interrupt. Designate this mode by writing a “0” to bit 1 of TimerNControlReg.

Working with Interrupts

The TimerNIntStatus and TimerNEOI registers handle interrupts in order to ensure safe operation of the interrupt clearing. **Clearing Interrupts**

Provided the timer is enabled, its interrupt remains asserted until it is cleared by reading one of two end-of-interrupt registers (TimerNEOI or TimersEOI, the individual and global end-of-interrupt registers, respectively). When the timer is disabled, the timer interrupt is cleared. You can clear an individual timer interrupt by reading its TimerNEOI register. You can clear all active timer interrupts at once by reading the global TimersEOI register or by disabling the timer.

If an end-of-interrupt register is read during the time when the internal interrupt signal is high, the timer interrupt is set. This occurs because setting timer interrupts takes precedence over clearing them.

Checking Interrupt Status

You can query the interrupt status of an individual timer without clearing its interrupt by reading the TimerNIntStatus register. You can query the interrupt status of all timers without clearing the interrupts by reading the global TimersIntStatus register.

Masking Interrupts

Each individual timer interrupt can be masked using its TimerNControlReg register. To mask an interrupt, you write a “1” to bit 2 of TimerNControlReg. If all individual timer interrupts are masked, then the combined interrupt is also masked.

Setting Interrupt Polarity

The polarity of the generated timer interrupts can be configured to be either active-high or active-low using the TIM_INTRPT_PLRITY parameter (Interrupt Polarity). In addition to an interrupt output signal for each timer, there is also a single, global interrupt flag, timer_intr_flag, that is asserted if any timer asserts its interrupt. This global interrupt flag shares the same polarity characteristic with the other generated interrupts; thus, multiple interrupt service schemes can be supported.

Timer Pause Mode

The operation of a timer can be paused by asserting the respective timer_N_pause input signal, which is synchronized to the timer_N_clk domain.

2.6.3.3 stc timer usage flow

1. configure STCn_INIT_VALUE, this is the start point of stc timer. If this register is not configured, default value 0 will be used.
2. Configure stc timer timescale. STCn_CFG.stcn_mode is used to configure timescale. 0: 27MHz, 1: 1MHz, 2: 90KHz.
3. Load initial value. Configure STCn_CFG.stcn_load to 1 to load initial value. Otherwise default value 0 will be loaded.
4. For audio module, configure STCn_CFG.aud0_cap_en or aud1_cap_en to 1 to enable audio capture signal.
5. Enable stc timer. Configure STCn_CFG.stcn_en to 1. Stc timer will run from start point.
6. For video/audio sync timer, wait for hardware capture signal. Stc timer will be sent to cpu0 and cpu1 automatically, dont need hardware capture signal.

2.6.3.4 stc timer confuiguration

initial value

STCn_INIT_VALUE_HIGH is the high 32bit of initial value. STCn_INIT_VALUE_LOW is the low 32bit of initial value.

mode

Configure STCn_CFG.stcn_mode to correct value according timescale needed.

load value

Configure STCn_CFG.stcn_load to 1.

clear timer

Configure STCn_CFG.clear to 1 to clear timer to 0.

enable timer

Configure STCn_CFG.stcn_en to 1 to start timer working.

audio capture enable

configure STCn_CFG.aud0_cap_en or aud1_cap_en to 1 to enable audio capture signal.

get timer result

Timer for video input

There are 3 group timer for video input : STC_VI_CAP_x_y_Value_z .

x means goup number with range from 0 to 2.

y means latest or old value. 0 represent the latest value. 1 represent the old value.

z means high or low 32 bit. HIGH is the high 32 bit value. LOW is the low 32 bit value.

timer for video output

video output timer : STC_VO_CAP_x_Value_z

x means latest or old value. 0 represent the latest value. 1 represent the old value.

z means high or low 32 bit. HIGH is the high 32 bit value. LOW is the low 32 bit value.

timer for audio

STC_AUDx_CAP_y_Value_z: stc timer for audio

x means audio channel which can be used for audio in or audio out. 0: audio channel 0, 1: audio channel 1

y means latest or old value. 0 represent the latest value. 1 represent the old value.

z means high or low 32 bit. HIGH is the high 32 bit value. LOW is the low 32 bit value.

2.6.4 register summary

Some registers are common between different general timers and some are different.

General timer register base address : refer to [Timer base address](#)

Table2-6-1 general timer register summary

Offset	Register	Description
0x00 +(N-1)*0x14	TimerNLoadCount (for N = 1; N <= NUM_TIMERS)	Value to be loaded into Timer N Reset value: 0x0
0x04 +(N-1)*0x14	TimerNCurrentValue (for N = 1; N <= NUM_TIMERS)	Current value of Timer N Reset value: (TIM_NEWMODE ==1) ? '2 power ((TIMER_WIDTH_1)-1)' : 0
0x08 +(N-1)*0x14	TimerNControlReg (for N = 1; N <= NUM_TIMERS)	Control Register for Timer N. This register controls enabling, operating mode (free-running or defined-count),...
0x0C + (N-1)*0x14	TimerNEOI (for N = 1; N <= NUM_TIMERS)	Clears the interrupt from Timer N Reset value: 0x0
0x10 +(N-1)*0x14	TimerNIntStatus (for N = 1; N <= NUM_TIMERS)	Contains the interrupt status for Timer N Reset value: 0x0
0xa0	TimersIntStatus	Contains the interrupt status of all timers in the component.Reset value: 0x0
0xa4	TimersEOI	Returns all zeroes (0) and clears all active interrupts. Reset value: 0x0
0xa8	TimersRawIntStatus	Contains the unmasked interrupt status of all timers in the component. Reset value: 0x0
0xac	TIMERS_COMP_VERSION	Current revision number of the DW_apb_timers component.Reset value: TIM_VERSION_ID
0xb0 +(N-1)*0x04	TimerNLoadCount2 (for N = 1; N <= NUM_TIMERS)	Value to be loaded into Timer N when toggle output changes from 0 to 1 Reset value: 0x0

STC timer register base address : refer to [STC base address](#)

Table 2-6-2 stc timer register summary

Offset	Register	Description
0X0	STC0_CFG	STC0 configuration
0X4	STC0_INIT_VALUE_HIGH	STC0 initial value high 32bit
0X8	STC0_INIT_VALUE_LOW	STC0 initial value low 32bit

Offset	Register	Description
0X20	STC2_CFG	STC2 configuration
0X24	STC2_INIT_VALUE_HIGH	STC2 initial value high 32bit
0X28	STC2_INIT_VALUE_LOW	STC2 initial value low 32bit
0X30	STC3_CFG	STC3 configuration
0X34	STC3_INIT_VALUE_HIGH	STC3 initial value high 32bit
0X38	STC3_INIT_VALUE_LOW	STC3 initial value low 32bit
0X40	STC_VI_CAP_0_0_Value_High	video in0 timer latest value high 32 bit
0X44	STC_VI_CAP_0_0_Value_Low	video in0 timer latest value low 32 bit
0X48	STC_VI_CAP_0_1_Value_High	video in0 timer old value high 32 bit
0X4c	STC_VI_CAP_0_1_Value_Low	video in0 timer old value low 32 bit
0X50	STC_VI_CAP_1_0_Value_High	video in1 timer latest value high 32 bit
0X54	STC_VI_CAP_1_0_Value_Low	video in1 timer latest value low 32 bit
0X58	STC_VI_CAP_1_1_Value_High	video in1 timer old value high 32 bit
0X5c	STC_VI_CAP_1_1_Value_Low	video in1 timer old value low 32 bit
0X60	STC_VI_CAP_2_0_Value_High	video in2 timer latest value high 32 bit
0X64	STC_VI_CAP_2_0_Value_Low	video in2 timer latest value low 32 bit
0X68	STC_VI_CAP_2_1_Value_High	video in2 timer old value high 32 bit
0X6c	STC_VI_CAP_2_1_Value_Low	video in2 timer old value low 32 bit
0X70	STC_VO_CAP_0_Value_High	video out timer latest value high 32 bit
0X74	STC_VO_CAP_0_Value_Low	video out timer latest value low 32 bit
0X78	STC_VO_CAP_1_Value_High	video out timer old value high 32 bit
0X7c	STC_VO_CAP_1_Value_Low	video out timer old value low 32 bit
0X80	STC_AUD0_CAP_0_Value_High	audio channel 0 timer latest value high 32 bit
0X84	STC_AUD0_CAP_0_Value_Low	audio channel 0 timer latest value low 32 bit
0X88	STC_AUD0_CAP_1_Value_High	audio channel 0 timer old value high 32 bit
0X8c	STC_AUD0_CAP_1_Value_Low	audio channel 0 timer old value low 32 bit
0X90	STC_AUD1_CAP_0_Value_High	audio channel 1 timer latest value high 32 bit
0X94	STC_AUD1_CAP_0_Value_Low	audio channel 1 timer latest value low 32 bit
0X98	STC_AUD1_CAP_1_Value_High	audio channel 1 timer old value high 32 bit
0X9c	STC_AUD1_CAP_1_Value_Low	audio channel 1 timer old value low 32 bit

2.6.5 register description

TimerNLoadCount

TimerNLoadCount is used to set value to be loaded into Timer N.

Offset Address: 0x00 + (N-1)*0x14 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
31:y	R	RSVD_TimerNLoadCount	TimerNLoadCount 31toTIMER_WIDTH_N Reserved field Exists: Always Range Variable[y]: TIMER_WIDTH_N	0
x:0	R/W	TimerNLoadCount	Value to be loaded into Timer N. This is the value from which counting commences. Any value written to this register is loaded into the associated timer. Exists: Always Range Variable[x]: TIMER_WIDTH_N - 1	0

TimerNCurrentValue

TimerNCurrentValue is the current value of Timer N.

Offset Address: 0x04 + (N-1)*0x14 Total Reset Value: 0xFFFF_FFFF

Bits	Access	Name	Description	Reset
31:y	R	RSVD_TimerNCurrentValue	TimerNCurrentValue 31toTIMER_WIDTH_N Reserved field Exists: Always Volatile: true Range Variable[y]: TIMER_WIDTH_N	{31-y+1{1'b1}}
x:0	R	TimerNCurrentValue	Current Value of Timer N. When TIM_NEWMODE=0, This register is supported only when timer_N_clk is synchronous to pclk. Reading this register when using independent clocks results in an undefined value. When TIM_NEWMODE=1, no restrictions apply. Exists: Always Volatile: true Range Variable[x]: TIMER_WIDTH_N - 1	{x+1{1'b1}}

TimerNControlReg

Control Register for Timer N. This register controls enabling, operating mode (freerunning or defined-count), and interrupt mask of Timer N. You can program each Timer1ControlReg to enable or disable a specific timer and to control its mode of operation.

Offset Address: 0x08 + (N-1)*0x14 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
31:4	R	RSVD_TimerNControlReg	TimerNControlReg 31to5 Reserved field Exists: Always	0

Bits	Access	Name	Description	Reset
3	RW	TIMER_PWM	Pulse Width Modulation of timer_N_toggle output. This field is only present when TIM_NEWMODE is enabled Values: - 0x1 (ENABLED): PWM for timer_N_toggle o/p is enabled - 0x0 (DISABLE): PWM for timer_N_toggle o/p is disabled Exists: TIM_NEWMODE==1	0
2	RW	TIMER_INTERRUPT_MASK	Timer interrupt mask for Timer N. Values: - 0x1 (MASKED): Timer N interrupt is masked -- 0x0 (UNMASKED): Timer N interrupt is unmasked Exists: Always	0
1	RW	TIMER_MODE	Timer mode for Timer N. ## Note: You must set the Timer1LoadCount register to all 1s before enabling the timer in free-running mode. Values: - 0x1 (USER_DEFINED): User-Defined mode of operation - 0x0 (FREE_RUNNING): Free Running mode of operation Exists: Always	0
0	RW	TIMER_ENABLE	Timer enable bit for Timer N. Values: - 0x1 (ENABLED): Timer N is enabled - 0x0 (DISABLE): Timer N is disabled Exists: Always	0

TimerNEOI

Timer N End-of-Interrupt Register.

Offset Address: 0x0C + (N-1)*0x14 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
31:1	R	RSVD_TimerNEOI	TimerNEOI 31to1 Reserved field Exists: Always	0
0	R	TimerNEOI	Reading from this register returns all zeroes (0) and clears the interrupt from Timer N. Exists: Always	0

TimerNIntStatus

Timer N Interrupt Status Register which contains the interrupt status for Timer N.

Offset Address: 0x10 + (N-1)*0x14 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
31:1	R	RSVD_TimerNIntStatus	TimerNIntStatus 31to1 Reserved field Exists: Always Volatile: true	0
0	R	TimerNIntStatus	Contains the interrupt status for Timer N. Values: - 0x1 (ACTIVE): Timer N Interrupt is active - 0x0 (INACTIVE): Timer N Interrupt is inactive Exists: Always Volatile: true	0

TimersIntStatus

Timers Interrupt Status Register which contains the interrupt status of all timers in the component.

Offset Address: 0xA0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
31:1	R	RSVD_TimerNIntStatus	TimerNIntStatus 31to1 Reserved field Exists: Always Volatile: true	0
0	R	TimerNIntStatus	Contains the interrupt status for Timer N. Values: - 0x1 (ACTIVE): Timer N Interrupt is active - 0x0 (INACTIVE): Timer N Interrupt is inactive Exists: Always Volatile: true	0

TimersIntStatus

Timers Interrupt Status Register which contains the interrupt status of all timers in the component.

Offset Address: 0xA0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
31:y	R	RSVD_TimersIntStatus	TimersIntStatus 31toNUM_TIMERS Reserved field Exists: Always Volatile: true Range Variable[y]: NUM_TIMERS	0

Bits	Access	Name	Description	Reset
x:0	R	TimersIntStatus	Contains the interrupt status of all timers in the component. If a bit of this register is 0, then the corresponding timerinterrupt is not active and the corresponding interrupt could be on either the timer_intr bus or the timer_intr_n bus, depending on the interrupt polarity you have chosen. Similarly, if a bit of this register is 1, then the corresponding interrupt bit has been set in the relevant interrupt bus. In both cases, the status reported is the status after the interrupt mask has been applied. Reading from this register does not clear any active interrupts. Values: - 0x1 (ACTIVE): Timer_intr(_n) is active - 0x0 (INACTIVE): Timer_intr(_n) is inactive Exists: Always Volatile: true Range Variable[x]: NUM_TIMERS - 1	0

TimerEOI

Timers End-of-Interrupt Register. Returns all zeroes (0) and clears all active interrupts.

Offset Address: 0xA4 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
31:y	R	RSVD_TIMERSEOI	TimersEOI 31toNUM_TIMERS Reserved field Exists: Always Range Variable[y]: NUM_TIMERS	0
x:0	R	TIMERSEOI	Reading this register returns all zeroes (0) and clears all active interrupts. Exists: Always Range Variable[x]: NUM_TIMERS - 1	0

TimersRawIntStatus

Timers Raw Interrupt Status Register. Contains the unmasked interrupt status of all timers in the component.

Offset Address: 0xA8 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
31:y	R	RSVD_TIMERSRAWINTSTAT	TimersRawIntStatus 31toNUM_TIMERS Reserved field Exists: Always Volatile: true Range Variable[y]: NUM_TIMERS	0

Bits	Access	Name	Description	Reset
x:0	R	TIMERSRAWINTSTAT	The register contains the unmasked interrupt status of all timers in the component. Values: - 0x1 (ACTIVE): Raw Timer_intr(_n) is active - 0x0 (INACTIVE): Raw Timer_intr(_n) is inactive Exists: Always Volatile: true Range Variable[x]: NUM_TIMERS - 1	0

TIMERS_COMP_VERSION

Timers Component Version. Current revision number of the DW_apb_timers component.

Offset Address: 0xAC Total Reset Value: 0x3231_312A

Bits	Access	Name	Description	Reset
31:0	R	TIMERSCOMPVERSION	Current revision number of the DW_apb_timers component Exists: Always	0

TimerNLoadCount2

Timer N Load Count2 Register. Value to be loaded into Timer N when toggle output changes from 0 to 1

Offset Address: 0xB0 + (N-1)*0x04 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
31:y	R	RSVD_TIMERNLOADCOUNT2	TimerNLoadCount2 31toTIMER_WIDTH_N Reserved field Exists: Always Range Variable[y]: TIMER_WIDTH_1	0
x:0	RW	TIMERNLOADCOUNT2	Value to be loaded into Timer N when timer_N_toggle output changes from 0 to 1. This value determines the width of the HIGH period of the timer_N_toggle output. Exists: Always Range Variable[x]: TIMER_WIDTH_1 - 1	0

STC0_CFG

Offset=0x0, Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
6	RW	aud1_cap_en	1: enable pdma int for aud1 cap	0x0
5	RW	aud0_cap_en	1: enable pdma int for aud0 cap	0x0
4:3	RW	stc0_mode	0: 27M, 1: 1M, 2: 90K	0x0
2	W1T	stc0_clear	1: clear counter	0x0
1	W1T	stc0_load	1: load new value	0x0

Bits	Access	Name	Descriptions	Reset
0	RW	stc0_en	1: stc enable	0x0

STC0_INIT_VALUE_HIGH

Offset=0x4, Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31:0	RW	stc0_init_value_high	time stame initial value, when load =1, this value is loaded to time stamp high 32bit	0x0

STC0_INIT_VALUE_LOW

Offset=0x8, Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31:0	RW	stc0_init_value_low	time stame initial value, when load =1, this value is loaded to time stamp low 32bit	0x0

STC1_CFG

Offset=0x10, Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
4:3	RW	stc1_mode	0: 27M, 1: 1M, 2: 90K	0x0
2	W1T	stc1_clear	1: clear counter	0x0
1	W1T	stc1_load	1: load new value	0x0
0	RW	stc1_en	1: stc enable	0x0

STC1_INIT_VALUE_HIGH

Offset=0x14, Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31:0	RW	stc1_init_value_high	time stame initial value, when load =1, this value is loaded to time stamp high 32bit	0x0

STC1_INIT_VALUE_LOW

Offset=0x18, Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31:0	RW	stc1_init_value_low	time stame initial value, when load =1, this value is loaded to time stamp low 32bit	0x0

STC2_CFG

Offset=0x20, Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
4:3	RW	stc2_mode	0: 27M, 1: 1M, 2: 90K	0x0
2	W1T	stc2_clear	1: clear counter	0x0
1	W1T	stc2_load	1: load new value	0x0
0	RW	stc2_en	1: stc enable	0x0

STC2_INIT_VALUE_HIGH

Offset=0x24, Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31:0	RW	stc2_init_value_high	time stame initial value, when load =1, this value is loaded to time stamp high 32bit	0x0

STC2_INIT_VALUE_LOW

Offset=0x28, Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31:0	RW	stc2_init_value_low	time stame initial value, when load =1, this value is loaded to time stamp low 32bit	0x0

STC3_CFG

Offset=0x30, Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
4:3	RW	stc3_mode	0: 27M, 1: 1M, 2: 90K	0x0
2	W1T	stc3_clear	1: clear counter	0x0
1	W1T	stc3_load	1: load new value	0x0
0	RW	stc3_en	1: stc enable	0x0

STC3_INIT_VALUE_HIGH

Offset=0x34, Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31:0	RW	stc3_init_value_high	time stame initial value, when load =1, this value is loaded to time stamp high 32bit	0x0

STC3_INIT_VALUE_LOW

Offset=0x38, Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31:0	RW	stc3_init_value_low	time stame initial value, when load =1, this value is loaded to time stamp low 32bit	0x0

STC_VI_CAP_0_0_Value_High

Offset=0x40,Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31:0	RWSC	vi_cap_0_0_value_high	csi0 new time stamp high 32bit	0x0

STC_VI_CAP_0_0_Value_low

Offset=0x44,Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31:0	RWSC	vi_cap_0_0_value_low	csi0 new time stamp low 32bit	0x0

STC_VI_CAP_0_1_Value_High

Offset=0x48,Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31:0	RWSC	vi_cap_0_1_value_high	csi0 old time stamp high 32bit	0x0

STC_VI_CAP_0_1_Value_low

Offset=0x4c,Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31:0	RWSC	vi_cap_0_1_value_low	csi0 old time stamp low 32bit	0x0

STC_VI_CAP_1_0_Value_High

Offset=0x50,Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31:0	RWSC	vi_cap_1_0_value_high	csi1 new time stamp high 32bit	0x0

STC_VI_CAP_1_0_Value_low

Offset=0x54,Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31:0	RWSC	vi_cap_1_0_value_low	csi1 new time stamp low 32bit	0x0

STC_VI_CAP_1_1_Value_High

Offset=0x58,Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31:0	RWSC	vi_cap_1_1_value_high	csi1 old time stamp high 32bit	0x0

STC_VI_CAP_1_1_Value_low

Offset=0x5c,Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31:0	RWSC	vi_cap_1_1_value_low	csi1 old time stamp low 32bit	0x0

STC_VI_CAP_2_0_Value_High

Offset=0x60,Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31:0	RWSC	vi_cap_2_0_value_high	csi2 new time stamp high 32bit	0x0

STC_VI_CAP_2_0_Value_low

Offset=0x64,Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31:0	RWSC	vi_cap_2_0_value_low	csi2 new time stamp low 32bit	0x0

STC_VI_CAP_2_1_Value_High

Offset=0x68,Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31:0	RWSC	vi_cap_2_1_value_high	csi2 old time stamp high 32bit	0x0

STC_VI_CAP_2_1_Value_low

Offset=0x6c,Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31:0	RWSC	vi_cap_2_1_value_low	csi2 old time stamp low 32bit	0x0

STC_VO_CAP_0_Value_High

Offset=0x70,Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31:0	RWSC	vo_cap_0_value_high	csi0 new time stamp high 32bit	0x0

STC_VO_CAP_0_Value_low

Offset=0x74,Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31:0	RWSC	vo_cap_0_value_low	csi0 new time stamp low 32bit	0x0

STC_VO_CAP_1_Value_High

Offset=0x78,Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset

Bits	Access	Name	Descriptions	Reset
31:0	RWSC	vo_cap_1_value_high	csi0 old time stamp high 32bit	0x0

STC_VO_CAP_1_Value_low

Offset=0x7c,Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31:0	RWSC	vo_cap_1_value_low	csi0 old time stamp low 32bit	0x0

STC_AUD0_CAP_0_Value_High

Offset=0x80,Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31:0	RWSC	aud0_cap_0_value_high	csi0 new time stamp high 32bit	0x0

STC_AUD0_CAP_0_Value_low

Offset=0x84,Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31:0	RWSC	aud0_cap_0_value_low	csi0 new time stamp low 32bit	0x0

STC_AUD0_CAP_1_Value_High

Offset=0x88,Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31:0	RWSC	aud0_cap_1_value_high	csi0 old time stamp high 32bit	0x0

STC_AUD0_CAP_1_Value_low

Offset=0x8c,Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31:0	RWSC	aud0_cap_1_value_low	csi0 old time stamp low 32bit	0x0

STC_AUD1_CAP_0_Value_High

Offset=0x90,Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31:0	RWSC	aud1_cap_0_value_high	csi0 new time stamp high 32bit	0x0

STC_AUD1_CAP_0_Value_low

Offset=0x94,Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset

Bits	Access	Name	Descriptions	Reset
31:0	RWSC	aud1_cap_0_value_low	csi0 new time stamp low 32bit	0x0

STC_AUD1_CAP_1_Value_High

Offset=0x98, Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31:0	RWSC	aud1_cap_1_value_high	csi0 old time stamp high 32bit	0x0

STC_AUD1_CAP_1_Value_low

Offset=0x9c, Total Reset Value: 0x0000_0000, Write_enable=no

Bits	Access	Name	Descriptions	Reset
31:0	RWSC	aud1_cap_1_value_low	csi0 old time stamp low 32bit	0x0

2.7 Watchdog

2.7.1 overview

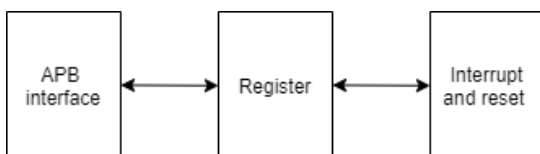
WDT is a programmable Watchdog Timer (WDT) peripheral. This component is an AMBA 2.0-compliant Advanced Peripheral Bus (APB) slave device. WDT can provide interrupt and reset to SOC. Output reset signal can be used to reset CPUs or whole SOC. K230 provides two wdts.

2.7.2 feature

The wdt supports the following features:

- APB3 interface support.
- watchdog counter width of 32 bits.
- Counter counts down from a preset value to 0 to indicate the occurrence of a timeout.
- If a timeout occurs the DW_apb_wdt can perform one of the following operations:
 - Generate a system reset
 - First generate an interrupt and even if it is cleared (or not) by the service routine by the time a second timeout occurs then generate a system reset.
- Programmable reset pulse length.

2.7.3 Function



wdt diagram

2.7.3.1 counter

The wdt counts from a preset (timeout) value in descending order to zero. When the counter reaches zero, depending on the output response mode selected, either a system reset or an interrupt occurs. When the counter reaches zero, it wraps to the selected timeout value and continues decrementing. The user can restart the counter to its

initial value. This is programmed by writing to the restart register at any time. The process of restarting the watchdog counter is sometimes referred to as kicking the dog. As a safety feature to prevent accidental restarts, the value 0x76 must be written to the Current Counter Value Register (WDT_CRR).

2.7.3.2 interrupt

The wdt can be programmed to generate an interrupt (and then a system reset) when a timeout occurs. When a 1 is written to the response mode field (RMOD, bit 1) of the Watchdog Timer Control Register (WDT_CR), the wdt generates an interrupt. When WDT_NEW_RMOD = 0, if the interrupt is not cleared by the time a second timeout occurs, then it generates a system reset. When WDT_NEW_RMOD = 1, even if the interrupt is cleared by the time a second timeout occurs, it generates a system reset.

2.7.3.3 system reset

When a 0 is written to the output response mode field (RMOD, bit 1) of the Watchdog Timer Control Register (WDT_CR), the DW_apb_wdt generates a system reset when a timeout occurs.

System Reset When WDT_NEW_RMOD = 1 wdt can be programmed to generate the wdt_intr (and then a system reset) when a timeout occurs, that is when the response mode field of the Watchdog Timer Control Register (WDT_CR.RMOD) is set to 1.

When WDT_NEW_RMOD is set to 1, irrespective of whether the wdt_intr is cleared (by reading the WDT_EOI register) and if the second timeout occurs, wdt generates a system reset. For subsequent timeouts, both the interrupt and the system reset are generated. However, this system reset generation can be avoided by performing a restart, that is by writing 0x76 into the WDT_CRR register before the second or subsequent timeout occurs. This behavior of a system reset (when WDT_NEW_RMOD = 1) is enabled only if the Response Mode is set to 1 through programming the WDT_CR.RMOD register bit to 1.

System Reset When WDT_NEW_RMOD = 0

When WDT_NEW_RMOD is set to 0, which is the default system reset behavior in which wdt generates a system reset, if the interrupt is not cleared by the time a second timeout occurs.

2.7.3.4 Reset Pulse Length

The reset pulse length is the number of pclk cycles for which a system reset is asserted. When a system reset is generated, it remains asserted for the number of cycles specified by the reset pulse length plus two cycles of synchronization delay.

2.7.4 operating mode

Default operating :

- Program WDT_TORR. Select required timeout period.
- Program WDT_CR. Set reset pulse length, response mode, and enable WDT.
- Write 0x76 to WDT_CRR.
- When wdt counter decrease to 0, assert interrupt.
- Can clear interrupt by reading WDT_EOI or restarting (kicking) the counter by writing 0x76 to WDT_CRR.
- Counter wraps to selected timeout period and continues to decrement.
- When wdt counter decrease to 0 and interrupt has not been cleared, assert system reset.

2.7.5 register summary

Register	Offset	Description	page

Register	Offset	Description	page
WDT_CR	0x0	Control Register. WDT_DFLT_RPL (3 bits), WDT_DFLT_RMOD (1 bit) and WDT_ALWAYS_EN...	page WDT_CR
WDT_TORR	0x4	Timeout Range Register	page WDT_TORR
WDT_CCVR	0x8	Current Counter Value Register.	page WDT_CCVR
WDT_CRR	0xc	Counter Restart Register	page WDT_CRR
WDT_STAT	0x10	Interrupt Status Register	page WDT_STAT
WDT_EOI	0x14	Interrupt Clear Register	page WDT_EOI
WDT_PROT_LEVEL	0x1c	WDT Protection level register	page WDT_PROT_LEVEL
WDT_COMP_PARAM_5	0xe4	Component Parameters Register 5 Note: This is a constant read-only register that contains encoded information about the component's parameter settings.	page WDT_COMP_PARAM_5
WDT_COMP_PARAM_4	0xe8	Component Parameters Register 4 Note: This is a constant read-only register that contains encoded information about the component's parameter settings.	page WDT_COMP_PARAM_4
WDT_COMP_PARAM_3	0xec	Component Parameters Register 3 Note: This is a constant read-only register that contains encoded information about the component's parameter settings.	page WDT_COMP_PARAM_3
WDT_COMP_PARAM_2	0xf0	Component Parameters Register 2 Note: This is a constant read-only register that contains encoded information about the component's parameter settings.	page WDT_COMP_PARAM_2
WDT_COMP_PARAM_1	0xf4	Component Parameters Register 1 Note: This is a constant read-only register that contains encoded information about the component's parameter settings.	page WDT_COMP_PARAM_1
WDT_COMP_VERSION	0xf8	Component Version Register	page WDT_COMP_VERSION
WDT_COMP_TYPE	0xfc	Component Type Register	page WDT_COMP_TYPE

2.7.6 register description

WDT_CR

Offset=0x00,

Bits	Name	Reset	Access	Descriptions

Bits	Name	Reset	Access	Descriptions
31:6	RSVD_WDT_CR	0	R	WDT_CR[31:6] Reserved bits and read as zero (0). Exists: Always
5	NO_NAME	0	R/W	Redundant R/W bit. Included for ping test purposes, as it is the only R/W register bit that is in every configuration of the wdt. Exists: Always
4:2	RPL	3'h3	R/W	Reset pulse length. Writes have no effect when the configuration parameter WDT_HC_RPL is 1, making the register bits read-only. 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. Note: When WDT_SYNC_CLK_MOPE_ENABLE = 1, the total reset pulse length also includes the reset synchronization delay and the time taken for pclk to be made available. Values: 0x0 (PCLK_CYCLES2): 2 pclk cycles 0x1 (PCLK_CYCLES4): 4 pclk cycles 0x2 (PCLK_CYCLES8): 8 pclk cycles 0x3 (PCLK_CYCLES16): 16 pclk cycles 0x4 (PCLK_CYCLES32): 32 pclk cycles 0x5 (PCLK_CYCLES64): 64 pclk cycles 0x6 (PCLK_CYCLES128): 128 pclk cycles 0x7 (PCLK_CYCLES256): 256 pclk cycles Exists: Always
1	RMOD	0	R/W	Response mode. Writes have no effect when the parameter WDT_HC_RMOD= 1, thus this register becomes read-only. Selects the output response generated to a timeout. Values: 0x0 (RESET): Generate a system reset 0x1 (INTERRUPT): First generate an interrupt and even if it is cleared by the time a second timeout occurs then generate a system reset Exists: Always
0	WDT_EN	0	R/W	WDT enable. When the configuration parameter WDT_ALWAYS_EN = 0, this bit can be set; otherwise, it is read-only. This bit is used to enable and disable the DW_apb_wdt. When disabled, the counter does not decrement. Thus, no interrupts or system resets are generated. The wdt is used to prevent system lock-up. To prevent a software bug from disabling the wdt, once this bit has been enabled, it can be cleared only by a system reset. Values: 0x0 (DISABLED): Watchdog timer disabled 0x1 (ENABLED): Watchdog timer enabled

Offset=0x04

Bits	Name	Reset	Access	Descriptions
31:4	Reserved	0	R	WDT_TORR[31:24] Reserved and read as zero (0). Exists: Always
3:0	TOP	4'h0	R/W	Timeout period. Writes have no effect when the configuration parameter WDT_HC_TOP = 1, thus making this register read-only. 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 the WDT_CNT_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: Where i = TOP and t = timeout period For i = 0 to 15 if WDT_USE_FIX_TOP == 1 t = 2(16 + i) else t = WDT_USER_TOP_(i) Values: 0x0 (USER0_OR_64K): Time out of WDT_USER_TOP_0 or 64K Clocks 0x1 (USER1_OR_128K): Time out of WDT_USER_TOP_1 or 128K Clocks 0x2 (USER2_OR_256K): Time out of WDT_USER_TOP_2 or 256K Clocks 0x3 (USER3_OR_512K): Time out of WDT_USER_TOP_3 or 512K Clocks 0x4 (USER4_OR_1M): Time out of WDT_USER_TOP_4 or 1M Clocks 0x5 (USER5_OR_2M): Time out of WDT_USER_TOP_5 or 2M Clocks 0x6 (USER6_OR_4M): Time out of WDT_USER_TOP_6 or 4M Clocks 0x7 (USER7_OR_8M): Time out of WDT_USER_TOP_7 or 8M Clocks 0x8 (USER8_OR_16M): Time out of WDT_USER_TOP_8 or 16M Clocks 0x9 (USER9_OR_32M): Time out of WDT_USER_TOP_9 or 32M Clocks 0xa (USER10_OR_64M): Time out of WDT_USER_TOP_10 or 64M Clocks 0xb (USER11_OR_128M): Time out of WDT_USER_TOP_11 or 128M Clocks 0xc (USER12_OR_256M): Time out of WDT_USER_TOP_12 or 256M Clocks 0xd (USER13_OR_512M): Time out of WDT_USER_TOP_13 or 512M Clocks 0xe (USER14_OR_1G): Time out of WDT_USER_TOP_14 or 1G Clocks 0xf (USER15_OR_2G): Time out of WDT_USER_TOP_15 or 2G Clocks Exists: Always

WDT_CCVR

Offset=0x08

Bits	Name	Reset	Access	Descriptions

Bits	Name	Reset	Access	Descriptions
31:0	WDT_CCVR	32'hFFFF	R	WDT Current Counter Value Register. This register, when read, is the current value of the internal counter. This value is read coherently when ever it is read, which is relevant when the APB_DATA_WIDTH is less than the counter width. Exists: Always Volatile: true

WDT_CRR

Offset=0x0c

Bits	Name	Reset	Access	Descriptions
31:8	RSVD_WDT_CRR	0	W	WDT_CRR[31:24] Reserved bits - Write Only Exists: Always
7:0	WDT_CRR	0	W	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. Values: - 0x76 (RESTART): Watchdog timer restart command Exists: Always

WDT_STAT

Offset=0x10

Bits	Name	Reset	Access	Descriptions
31:1	RSVD_WDT_STAT	0	R	WDT_STAT[31] Reserved bits - Read Only Exists: Always Volatile: true
0	WDT_STAT	0	R	Interrupt status register This register shows the interrupt status of the WDT. Values: ■ 0x0 (INACTIVE): Interrupt is inactive ■ 0x1 (ACTIVE): Interrupt is active regardless of polarity Exists: Always Volatile: true

WDT_EOI

Offset=0x14

Bits	Name	Reset	Access	Descriptions
31:1	RSVD_WDT_EOI	0	R	RSVD_WDT_EOI[31] Reserved bits and read as zero (0). Exists: Always Volatile: true

Bits	Name	Reset	Access	Descriptions
0	WDT_EOI	0	R	Interrupt Clear Register. Clears the watchdog interrupt. This can be used to clear the interrupt without restarting the watchdog counter. Exists: Always Volatile: true

WDT_PROT_LEVEL

Offset=0x1c

Bits	Name	Reset	Access	Descriptions
31:3	RSVD_WDT_PROT_LEVEL	0	R	WDT_PROT_LEVEL[31:29]Reserved field- read-only Exists: Always
2:0	WDT_PROT_LEVEL	3'h2	R/W	Protection level register. Enabling protection on any of its three bits would require a match on the input PPROT signal to gain access to protected registers of the WDT. Exists: Always

WDT_COMP_PARAM_5

Offset=0xe4

Bits	Name	Reset	Access	Descriptions
31:0	CP_WDT_USER_TOP_MAX	32'h0	R	Upper limit of Timeout Period parameters. Exists: Always

WDT_COMP_PARAM_4

Offset=0xe8

Bits	Name	Reset	Access	Descriptions
31:0	CP_WDT_USER_TOP_INIT_MAX	32'h0	R	Upper limit of Initial Timeout Period parameters. Exists: Always

WDT_COMP_PARAM_3

Offset=0xec

Bits	Name	Reset	Access	Descriptions
31:0	CD_WDT_TOP_RST	32'h0	R	The value of this register is derived from the WDT_TOP_RST parameter. Exists: Always

WDT_COMP_PARAM_2

Offset=0xf0

Bits	Name	Reset	Access	Descriptions

Bits	Name	Reset	Access	Descriptions
31:0	CP_WDT_CNT_RST	32'hffff	R	The value of this register is derived from the WDT_RST_CNT parameter. Exists: Always

WDT_COMP_PARAM_1

Offset=0xf4

Bits	Name	Reset	Access	Descriptions
31:29	RSVD_31_29	0	R	WDT_COMP_PARAM_1[31:29]Reserved bits and read as zero (0). Exists: Always
28:24	WDT_CNT_WIDTH	5'h10	R	The Watchdog Timer counter width. Exists: Always
23:20	WDT_DFLT_TOP_INIT	4'h0	R	Describes the initial timeout period that is available directly after reset. It controls the reset value of the register. Exists: Always
19:16	WDT_DFLT_TOP	4'h0	R	Selects the timeout period that is available directly after reset. It controls the reset value of the register. Exists: Always
15:13	RSVD_15_13	3'h0	R	WDT_COMP_PARAM_1[15:13] Reserved bits and read as zero (0). Exists: Always
12:10	WDT_DFLT_RPL	3'h3	R	The reset pulse length that is available directly after reset. Exists: Always
9:8	APB_DATA_WIDTH	2'h2	R	Width of the APB Data Bus to which this component is attached. Values: ■ 0x0 (APB_8BITS): APB data width is 8 bits ■ 0x1 (APB_16BITS): APB data width is 16 bits ■ 0x2 (APB_32BITS): APB data width is 32 bits Exists: Always
7	WDT_PAUSE	1'h0	R	Configures the peripheral to have a pause enable signal (pause) on the interface that can be used to freeze the watchdog counter during pause mode. Values: ■ 0x0 (DISABLED): Pause enable signal is non existent ■ 0x1 (ENABLED): Pause enable signal is included Exists: Always

Bits	Name	Reset	Access	Descriptions
6	WDT_USE_FIX_TOP	1'h1	R	When this parameter is set to 1, timeout period range is fixed. The range increments by the power of 2 from 2^{16} to $2^{(WDT\_CNT\_WIDTH-1)}$. When this parameter is set to 0, the user must define the timeout period range (2^8 to $2^{(WDT\_CNT\_WIDTH)-1}$) using the WDT_USER_TOP_(i) parameter. Values: ■ 0x0 (USERDEFINED): User must define timeout values ■ 0x1 (PREDEFINED): Use predefined timeout values Exists: Always
5	WDT_HC_TOP	1'h0	R	When set to 1, the selected timeout period(s) is set to be hard coded. Values: ■ 0x0 (PROGRAMMABLE): Timeout period is programmable ■ 0x1 (HARDCODED): Timeout period is hard coded Exists: Always
4	WDT_HC_RPL	1'h0	R	Configures the reset pulse length to be hard coded. Values: ■ 0x0 (PROGRAMMABLE): Reset pulse length is programmable ■ 0x1 (HARDCODED): Reset pulse length is hardcoded Exists: Always
3	WDT_HC_RMOD	1'h0	R	Configures the output response mode to be hard coded. Values: ■ 0x0 (PROGRAMMABLE): Output response mode is programmable ■ 0x1 (HARDCODED): Output response mode is hard coded Exists: Always
2	WDT_DUAL_TOP	1'b0	R	When set to 1, includes a second timeout period that is used for initialization prior to the first kick. Values: ■ 0x0 (DISABLED): Second timeout period is not present ■ 0x1 (ENABLED): Second timeout period is present Exists: Always
1	WDT_DFLT_RMOD	1'b0	R	Describes the output response mode that is available directly after reset. Indicates the output response the WDT gives if a zero count is reached; that is, a system reset if equals 0 and an interrupt followed by a system reset, if equals 1. If WDT_HC_RMOD is 1, then default response mode is the only possible output response mode. Values: ■ 0x0 (DISABLED): System reset only ■ 0x1 (ENABLED): Interrupt and system reset Exists: Always

Bits	Name	Reset	Access	Descriptions
0	WDT_ALWAYS_EN	1'b0	R	Configures the WDT to be enabled from reset. If this setting is 1, the WDT is always enabled and a write to the WDT_EN field (bit 0) of the Watchdog Timer Control Register (WDT_CR) to disable it has no effect. Values: ■ 0x0 (DISABLED): Watchdog timer disabled on reset ■ 0x1 (ENABLED): Watchdog timer enabled on reset Exists: Always

WDT_COMP_VERSION

Offset=0xf8

Bits	Name	Reset	Access	Descriptions
31:0	WDT_COMP_VERSION	32'h3131_302a	R	ASCII value for each number in the version, followed by . <i>For example, 32_30_31_2A represents the version 2.01.</i> Exists: Always

WDT_COMP_TYPE

Offset=0xfc

Bits	Name	Reset	Access	Descriptions
31:0	WDT_COMP_TYPE	32'h4457_0120	R	Constant value 0x44570120 Exists: Always

2.8 RTC

2.8.1 Overview

Function Real Time Clock (RTC), mainly includes timing (calendar), timing interrupt (timing alarm clock), cycle interrupt (hourly alarm clock) these three parts of the function, among which the calendar function we use the most. RTC contains two 32bit time counting registers that can output the year (including the leap year identifier), month, day, week, hour, minute and second. RTC timing requires the use of an external crystal oscillator clock source (crystal oscillator frequency: $32768\text{Hz}=2^{15}\text{Hz}$). RTC, placed in PMU, has a separate power supply system and a separate reset. It can continue to work in the case of power failure of the host (normal power failure of the system and abnormal shutdown of the system), so it can provide accurate real-time time for people or accurate time benchmark for the system. Its typical application scenarios include: calendar, alarm clock, hour ticking, etc.

2.8.2 Feature

The RTC has the following features:

- typical working frequency: external clock source 32768 Hz
- involved in the work mode: normal working mode, independent power supply working mode
- interface type: APB3.0 interface
- support calendar initial value configuration, configuration items including year, month, day, week, when, minutes and seconds

- can read less than 1 second counter count, and the initial values can be configured, minimum scale unit for each calibration cycle of the crystals
- support disruption: a timer interrupt (configurable project includes year, month, day, week,, minutes and seconds), the interrupt for level interrupts, need to register to clear; Periodic interruption (including annual, monthly, daily, weekly, every hour, every minute, every second, every 1/8 of a second, 1/64 of a second trigger interrupt) the interrupts for pulse, pulse width is two 32768 hz clock cycles
- typical application scenario: calendar, timing interrupt trigger interrupt trigger, cycle
- other features: a regular in the system domain, need external power supply; A low speed clock of 32768Hz is required

2.8.3 Block Diagram

Figure 2-8-1 shows the following functional groupings of the main interfaces to the RTC block

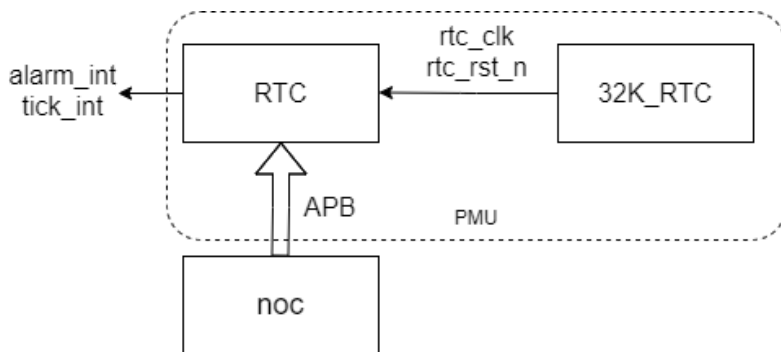


Figure 2-8-1 RTC Block Diagram

As shown in the Figure 2-8-1, the RTC module needs a high-precision 32.768KHz clock as the basic clock of real time timing. Through the APB bus interface, the initial value configuration of its internal time register, and the interrupt control configuration of its timing time register and interrupt control register can be carried out. RTC can produce periodic and periodic interrupts. RTC is placed in PMU, and LDO provides 0.8V power supply to it, so that it can work in normal mode and independent power supply mode respectively (the two modes have no perception difference for RTC). In normal working mode, the system does not power off, the pclk clock is effective, the RTC register can be configured; In the independent power supply mode, the system is powered off and the pclk clock fails. RTC only retains the functions of real-time timing and interrupt, and the register cannot be configured.

2.8.4 Function Description

RTC, as shown in the Figure 2-8-2, consists of three sub-modules: RTC_register, RTC_Timer and RTC_interrupt. The RTC_register module is a register module of RTC. It mainly interacts with the system through the APB bus and stores the configuration information of RTC delivered by the system (including the initialization time of timed RTC, timing time of interrupt and interrupt control information). When the system is powered off, the registers inside the module cannot be rewritten.

The RTC_Timer module is the main functional module of RTC, which is used to realize the real-time timing function and complete the counting of year, month, day, week, hour, minute and second. RTC The minimum unit of timing is one cycle of a 32768Hz clock. RTC The function of automatically setting a leap in leap years is supported.

The RTC_interrupt module is the interrupt generation unit of RTC, which mainly generates periodic interrupt and periodic interrupt. The timed interrupt supports each independent timed interrupt of the year, month, day, week, hour, minute, and second, or several combinations of timed interrupts. The year triggers the comparison of leap year flags, and the century triggers the comparison of the preceding and later century flags. Periodic interrupt can be triggered every year, month, day, week, hour, minute, second, 1/8 second, and 1/64 second.

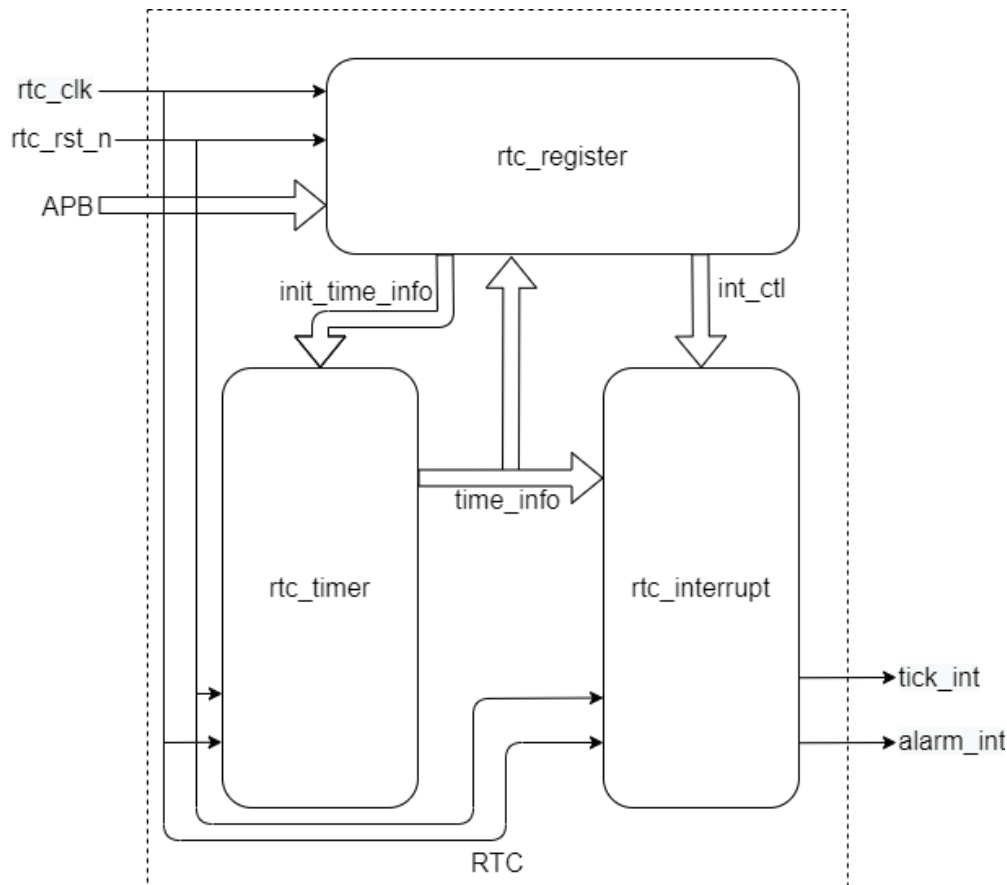


Figure 2-8-2 RTC Architecture Diagram

RTC provides three functions: perpetual calendar, periodic interrupt and periodic interrupt:

1. Calendar function is mainly realized by basic counter, second counter, minute counter, hour counter, day counter, week counter, month counter and year counter. The basic counter is controlled by 323768Hz clock, each clock increases by 1; The second counter is increased by 1 when the base counter returns to 0. The minute counter increases by 1 when the second counter returns to 0. Similarly, other counters are also carried in this way to achieve real-time timing. According to the leap algorithm, the leap register can be automatically set in leap year.
2. The periodic interrupt and periodic interrupt functions are generated by the interrupt unit.
 - Timing interrupt: The interrupt control register first determines which values in the preset value register of timing alarm clock (including: year, month, day, week, hour, minute, second) are compared with the value in the time and date register. When matched, the corresponding timing alarm clock interrupt is triggered. Finally, all timing alarm clock interrupts are matched as the output of the total alarm clock interrupt, which is enabled and cleared by the system.
 - Cycle interrupt function: When the year, month, day, week, hour, minute and second increase by 1 hour, every 1/8 second and every 1/64 second, the corresponding cycle interrupt will be triggered respectively. Then the interrupt control register will decide which time interrupt will be output as the source of cycle interrupt. The interrupt is enabled by the system and maintains two cycles.

2.8.5 Register Description

2.8.5.1 Register Summary

The RTC base addresses is 0x9100_0C00.

Table 2-8-1 Summary of RTC registers

Offset Address	Register	Description	Page
0x00	DATE	It is used to store the time register of century (including pre-century...	2-8-2
0x04	TIME	A time register used to store weeks, hours, minutes and seconds	2-8-3
0x08	ALARM_DATE	A preset time register used to store the century (including pre-century...	2-8-4
0x0C	ALARM_TIME	A preset time register that stores weeks, hours, minutes, and seconds...	2-8-5
0x10	COUNT	A register used to store the count value and maximum count comparison...	2-8-6
0x14	INT_CTRL	A register used to store RTC interrupt control information	2-8-7

2.8.5.2 Register Description

2.8.5.2.1 DATE

- Name: Date register
- Description: It is used to store the time register of century (including pre-century and post-century flag bits), year (including leap year leap flag bits), month, and day
- Reset Value: 0x1_0101
- Offset Address: 0x00

Table [2-8-2](#) Fields for Register: DATE

Bits	Access	Name	Description	Reset Value
31	R	RSVD	Reserved	0x0
30:24	R/W	year_H	Annual high (thousand, hundred) digit count value, 0~127	0x0
23	R/W	leap_year	Leap year leap flag bit: 0: non-leap year; 1: Leap year	0x0
22:16	R/W	year_L	The low (ten or ten) digit value of the year, 1~100	0x1
15:12	R	RSVD	Reserved	0x0
11:8	R/W	month	Monthly value, 1~12	0x1
7:5	R	RSVD	Reserved	0x0
4:0	R/W	day	Daily value, 1~31	0x1

2.8.5.2.2 TIME

- Name: Time register
- Description: A time register used to store weeks, hours, minutes and seconds
- Reset Value: 0x00
- Offset Address: 0x04

Table [2-8-3](#) Fields for Register: TIME

Bits	Access	Name	Description	Reset Value
31:27	R	RSVD	Reserved	0x0
26:24	R/W	week	Weekly value, 0~6	0x0

Bits	Access	Name	Description	Reset Value
23:21	R	RSVD	Reserved	0x0
20:16	R/W	hour	Hour value, 0~23	0x0
15:14	R	RSVD	Reserved	0x0
13:8	R/W	minute	The value of minutes, 0~59	0x0
7:6	R	RSVD	Reserved	0x0
5:0	R/W	second	Number of seconds, 0~59	0x0

2.8.5.2.3 ALARM_DATE

- Name: Preset date register
- Description: A preset time register used to store the century (including pre-century and post-century flag bits), year (including leap year leap flag bits), month, and day of the preset timing interrupt
- Reset Value: 0x1_0101
- Offset Address: 0x08

Table 2-8-4 Fields for Register: ALARM_DATE

Bits	Access	Name	Description	Reset Value
31	R	RSVD	Reserved	0x0
30:24	R/W	alarm_year_H	Timing interrupt preset the high (thousand, hundred) digit value of the year, 0~127	0x0
23	R	RSVD	Reserved	0x0
22:16	R/W	alarm_year_L	Timing interrupt preset the low (ten or ten) digit value of the year, 1~100	0x1
15:12	R	RSVD	Reserved	0x0
11:8	R/W	alarm_month	Timing interrupt preset value in monthly, 1~12	0x1
7:5	R	RSVD	Reserved	0x0
4:0	R/W	alarm_day	Timing interrupt preset value in daily, 1~31	0x1

2.8.5.2.4 ALARM_TIME

- Name: Preset time register
- Description: A preset time register that stores weeks, hours, minutes, and seconds of preset timing interrupts
- Reset Value: 0x00
- Offset Address: 0x0C

Table 2-8-5 Fields for Register: ALARM_TIME

Bits	Access	Name	Description	Reset Value
31:27	R	RSVD	Reserved	0x0
26:24	R/W	alarm_week	Timing interrupt preset value in weekly, 0~6	0x0
23:21	R	RSVD	Reserved	0x0

Bits	Access	Name	Description	Reset Value
20:16	R/W	alarm_hour	Timing interrupt preset value in hour, 0~23	0x0
15:14	R	RSVD	Reserved	0x0
13:8	R/W	alarm_minute	Timing interrupt preset value in minutes, 0~59	0x0
7:6	R	RSVD	Reserved	0x0
5:0	R/W	alarm_second	Timing interrupt preset value in seconds, 0~59	0x0

2.8.5.2.5 COUNT

- Name: Count value register
- Description: A register used to store the count value and maximum count comparison value of the base timer
- Reset Value: 0x7FFF_0000
- Offset Address: 0x10

Table 2-8-6 Fields for Register: COUNT

Bits	Access	Name	Description	Reset Value
31	R	RSVD	Reserved	0x0
30:16	R/W	sum_count	The preset maximum count value of the basic counter, typical value is 32,767	0x7FFF
15	R	RSVD	Reserved	0x0
14:0	R/W	curr_count	Current count value of basic counter, 0~32767	0x0

2.8.5.2.6 INT_CTRL

- Name: Interrupt control register
- Description: A register used to store RTC interrupt control information
- Reset Value: 0x00
- Offset Address: 0x14

Table 2-8-7 Fields for Register: INT_CTRL

Bits	Access	Name	Description	Reset Value
31	R	RSVD	Reserved	0x0
30	R/W	year_cmp	Annual value comparison triggers a timed interrupt	0x0
29	R/W	month_cmp	The monthly value comparison triggers a fixed interruption	0x0
28	R/W	day_cmp	Day value comparison triggers a timed interrupt	0x0
27	R/W	week_cmp	Weekly value comparison triggers a timed interrupt	0x0
26	R/W	hour_cmp	Time interrupt triggered by hour value comparison	0x0
25	R/W	minute_cmp	Minute value comparison triggers a timed interrupt	0x0
24	R/W	second_cmp	The second value comparison triggers a timed interrupt	0x0
23:18	R	RSVD	Reserved	0x0
17	RWSC	alarm_clr	Timed Interrupt Clear	0x0

Bits	Access	Name	Description	Reset Value
16	R/W	alarm_en	Timed interrupt enable	0x0
15:13	R	RSVD	Reserved	0x0
12:9	R/W	tick_sel	Cycle interrupt trigger source selection: 0000: 1/64 second interrupt 0001: 1/8 second interrupt 0010: 1 second interrupt 0011: 1 minute interrupt 0100: 1 hour interrupt 0101: 1 week interrupt 0110: 1 day interrupt 0111: 1 month interrupt 1000: 1 whole year interrupt	0x0
8	R/W	tick_en	Cycle interrupt enable	0x0
7:2	R	RSVD	Reserved	0x0
1	R/W	timer_r_en	Timer current value read out enable	0x0
0	R/W	timer_w_en	Timer initial value write enable	0x0

2.8.6 Application Notes

1. A note on the time register

The initial time of RTC can be configured via a register. The low year_L is timed from 1 to 100 for the first century; The high of the year (year_H) starts at 0 and ends at a maximum of 127. For example, in 2000, the low of the year is 100, and the high of the year is 19; For example, in 2005, the low of the year is 5 and the high of the year is 20.

2. A note on the interrupt register

RTC A timed interrupt can be triggered by the year, month, day, week, hour, minute and second alone or by several combinations of them. It is a level interrupt and requires register clearing. The trigger source of cycle interrupt can be selected as year, month, day, week, hour, minute, second, 1/8 second and 1/64 second respectively, which is pulse interrupt. The pulse width is two 32768Hz clock cycles.

3. A note on the base counter registers

The count clock of the basic counter is 32768Hz, so the count from 0 to 32767 is 1 second. The initial count value can be configured, and the maximum count value is generally configured as 32767.

4. Configuration flow about registers

When configuring the register of RTC, you need to configure the ALARM_DATE and ALARM_TIME registers, the interrupt control (INT_CTRL) register, the initial TIME (DATE and TIME) register, and the basic counter (COUNT) register.

NOTE: After the RTC module is powered on for the first time, it usually keeps working and does not reset.

2.9 Mailbox

2.9.1 Overview

This article describes the design of the K230, which is located within SYSCTL and involves the following functions.

- (1)cpu0_sys and cpu1_sys shake hands with each other and share storage resources through a hardware mutual exclusion (interlock) mechanism.
- (2) The cpu0_sys and cpu1_sys modules perform register configuration for certain submodules.
- (3) Query reading of certain submodule status information by cpu0_sys and cpu1_sys.

2.9.2 Features

- (1)Bus interface: Mailbox with APB 3.0 interface.
- (2)Data bit width: 32bit, configurable via APB bus.
- (3)Configuring the clock: configuring the clock to be the pclk clock.
- (4)Support for hardware locks.
- (5)Support for cpu0_sys and cpu1_sys to send interrupts to each other.
- (6)Support for register configuration and status reading of submodules.

2.9.3 Function Description

Figure 2-9-1 Block diagram of mailbox structure

As shown above, the Mailbox clock reset signal comes from mailbox_pclk and mailbox_presetn of sysctrl, and the default operating frequency is 100MHz. Mailbox is composed of interface control module, mutual exclusion lock module and RegFile module. The interface control module is responsible for communicating with the bus interface; the mutual exclusion lock module provides mutual exclusion lock service to cpu0_sys and cpu1_sys; RegFile is responsible for signal configuration or state latching of each module.

2.9.4 Operation Mode

The mailbox modules can all be configured with information and sub-module status reading via APB3.0 bus.

2.9.5 Register Summary

The base address of Mailbox is 0x9110_4000. Table 2- Summary of Mailbox registers

Offset Address	Register	Description
0x0000	CPU2DSP_INT_EN0	cpu0_sys2cpu1_sys interrupt enable register 0
0x0004	CPU2DSP_INT_SET0	cpu0_sys2cpu1_sys interrupt setting register 0
0x0008	CPU2DSP_INT_CLEAR0	cpu0_sys2cpu1_sys interrupt clear register 0
0x000C	CPU2DSP_INT_STATUS0	cpu0_sys2cpu1_sys interrupt status register 0
0x0010	CPU2DSP_INT_ERR0	cpu0_sys2cpu1_sys interrupt error register 0
0x0014	DSP2CPU_INT_EN0	cpu1_sys2cpu0_sys interrupt enable register 0
0x0018	DSP2CPU_INT_SET0	cpu1_sys2cpu0_sys interrupt setting register 0
0x001C	DSP2CPU_INT_CLEAR0	cpu1_sys2cpu0_sys interrupt clear register 0
0x0020	DSP2CPU_INT_STATUS0	cpu1_sys2cpu0_sys interrupt status register 0
0x0024	DSP2CPU_INT_ERR0	cpu1_sys2cpu0_sys interrupt error register 0

Offset Address	Register	Description
0x0028	CPU2DSP_INT_EN1	Same as group 0 register description
0x002C	CPU2DSP_INT_SET1	Same as group 0 register description
0x0030	CPU2DSP_INT_CLEAR1	Same as group 0 register description
0x0034	CPU2DSP_INT_STATUS1	Same as group 0 register description
0x0038	CPU2DSP_INT_ERR1	Same as group 0 register description
0x003C	DSP2CPU_INT_EN1	Same as group 0 register description
0x0040	DSP2CPU_INT_SET1	Same as group 0 register description
0x0044	DSP2CPU_INT_CLEAR1	Same as group 0 register description
0x0048	DSP2CPU_INT_STATUS1	Same as group 0 register description
0x004C	DSP2CPU_INT_ERR1	Same as group 0 register description
0x0050	CPU2DSP_INT_EN2	Same as group 0 register description
0x0054	CPU2DSP_INT_SET2	Same as group 0 register description
0x0058	CPU2DSP_INT_CLEAR2	Same as group 0 register description
0x005C	CPU2DSP_INT_STATUS2	Same as group 0 register description
0x0060	CPU2DSP_INT_ERR2	Same as group 0 register description
0x0064	DSP2CPU_INT_EN2	Same as group 0 register description
0x0068	DSP2CPU_INT_SET2	Same as group 0 register description
0x006C	DSP2CPU_INT_CLEAR2	Same as group 0 register description
0x0070	DSP2CPU_INT_STATUS2	Same as group 0 register description
0x0074	DSP2CPU_INT_ERR2	Same as group 0 register description
0x0078	CPU2DSP_INT_EN3	Same as group 0 register description
0x007C	CPU2DSP_INT_SET3	Same as group 0 register description
0x0080	CPU2DSP_INT_CLEAR3	Same as group 0 register description
0x0084	CPU2DSP_INT_STATUS3	Same as group 0 register description
0x0088	CPU2DSP_INT_ERR3	Same as group 0 register description
0x008C	DSP2CPU_INT_EN3	Same as group 0 register description
0x0090	DSP2CPU_INT_SET3	Same as group 0 register description
0x0094	DSP2CPU_INT_CLEAR3	Same as group 0 register description
0x0098	DSP2CPU_INT_STATUS3	Same as group 0 register description
0x009C	DSP2CPU_INT_ERR3	Same as group 0 register description
0x00A0	MAILBOX0	MAILBOX interlock register 0
0x00A4	MAILBOX1	MAILBOX interlock register 1
0x00A8	MAILBOX2	MAILBOX interlock register 2

Offset Address	Register	Description
0x00AC	MAILBOX3	MAILBOX interlock register 3
0x00B0-	MAILBOX4-	MAILBOX interlock register 4 -
0x0298	MAILBOX126	MAILBOX interlock register 126
0x029C	MAILBOX127	MAILBOX interlock register 127
0x02A0	UART_LP_REQ_SCLK	UART SLCK Low Power Status Register
0x02A4	UART_LP_REQ_PCLK	UART PLCK low-power status register
0x02A8	QOS_CTRL0	QOS control register 0
0x02AC	QOS_CTRL1	QOS control register 1
0x02B0	QOS_CTRL2	QOS control register 2
0x02B4	QOS_CTRL3	QOS control register 3
0x02B8	QOS_CTRL4	QOS control register 4
0x02BC	NOC_WR_BUFFERABLE_CTRL	Noc bufferable control register
0x02C0	DDR_URGENT_MASK	DDR configuration register 0
0x02C4	DDR_BypassOutEnDAT	DDR configuration register 1
0x02C8	DDR_BypassInDataDat	DDR configuration register 2
0x02CC	DDR_BypassDat_H	DDR configuration register 3
0x02D0	DDR_BypassOutEnAC	DDR configuration register 4
0x02D4	DDR_BypassInDataAC	DDR configuration register 5
0x02D8	DDR_BypassInDataAC	DDR configuration register 6
0x02DC	MIPI_TX_BISTON	
0x02E0	SSI_SLEEP	SSI Low Power Register
0x02E4	PVDET_EN	PVDET enable register
0x02E8	PVDET_OUT_DET	PVDET output detection register
0x02EC	OTP_HS_MSC_SEL	
0x02F0	DDRC_SELFREF_TYPE	
0x02F4	CODEC_AD2DA_LOOP	
0x02F8	TIMER_EN	

2.9.6 Mailbox Register Description

CPU2DSP_INT_EN0

cpu0_sys2cpu1_sys interrupt enable register 0.

Offset:0x0 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:16	RW	intr_raw_en	1: enable 0:disable	0

Bits	Access	Name	Description	Reset
15:2	Reserved	Reserved	Reserved	-
1	RW	intr_soft_rst	Mutual interrupt global reset, clear all interrupt counts for CPU2DSP and DSP2CPU 1: Soft reset is valid 0: Reset is invalid	0
0	RW	intr_en	1:enable 0:disable	0

CPU2DSP_INT_SET0

cpu0_sys2cpu1_sys interrupt setting register 0.

Offset:0x4 Total Reset Value:-

Bits	Access	Name	Description	Reset
31:0	W	intr set	Take the value range [0, 15]. The interrupt status register CPU2DSP_INT_STATUS corresponds to the interrupt count from intr_raw0 to intr_raw15, and the interrupt count is increased by 1 for each setting.	-

CPU2DSP_INT_CLEAR0

cpu0_sys2cpu1_sys interrupt clear register 0.

Offset:0x8 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:0	W	intr set	Take the value range [0, 15]. The interrupt count of CPU2DSP_INT_STATUS corresponds to the interrupt count of intr_raw0 ~ intr_raw15, respectively, and the interrupt count is decremented by 1 for each setting.	-

CPU2DSP_INT_STATUS0

cpu0_sys2cpu1_sys interrupt status register 0.

Offset:0xc Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:30	R	intr_raw[15]	When writing 15 to 16'h4, the interrupt count adds 1; when writing 15 to 16'h8, the interrupt count decreases 1; when cpu2dsp_intr_soft_rst, the interrupt count clears 0	0
29:28	R	intr_raw[14]	When writing 14 to 16'h4, the interrupt count is added by 1; when writing 14 to 16'h8, the interrupt count is subtracted by 1; when cpu2dsp_intr	0
27:26	R	intr_raw[13]	When writing 13 to 16'h4, the interrupt count is added 1; when writing 13 to 16'h8, the interrupt count is subtracted 1; when cpu2dsp_intr	0
25:24	R	intr_raw[12]	When writing 12 to 16'h4, the interrupt count is added 1; when writing 12 to 16'h8, the interrupt count is subtracted 1; when cpu2dsp_intr	0
23:22	R	intr_raw[11]	When writing 11 to 16'h4, the interrupt count is added 1; when writing 11 to 16'h8, the interrupt count is subtracted 1; when cpu2dsp_intr	0
21:20	R	intr_raw[10]	When writing 10 to 16'h4, the interrupt count is added 1; when writing 10 to 16'h8, the interrupt count is subtracted 1; when cpu2dsp_intr	0

Bits	Access	Name	Description	Reset
19:18	R	intr_raw[9]	When writing 9 to 16'h4, the interrupt count is added 1; when writing 9 to 16'h8, the interrupt count is subtracted 1; when cpu2dsp_intr	0
17:16	R	intr_raw[8]	When writing 8 to 16'h4, the interrupt count is added 1; when writing 8 to 16'h8, the interrupt count is subtracted 1; when cpu2dsp_intr	0
15:14	R	intr_raw[7]	When writing 7 to 16'h4, the interrupt count is added 1; when writing 7 to 16'h8, the interrupt count is subtracted 1; when cpu2dsp_intr	0
13:12	R	intr_raw[6]	When writing 6 to 16'h4, the interrupt count is added 1; when writing 6 to 16'h8, the interrupt count is subtracted 1; when cpu2dsp_intr	0
11:10	R	intr_raw[5]	When writing 5 to 16'h4, the interrupt count is added 1; when writing 5 to 16'h8, the interrupt count is subtracted 1; when cpu2dsp_intr	0
9:8	R	intr_raw[4]	When writing 4 to 16'h4, the interrupt count is added 1; when writing 4 to 16'h8, the interrupt count is subtracted 1; when cpu2dsp_intr	0
7:6	R	intr_raw[3]	When writing 3 to 16'h4, the interrupt count is added 1; when writing 3 to 16'h8, the interrupt count is subtracted 1; when cpu2dsp_intr	0
5:4	R	intr_raw[2]	When writing 2 to 16'h4, the interrupt count is added 1; when writing 2 to 16'h8, the interrupt count is subtracted 1; when cpu2dsp_intr	0
3:2	R	intr_raw[1]	When writing 1 to 16'h4, the interrupt count is added 1; when writing 1 to 16'h8, the interrupt count is subtracted 1; when cpu2dsp_intr	0
1:0	R	intr_raw[0]	When writing 0 to 16'h4, the interrupt count is added 1; when writing 0 to 16'h8, the interrupt count is subtracted 1; when cpu2dsp_intr	0

CPU2DSP_INT_ERR0

cpu0_sys2cpu1_sys interrupt error register 0.

Offset:0x10 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31	R	intr_raw_overflow[15]	When 15 is written to 16'h4, the corresponding interrupt count is 3 and an overflow error occurs	0
30	R	intr_raw_overflow[14]	When 14 is written to 16'h4, the corresponding interrupt count is 3 and an overflow error occurs	0
29	R	intr_raw_overflow[13]	When 13 is written to 16'h4, the corresponding interrupt count is 3 and an overflow error occurs	0
28	R	intr_raw_overflow[12]	When 12 is written to 16'h4, the corresponding interrupt count is 3 and an overflow error occurs	0
27	R	intr_raw_overflow[11]	When writing 11 to 16'h4, the corresponding interrupt count is 3 and an overflow error occurs	0
26	R	intr_raw_overflow[10]	When 10 is written to 16'h4, the corresponding interrupt count is 3 and an overflow error occurs	0
25	R	intr_raw_overflow[9]	When 9 is written to 16'h4, the corresponding interrupt count is 3 and an overflow error occurs	0

Bits	Access	Name	Description	Reset
24	R	intr_raw_overflow[8]	When writing 8 to 16'h4, the corresponding interrupt count is 3 and an overflow error occurs	0
23	R	intr_raw_overflow[7]	When 7 is written to 16'h4, the corresponding interrupt count is 3 and an overflow error occurs	0
22	R	intr_raw_overflow[6]	When writing 6 to 16'h4, the corresponding interrupt count is 3 and an overflow error occurs	0
21	R	intr_raw_overflow[5]	When writing 5 to 16'h4, the corresponding interrupt count is 3 and an overflow error occurs	0
20	R	intr_raw_overflow[4]	When writing 4 to 16'h4, the corresponding interrupt count is 3 and an overflow error occurs	0
19	R	intr_raw_overflow[3]	When writing 3 to 16'h4, the corresponding interrupt count is 3 and an overflow error occurs	0
18	R	intr_raw_overflow[2]	When writing 2 to 16'h4, the corresponding interrupt count is 3 and an overflow error occurs	0
17	R	intr_raw_overflow[1]	When writing 1 to 16'h4, the corresponding interrupt count is 3 and an overflow error occurs	0
16	R	intr_raw_overflow[0]	When writing 0 to 16'h4, the corresponding interrupt count is 3 and an overflow error occurs	0
15	R	intr_raw_underflow[15]	When 15 is written to 16'h8, the corresponding interrupt count is 0 and a lower overflow error occurs	0
14	R	intr_raw_underflow[14]	When 14 is written to 16'h8, the corresponding interrupt count is 0 and a lower overflow error occurs	0
13	R	intr_raw_underflow[13]	When 13 is written to 16'h8, the corresponding interrupt count is 0 and a lower overflow error occurs	0
12	R	intr_raw_underflow[12]	When 12 is written to 16'h8, the corresponding interrupt count is 0 and a lower overflow error occurs	0
11	R	intr_raw_underflow[11]	When writing 11 to 16'h8, the corresponding interrupt count is 0 and a lower overflow error occurs	0
10	R	intr_raw_underflow[10]	When 10 is written to 16'h8, the corresponding interrupt count is 0 and a lower overflow error occurs	0
9	R	intr_raw_underflow[9]	When writing 9 to 16'h8, the corresponding interrupt count is 0 and a lower overflow error occurs	0
8	R	intr_raw_underflow[8]	When writing 8 to 16'h8, the corresponding interrupt count is 0 and a lower overflow error occurs	0
7	R	intr_raw_underflow[7]	When writing 7 to 16'h8, the corresponding interrupt count is 0 and a lower overflow error occurs	0
6	R	intr_raw_underflow[6]	When writing 6 to 16'h8, the corresponding interrupt count is 0 and a lower overflow error occurs	0
5	R	intr_raw_underflow[5]	When writing 5 to 16'h8, the corresponding interrupt count is 0 and a lower overflow error occurs	0

Bits	Access	Name	Description	Reset
4	R	intr_raw_underflow[4]	When writing 4 to 16'h8, the corresponding interrupt count is 0 and a lower overflow error occurs	0
3	R	intr_raw_underflow[3]	When writing 3 to 16'h8, the corresponding interrupt count is 0 and a lower overflow error occurs	0
2	R	intr_raw_underflow[2]	When writing 2 to 16'h8, the corresponding interrupt count is 0 and a lower overflow error occurs	0
1	R	intr_raw_underflow[1]	When writing 1 to 16'h8, the corresponding interrupt count is 0 and a lower overflow error occurs	0
0	R	intr_raw_underflow[0]	When writing 0 to 16'h8, the corresponding interrupt count is 0 and a lower overflow error occurs	0

DSP2CPU_INT_EN0

cpu1_sys2cpu0_sys interrupt enable register 0.

Offset:0x14 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:16	RW	intr_raw_en	1: enable 0:disable	0
15:1	Reserved	Reserved	Reserved	-
0	RW	intr_en	1:enable 0:disable	0

DSP2CPU_INT_SET0

cpu1_sys2cpu0_sys interrupt setting register 0.

Offset:0x18 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:0	W	intr set	Take the value range [0, 15]. The interrupt count of intr_raw0 ~ intr_raw15 of DSP2CPU_INT_STATUS corresponds to the interrupt status register, and the interrupt count is increased by 1 for each setting.	-

DSP2CPU_INT_CLEAR0

cpu1_sys2cpu0_sys interrupt clear register 0.

Offset:0x1c Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:0	W	intr set	Take the value range [0, 15]. The interrupt count of intr_raw0 ~ intr_raw15 of DSP2CPU_INT_STATUS corresponds to the interrupt count of interrupt status register, and the interrupt count is decremented by 1 for each setting.	-

DSP2CPU_INT_STATUS0

cpu1_sys2cpu0_sys interrupt status register 0.

Offset:0x20 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset

Bits	Access	Name	Description	Reset
31:30	R	intr_raw[15]	When writing 15 to 16'h14, the interrupt count is added 1; when writing 15 to 16'h18, the interrupt count is subtracted 1; when cpu2dsp_intr_soft_rst, the interrupt count is cleared 0	0
29:28	R	intr_raw[14]	When writing 14 to 16'h14, the interrupt count is added 1; when writing 14 to 16'h18, the interrupt count is subtracted 1; when cpu2dsp_intr	0
27:26	R	intr_raw[13]	When writing 13 to 16'h14, the interrupt count is added 1; when writing 13 to 16'h18, the interrupt count is subtracted 1; when cpu2dsp_intr	0
25:24	R	intr_raw[12]	When writing 12 to 16'h14, the interrupt count is added 1; when writing 12 to 16'h18, the interrupt count is subtracted 1; when cpu2dsp_intr	0
23:22	R	intr_raw[11]	When writing 11 to 16'h14, the interrupt count is added 1; when writing 11 to 16'h18, the interrupt count is subtracted 1; when cpu2dsp_intr	0
21:20	R	intr_raw[10]	When writing 10 to 16'h14, the interrupt count is added 1; when writing 10 to 16'h18, the interrupt count is subtracted 1; when cpu2dsp_intr	0
19:18	R	intr_raw[9]	When writing 9 to 16'h14, the interrupt count is added 1; when writing 9 to 16'h18, the interrupt count is subtracted 1; when cpu2dsp_intr	0
17:16	R	intr_raw[8]	When writing 8 to 16'h14, the interrupt count is added 1; when writing 8 to 16'h18, the interrupt count is subtracted 1; when cpu2dsp_intr	0
15:14	R	intr_raw[7]	When writing 7 to 16'h14, the interrupt count is added 1; when writing 7 to 16'h18, the interrupt count is subtracted 1; when cpu2dsp_intr	0
13:12	R	intr_raw[6]	When writing 6 to 16'h14, the interrupt count is added 1; when writing 6 to 16'h18, the interrupt count is subtracted 1; when cpu2dsp_intr	0
11:10	R	intr_raw[5]	When writing 5 to 16'h14, the interrupt count is added 1; when writing 5 to 16'h18, the interrupt count is subtracted 1; when cpu2dsp_intr	0
9:8	R	intr_raw[4]	When writing 4 to 16'h14, the interrupt count is added 1; when writing 4 to 16'h18, the interrupt count is subtracted 1; when cpu2dsp_intr	0
7:6	R	intr_raw[3]	When writing 3 to 16'h14, the interrupt count is added 1; when writing 3 to 16'h18, the interrupt count is subtracted 1; when cpu2dsp_intr	0
5:4	R	intr_raw[2]	When writing 2 to 16'h14, the interrupt count is added 1; when writing 2 to 16'h18, the interrupt count is subtracted 1; when cpu2dsp_intr	0
3:2	R	intr_raw[1]	When writing 1 to 16'h14, the interrupt count is added 1; when writing 1 to 16'h18, the interrupt count is subtracted 1; when cpu2dsp_intr	0
1:0	R	intr_raw[0]	When writing 0 to 16'h14, the interrupt count is added 1; when writing 0 to 16'h18, the interrupt count is subtracted 1; when cpu2dsp_intr	0

DSP2CPU_INT_ERR0

cpu1_sys2cpu0_sys interrupt error register 0.

Offset:0x24 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31	R	intr_raw_overflow[15]	When 15 is written to 16'h14, the corresponding interrupt count is 3 and an overflow error occurs	0

Bits	Access	Name	Description	Reset
30	R	intr_raw_overflow[14]	When writing 14 to 16'h14, the corresponding interrupt count is 3 and an overflow error occurs	0
29	R	intr_raw_overflow[13]	When 13 is written to 16'h14, the corresponding interrupt count is 3 and an overflow error occurs	0
28	R	intr_raw_overflow[12]	When 12 is written to 16'h14, the corresponding interrupt count is 3 and an overflow error occurs	0
27	R	intr_raw_overflow[11]	When writing 11 to 16'h14, the corresponding interrupt count is 3 and an overflow error occurs	0
26	R	intr_raw_overflow[10]	When writing 10 to 16'h14, the corresponding interrupt count is 3 and an overflow error occurs	0
25	R	intr_raw_overflow[9]	When writing 9 to 16'h14, the corresponding interrupt count is 3 and an overflow error occurs	0
24	R	intr_raw_overflow[8]	When writing 8 to 16'h14, the corresponding interrupt count is 3 and an overflow error occurs	0
23	R	intr_raw_overflow[7]	When writing 7 to 16'h14, the corresponding interrupt count is 3 and an overflow error occurs	0
22	R	intr_raw_overflow[6]	When writing 6 to 16'h14, the corresponding interrupt count is 3 and an overflow error occurs	0
21	R	intr_raw_overflow[5]	When writing 5 to 16'h14, the corresponding interrupt count is 3 and an overflow error occurs	0
20	R	intr_raw_overflow[4]	When writing 4 to 16'h14, the corresponding interrupt count is 3 and an overflow error occurs	0
19	R	intr_raw_overflow[3]	When writing 3 to 16'h14, the corresponding interrupt count is 3 and an overflow error occurs	0
18	R	intr_raw_overflow[2]	When writing 2 to 16'h14, the corresponding interrupt count is 3 and an overflow error occurs	0
17	R	intr_raw_overflow[1]	When writing 1 to 16'h14, the corresponding interrupt count is 3 and an overflow error occurs	0
16	R	intr_raw_overflow[0]	When writing 0 to 16'h14, the corresponding interrupt count is 3 and an overflow error occurs	0
15	R	intr_raw_underflow[15]	When 15 is written to 16'h18, the corresponding interrupt count is 0 and a lower overflow error occurs	0
14	R	intr_raw_underflow[14]	When 14 is written to 16'h18, the corresponding interrupt count is 0 and a lower overflow error occurs	0
13	R	intr_raw_underflow[13]	When 13 is written to 16'h18, the corresponding interrupt count is 0 and a lower overflow error occurs	0
12	R	intr_raw_underflow[12]	When 12 is written to 16'h18, the corresponding interrupt count is 0 and a lower overflow error occurs	0
11	R	intr_raw_underflow[11]	When writing 11 to 16'h18, the corresponding interrupt count is 0 and a lower overflow error occurs	0

Bits	Access	Name	Description	Reset
10	R	intr_raw_underflow[10]	When 10 is written to 16'h18, the corresponding interrupt count is 0 and a lower overflow error occurs	0
9	R	intr_raw_underflow[9]	When writing 9 to 16'h18, the corresponding interrupt count is 0 and a lower overflow error occurs	0
8	R	intr_raw_underflow[8]	When writing 8 to 16'h18, the corresponding interrupt count is 0 and a lower overflow error occurs	0
7	R	intr_raw_underflow[7]	When writing 7 to 16'h18, the corresponding interrupt count is 0 and a lower overflow error occurs	0
6	R	intr_raw_underflow[6]	When writing 6 to 16'h18, the corresponding interrupt count is 0 and a lower overflow error occurs	0
5	R	intr_raw_underflow[5]	When writing 5 to 16'h18, the corresponding interrupt count is 0 and a lower overflow error occurs	0
4	R	intr_raw_underflow[4]	When writing 4 to 16'h18, the corresponding interrupt count is 0 and a lower overflow error occurs	0
3	R	intr_raw_underflow[3]	When writing 3 to 16'h18, the corresponding interrupt count is 0 and a lower overflow error occurs	0
2	R	intr_raw_underflow[2]	When writing 2 to 16'h18, the corresponding interrupt count is 0 and a lower overflow error occurs	0
1	R	intr_raw_underflow[1]	When writing 1 to 16'h18, the corresponding interrupt count is 0 and a lower overflow error occurs	0
0	R	intr_raw_underflow[0]	When writing 0 to 16'h18, the corresponding interrupt count is 0 and a lower overflow error occurs	0

CPU2DSP_INT_EN1

cpu0_sys2cpu1_sys interrupt enable register 1.

Offset:0x28 Total Reset Value:0x0000_0000 The detailed description is the same as Group 0 registers.

CPU2DSP_INT_SET1

cpu0_sys2cpu1_sys interrupt setting register 1.

Offset:0x2c Total Reset Value:0x0000_0000 The detailed description is the same as Group 0 registers.

CPU2DSP_INT_CLEAR1

cpu0_sys2cpu1_sys interrupt clear register 1.

Offset:0x30 Total Reset Value:0x0000_0000 The detailed description is the same as Group 0 registers.

CPU2DSP_INT_STATUS1

cpu0_sys2cpu1_sys interrupt status register 1.

Offset:0x34 Total Reset Value:0x0000_0000 The detailed description is the same as Group 0 registers.

CPU2DSP_INT_ERR1

cpu0_sys2cpu1_sys interrupt error register 1.

Offset:0x38 Total Reset Value:0x0000_0000 The detailed description is the same as Group 0 registers.

DSP2CPU_INT_EN1

cpu1_sys2cpu0_sys interrupt enable register 1.

Offset:0x3c Total Reset Value:0x0000_0000 The detailed description is the same as Group 0 registers.

DSP2CPU_INT_SET1

cpu1_sys2cpu0_sys interrupt setting register 1.

Offset:0x40 Total Reset Value:0x0000_0000 The detailed description is the same as Group 0 registers.

DSP2CPU_INT_CLEAR1

cpu1_sys2cpu0_sys interrupt clear register 1.

Offset:0x44 Total Reset Value:0x0000_0000 The detailed description is the same as Group 0 registers.

DSP2CPU_INT_STATUS1

cpu1_sys2cpu0_sys interrupt status register 1.

Offset:0x48 Total Reset Value:0x0000_0000 The detailed description is the same as Group 0 registers.

DSP2CPU_INT_ERR1

cpu1_sys2cpu0_sys interrupt error register 1.

Offset:0x4c Total Reset Value:0x0000_0000 The detailed description is the same as Group 0 registers.

CPU2DSP_INT_EN2

cpu0_sys2cpu1_sys interrupt enable register 2.

Offset:0x50 Total Reset Value:0x0000_0000 The detailed description is the same as Group 0 registers.

CPU2DSP_INT_SET2

cpu0_sys2cpu1_sys interrupt setting register 2.

Offset:0x54 Total Reset Value:0x0000_0000 The detailed description is the same as Group 0 registers.

CPU2DSP_INT_CLEAR2

cpu0_sys2cpu1_sys interrupt clear register 2.

Offset:0x58 Total Reset Value:0x0000_0000 The detailed description is the same as Group 0 registers.

CPU2DSP_INT_STATUS2

cpu0_sys2cpu1_sys interrupt status register 2.

Offset:0x5c Total Reset Value:0x0000_0000 The detailed description is the same as Group 0 registers.

CPU2DSP_INT_ERR2

cpu0_sys2cpu1_sys interrupt error register 2.

Offset:0x60 Total Reset Value:0x0000_0000 The detailed description is the same as Group 0 registers.

DSP2CPU_INT_EN2

cpu1_sys2cpu0_sys interrupt enable register 2.

Offset:0x64 Total Reset Value:0x0000_0000 The detailed description is the same as Group 0 registers.

DSP2CPU_INT_SET2

cpu1_sys2cpu0_sys interrupt setting register 2.

Offset:0x68 Total Reset Value:0x0000_0000 The detailed description is the same as Group 0 registers.

DSP2CPU_INT_CLEAR2

cpu1_sys2cpu0_sys interrupt clear register 2.

Offset:0x6c Total Reset Value:0x0000_0000 The detailed description is the same as Group 0 registers.

DSP2CPU_INT_STATUS2

cpu1_sys2cpu0_sys interrupt status register 2.

Offset:0x70 Total Reset Value:0x0000_0000 The detailed description is the same as Group 0 registers.

DSP2CPU_INT_ERR2

cpu1_sys2cpu0_sys interrupt error register 2.

Offset:0x74 Total Reset Value:0x0000_0000 The detailed description is the same as Group 0 registers.

CPU2DSP_INT_EN3

cpu0_sys2cpu1_sys interrupt enable register 3.

Offset:0x78 Total Reset Value:0x0000_0000 The detailed description is the same as Group 0 registers.

CPU2DSP_INT_SET3

cpu0_sys2cpu1_sys interrupt setting register 3.

Offset:0x7c Total Reset Value:0x0000_0000 The detailed description is the same as Group 0 registers.

CPU2DSP_INT_CLEAR3

cpu0_sys2cpu1_sys interrupt clear register 3.

Offset:0x80 Total Reset Value:0x0000_0000 The detailed description is the same as Group 0 registers.

CPU2DSP_INT_STATUS3

cpu0_sys2cpu1_sys interrupt status register 3.

Offset:0x84 Total Reset Value:0x0000_0000 The detailed description is the same as Group 0 registers.

CPU2DSP_INT_ERR3

cpu0_sys2cpu1_sys interrupt error register 3.

Offset:0x88 Total Reset Value:0x0000_0000 The detailed description is the same as Group 0 registers.

DSP2CPU_INT_EN3

cpu1_sys2cpu0_sys interrupt enable register 3.

Offset:0x8c Total Reset Value:0x0000_0000 The detailed description is the same as Group 0 registers.

DSP2CPU_INT_SET3

cpu1_sys2cpu0_sys interrupt setting register 3.

Offset:0x90 Total Reset Value:0x0000_0000 The detailed description is the same as Group 0 registers.

DSP2CPU_INT_CLEAR3

cpu1_sys2cpu0_sys interrupt clear register 3.

Offset:0x94 Total Reset Value:0x0000_0000 The detailed description is the same as Group 0 registers.

DSP2CPU_INT_STATUS3

cpu1_sys2cpu0_sys interrupt status register 3.

Offset:0x98 Total Reset Value:0x0000_0000 The detailed description is the same as Group 0 registers.

DSP2CPU_INT_ERR3

cpu1_sys2cpu0_sys interrupt error register 3.

Offset:0x9c Total Reset Value:0x0000_0000 The detailed description is the same as Group 0 registers.

MAILBOX0

MAILBOX interlock register 0.

Offset:0xa0 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:1	-	-	Reserved	-
0	RW	Lock	1: locked 0: not locked	0

MAILBOX1

MAILBOX interlock register 1.

Offset:0xa4 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:1	-	-	Reserved	-
0	RW	Lock	1: locked 0: not locked	0

MAILBOX2
MAILBOX interlock register 2.
Offset:0xa8 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:1	-	-	Reserved	-
0	RW	Lock	1: locked 0: not locked	0

MAILBOX3
MAILBOX interlock register 3.
Offset:0xac Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:1	-	-	Reserved	-
0	RW	Lock	1: locked 0: not locked	0

MAILBOX4~MAILBOX126
MAILBOX interlock register 4~126.
Offset:0xb0~0x298 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:1	-	-	Reserved	-
0	RW	Lock	1: locked 0: not locked	0

MAILBOX127
MAILBOX interlock register 127.
Offset:0x29c Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:1	-	-	Reserved	-
0	RW	Lock	1: locked 0: not locked	0

UART_LP_REQ_SCLK
UART SLCK low-power status register.
Offset:0x2a0 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset

Bits	Access	Name	Description	Reset
31:5	-	-	Reserved	-
4	R	UART4 sclk low power req	The sclk domain clock gate signal indicates that the UART is inactive, so clocks may be gated to put the device in a low-power(lp) mode. exists: CLK_GATE_EN == 1 && CLOCK_MODE == 2 1: active 0: non-active	0
3	R	UART3 sclk low power req	The sclk domain clock gate signal indicates that the UART is inactive, so clocks may be gated to put the device in a low-power(lp) mode. exists: CLK_GATE_EN == 1 && CLOCK_MODE == 2 1: active 0: non-active	0
2	R	UART 2 sclk low power req	The sclk domain clock gate signal indicates that the UART is inactive, so clocks may be gated to put the device in a low-power(lp) mode. exists: CLK_GATE_EN == 1 && CLOCK_MODE == 2 1: active 0: non-active	0
1	R	UART 1 sclk low power req	The sclk domain clock gate signal indicates that the UART is inactive, so clocks may be gated to put the device in a low-power(lp) mode. exists: CLK_GATE_EN == 1 && CLOCK_MODE == 2 1: active 0: non-active	0
0	R	UART 0 sclk low power req	The sclk domain clock gate signal indicates that the UART is inactive, so clocks may be gated to put the device in a low-power(lp) mode. exists: CLK_GATE_EN == 1 && CLOCK_MODE == 2 1: active 0: non-active	0

UART_LP_REQ_PCLK

UART PLCK low-power status register.

Offset:0x2a4 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:5	-	-	Reserved	-
4	R	UART 4 pclk low power req	The pclk domain clock gate signal indicates that the UART is inactive, so clocks may be gated to put the device in a low-power(lp) mode. exists: CLK_GATE_EN == 1 1: active 0: non-active	0
3	R	UART 3 pclk low power req	The pclk domain clock gate signal indicates that the UART is inactive, so clocks may be gated to put the device in a low-power(lp) mode. exists: CLK_GATE_EN == 1 1: active 0: non-active	0

Bits	Access	Name	Description	Reset
2	R	UART 2 pclk low power req	The pclk domain clock gate signal indicates that the UART is inactive, so clocks may be gated to put the device in a low-power(lp) mode. exists: CLK_GATE_EN == 1 1: active 0: non-active	0
1	R	UART 1 pclk low power req	The pclk domain clock gate signal indicates that the UART is inactive, so clocks may be gated to put the device in a low-power(lp) mode. exists: CLK_GATE_EN == 1 1: active 0: non-active	0
0	R	UART 0 pclk low power req	The pclk domain clock gate signal indicates that the UART is inactive, so clocks may be gated to put the device in a low-power(lp) mode. exists: CLK_GATE_EN == 1 1: active 0: non-active	0

QOS_CTRL0

QoS control register 0.

Offset:0x2a8 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:28	RW	Cpu1 axi read QoS	QoS level, range from 0 to 7 0: minimum QoS 7: maximum QoS	0
27:24	RW	Cpu1axi write QoS	QoS level, range from 0 to 7 0: minimum QoS 7: maximum QoS	0
23:20	RW	Cpu0 axi read QoS	QoS level, range from 0 to 7 0: minimum QoS 7: maximum QoS	0
19:16	RW	cpu0 axi write QoS	QoS level, range from 0 to 7 0: minimum QoS 7: maximum QoS	0
15:12	RW	Ai axi1 read QoS	QoS level, range from 0 to 7 0: minimum QoS 7: maximum QoS	0
11:8	RW	Ai axi1 write QoS	QoS level, range from 0 to 7 0: minimum QoS 7: maximum QoS	0
7:4	RW	Ai axi0 read QoS	QoS level, range from 0 to 7 0: minimum QoS 7: maximum QoS	0

Bits	Access	Name	Description	Reset
3:0	RW	Ai axi0 write QoS	QoS level, range from 0 to 7 0: minimum QoS 7: maximum QoS	0

QOS_CTRL1

QOS control register 1.

Offset:0x2ac Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:28	RW	Hi sd read QoS	QoS level, range from 0 to 7 0: minimum QoS 7: maximum QoS	0
27:24	RW	Hi sd write QoS	QoS level, range from 0 to 7 0: minimum QoS 7: maximum QoS	0
23:20	RW	g2p5d read QoS	QoS level, range from 0 to 7 0: minimum QoS 7: maximum QoS	0
19:16	RW	g2p5d write QoS	QoS level, range from 0 to 7 0: minimum QoS 7: maximum QoS	0
15:12	RW	dpu read QoS	QoS level, range from 0 to 7 0: minimum QoS 7: maximum QoS	0
11:8	RW	dpu write QoS	QoS level, range from 0 to 7 0: minimum QoS 7: maximum QoS	0
7:4	RW	display read QoS	QoS level, range from 0 to 7 0: minimum QoS 7: maximum QoS	0
3:0	RW	display write QoS	QoS level, range from 0 to 7 0: minimum QoS 7: maximum QoS	0

QOS_CTRL2

QOS control register 2.

Offset:0x2b0 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:28	RW	Isp_3dnr read QoS	QoS level, range from 0 to 7 0: minimum QoS 7: maximum QoS	0

Bits	Access	Name	Description	Reset
27:24	RW	isp_3dnr write QoS	QoS level, range from 0 to 7 0: minimum QoS 7: maximum QoS	0
23:20	-	-	-	-
19:16	RW	Hi usb write QoS	QoS level, range from 0 to 7 0: minimum QoS 7: maximum QoS	0
15:12	RW	Hi Ssi1_2 read QoS	QoS level, range from 0 to 7 0: minimum QoS 7: maximum QoS	0
11:8	RW	Hi Ssi1_2 write QoS	QoS level, range from 0 to 7 0: minimum QoS 7: maximum QoS	0
7:4	RW	Hi ssi0 read QoS	QoS level, range from 0 to 7 0: minimum QoS 7: maximum QoS	0
3:0	RW	Hi ssi0 write QoS	QoS level, range from 0 to 7 0: minimum QoS 7: maximum QoS	0

QOS_CTRL3

QOS control register 3.

Offset:0x2b4 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:28	RW	Isp_scal read QoS	QoS level, range from 0 to 7 0: minimum QoS 7: maximum QoS	0
27:24	RW	Isp_scal write QoS	QoS level, range from 0 to 7 0: minimum QoS 7: maximum QoS	0
23:20	RW	Isp_mp_mcm read QoS	QoS level, range from 0 to 7 0: minimum QoS 7: maximum QoS	0
19:16	RW	Isp_mp_mcm write QoS	QoS level, range from 0 to 7 0: minimum QoS 7: maximum QoS	0
15:12	RW	Isp_hdr read QoS	QoS level, range from 0 to 7 0: minimum QoS 7: maximum QoS	0
11:8	RW	Isp_hdr write QoS	QoS level, range from 0 to 7 0: minimum QoS 7: maximum QoS	0

Bits	Access	Name	Description	Reset
7:4	RW	Isp_dwe read QoS	QoS level, range from 0 to 7 0: minimum QoS 7: maximum QoS	0
3:0	RW	Isp_dwe write QoS	QoS level, range from 0 to 7 0: minimum QoS 7: maximum QoS	0

QOS_CTRL4

QOS control register 4.

Offset:0x2b8 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:28	RW	vpu read QoS	QoS level, range from 0 to 7 0: minimum QoS 7: maximum QoS	0
27:24	RW	vpu write QoS	QoS level, range from 0 to 7 0: minimum QoS 7: maximum QoS	0
23:20	RW	stor read QoS	QoS level, range from 0 to 7 0: minimum QoS 7: maximum QoS	0
19:16	RW	stor write QoS	QoS level, range from 0 to 7 0: minimum QoS 7: maximum QoS	0
15:12	-	-	-	-
11:8	-	-	-	-
7:4	RW	sec read QoS	QoS level, range from 0 to 7 0: minimum QoS 7: maximum QoS	0
3:0	RW	sec write QoS	QoS level, range from 0 to 7 0: minimum QoS 7: maximum QoS	0

NOC_WR_BUFFERABLE_CTRL

NOC Bufferable Control Register.

Offset:0x2bc Total Reset Value:0xffff_fff

Bits	Access	Name	Description	Reset
31:20	-		Reserved	-
19	RW	Master19_bufferable	1: Bufferable 0: Non-bufferable	1
18	RW	Master18_bufferable	1: Bufferable 0: Non-bufferable	1

Bits	Access	Name	Description	Reset
17	RW	Master17_bufferable	1: Bufferable 0: Non-bufferable	1
16	RW	Master16_bufferable	1: Bufferable 0: Non-bufferable	1
15	RW	Master15_bufferable	1: Bufferable 0: Non-bufferable	1
14	RW	Master14_bufferable	1: Bufferable 0: Non-bufferable	1
13	RW	Master13_bufferable	1: Bufferable 0: Non-bufferable	1
12	RW	Master12_bufferable	1: Bufferable 0: Non-bufferable	1
11	RW	Master11_bufferable	1: Bufferable 0: Non-bufferable	1
10	RW	Master10_bufferable	1: Bufferable 0: Non-bufferable	1
9	RW	Master9_bufferable	1: Bufferable 0: Non-bufferable	1
8	RW	Master8_bufferable	1: Bufferable 0: Non-bufferable	1
7	RW	Master7_bufferable	1: Bufferable 0: Non-bufferable	1
6	RW	Master6_bufferable	1: Bufferable 0: Non-bufferable	1
5	RW	Master5_bufferable	1: Bufferable 0: Non-bufferable	1
4	RW	Master4_bufferable	1: Bufferable 0: Non-bufferable	1
3	RW	Master3_bufferable	1: Bufferable 0: Non-bufferable	1
2	RW	Master2_bufferable	1: Bufferable 0: Non-bufferable	1
1	RW	Master1_bufferable	1: Bufferable 0: Non-bufferable	1
0	RW	Master0_bufferable	1: Bufferable 0: Non-bufferable	1

DDR_REG0

DDR_URGENT_MASK.

Offset:0x2c0 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:30	-		Reserve	-
29	RW	arurgentb_4	1: urgent transaction (use with caution, use with port_urgent_en enabled) 0: not urgent transaction.	0
28	RW	awurgent_4	1: urgent transaction (use with caution, use with port_urgent_en enabled) 0: not urgent transaction.	0
27	-		Reserve	-
26	RW	arurgentb_3	1: urgent transaction (use with caution, use with port_urgent_en enabled) 0: not urgent transaction.	0
25	RW	awurgent_3	1: urgent transaction (use with caution, use with port_urgent_en enabled) 0: not urgent transaction.	0
24	-		Reserve	-
23	RW	arurgentb_2	1: urgent transaction (use with caution, use with port_urgent_en enabled) 0: not urgent transaction.	0
22	RW	awurgent_2	1: urgent transaction (use with caution, use with port_urgent_en enabled) 0: not urgent transaction.	0
21	-		Reserve	-
20	RW	arurgentb_1	1: urgent transaction (use with caution, use with port_urgent_en enabled) 0: not urgent transaction.	0
19	RW	awurgent_1	1: urgent transaction (use with caution, use with port_urgent_en enabled) 0: not urgent transaction.	0
18	-		Reserve	-
17	RW	arurgentb_0	1: urgent transaction (use with caution, use with port_urgent_en enabled) 0: not urgent transaction.	0
16	RW	awurgent_0	1: urgent transaction (use with caution, use with port_urgent_en enabled) 0: not urgent transaction.	0
15		reserve		0
14:5	RW	pa_rmask	1: mask corresponding port (read) 0: not mask	0
4:0	RW	pa_wmask	1: mask corresponding port (write) 0: not mask	0

DDR_BypassOutEnDAT.

Bits	Access	Name	Description	Reset
31:0	R/W	BypassOutEnDAT[31:0]	Flyover signal (corresponding control BP_D[31:0])	0

DDR_BypassInDataDat.

Bits	Access	Name	Description	Reset
31:0	R	BypassInDataDAT[31:0]	Flyover signal (corresponding BP_D[31:0] for sample)	0

DDR_BypassDat_H.

Bits	Access	Name	Description	Reset
31:16	R	BypassInDataDAT[47:32]	Flyover signal (corresponding BP_D[47:32] for sample)	0
15:0	R/W	BypassOutEnDAT[47:32]	Flyover signal (corresponding control BP_D[47:32])	0

DDR_BypassOutEnAC.

Bits	Access	Name	Description	Reset
31:0	R/W	BypassOutEnAC[31:0]	Flyover signal (corresponding control BP_A[31:0])	0

DDR_BypassInDataAC.

DDR_URGENT_MASK.

Bits	Access	Name	Description	Reset
31:24	R	BypassInDataAC[39:32]	Flyover signal (corresponding BP_A[39:32] for sample)	0
23:16	R/W	BypassOutEnAC[39:32]	Flyover signal (corresponding control BP_A[39:32])	0
15:10			Reserved	
9	R/W	BypassOutDataDAT	input data for Bypass mode (one signal controls all Dat)	0
8	R/W	BypassOutDataAC	input data for Bypass mode (one signal controls all ac)	0
7:6	R	BypassInDataMASTER	output data for Bypass mode	0
5	R/W	BypassOutDataMASTER	input data for Bypass mode	0

Bits	Access	Name	Description	Reset
4:3	R/W	BypassOutEnMASTER	input enable for Bypass mode	0
2	R/W	BypassModeEnMASTER	Bypass mode enable for Master	0
1	R/W	BypassModeEnDAT	Bypass mode enable for DAT	0
0	R/W	BypassModeEnAC	Bypass mode enable for AC	0

DDRC_Refresh

DDRC 0 Self Refresh status and type.

Offset:0x2f0 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:2	-	-	-	-
1:00	R	DDRC_SELFREF_TYPE[31:0]	Flags if self-refresh (except LPDDR4), or SR-Powerdown (LPDDR4) is entered, and if it is under automatic self-refresh control only or not. Values: 0x0 (NoSRorSRPD): SDRAM is not in self-refresh (except LPDDR4) or SR-Powerdown (LPDDR4). If retry is enabled by CRCPARCTL1.crc_parity_retry_enable, this also indicates that the SRE command is still in parity error window or retry is in-progress 0x1 (SRPHY): SDRAM is in self-refresh, which is caused by PHY Master Request 0x2 (SRorSRPDnoAUTO): SDRAM is in self-refresh (except LPDDR4) or SR-Powerdown (LPDDR4), which It was not caused solely under automatic self-refresh control. It could have been caused by Hardware Low Power Interface and/or Software (PWRCTL.selfref_sw). If retry is enabled, this ensures that the SRE command is executed correctly without parity error 0x3 (SRorSRPDnDAUTO): SDRAM is in self-refresh (except LPDDR4) or SR-Powerdown (LPDDR4), which was If retry is enabled, This ensures that the SRE command is executed correctly without parity error	0

MIPI_TX_BISTON

MIPI Tx Dphy Configuration Register.

Offset:0x2dc Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:1	-	-	-	-
0	RW	MIPI_TX_BISTON	Starts the pattern generator. Active State:High	0

SSI_SLEEP

SSI Low Power Register.

Offset:0x2e0 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:3	-	-	Reserved	-
2	R	SSI2_SLEEP	0: ssi0 is working, cannot close ssi_clk 1: The system controller can turn off ssi_clk	0
1	R	SSI1_SLEEP	0: ssi0 is working, cannot close ssi_clk 1: The system controller can turn off ssi_clk	0
0	R	SSI0_SLEEP	0: ssi0 is working, cannot close ssi_clk 1: The system controller can turn off ssi_clk	0

PVDET_EN

PVDET enable register.

Offset:0x2e4 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:6	-	-	Reserved	-
5	RW	PVDET_ENB5	1: no active 0: pvdet output enable, active	0
4	RW	PVDET_ENB4	1: no active 0: pvdet output enable, active	0
3	RW	PVDET_ENB3	1: no active 0: pvdet output enable, active	0
2	RW	PVDET_ENB2	1: no active 0: pvdet output enable, active	0
1	RW	PVDET_ENB1	1: no active 0: pvdet output enable, active	0
0	RW	PVDET_ENB0	1: no active 0: pvdet output enable, active	0

PVDET_OUT_DET

PVDET output detection register.

Offset:0x2e8 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:6	-	-	Reserved	-
5	R	PVDET_OUT5	1: 1.8v 0: 3.3v	0
4	R	PVDET_OUT4	1: 1.8v 0: 3.3v	0
3	R	PVDET_OUT3	1: 1.8v 0: 3.3v	0
2	R	PVDET_OUT2	1: 1.8v 0: 3.3v	0
1	R	PVDET_OUT1	1: 1.8v 0: 3.3v	0
0	R	PVDET_OUT0	1: 1.8v 0: 3.3v	0

OTP_HS_MSC_SEL

OTP_HS_MSC_SEL register.

Offset:0x2ec Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:2	-	-	Reserved	-
1	RW	SW_OTP_SEL	0: SW 1: OTP	0
0	RW	SW_CTRL_HS_MSC_SEL	0: Software controlled voltage switching 1: Software controlled voltage switching	0

CODEC_AD2DA_LOOP

CODEC register.

Offset:0x2f4 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:1	-	-	Reserved	-
0	RW	CODEC_AD2DA_LOOP	0: not loop 1: ADC to DAC loop active	0

TIMER_EN

TIMER register.

Offset:0x2f8 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:6	-	-	Reserved	-
5:0	RW	Timer_en		0

2.10 Temperature Sensor

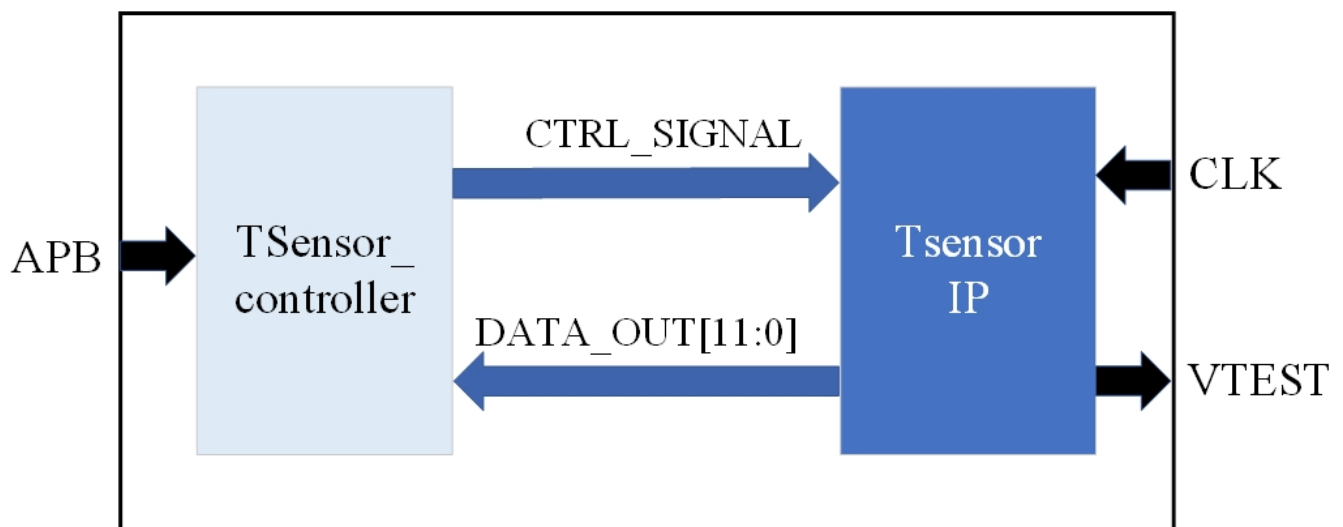
2.10.1 Overview

The TSENSOR is a high precision and low power temperature sensor that embedded in a temperature sensor monitor and a 12-bit resolution analog to digital converter. The TSENSOR has the resolution lower than 0.05°C and 100uA power consumption both on single output mode and continuous mode. The accuracy is within $\pm 3^{\circ}\text{C}$ worst without any trimming.

The TSENSOR is suitable for applications that the working frequency is adjusted according to the temperature value monitored in real time by the TSENSOR. The input clock of TSENSOR is 100KHz.

2.10.2 Block Diagram

Figure 2-10-1 TSENSOR Block Diagram The block diagram of TSENSOR is shown in Figure 2-10-1. It contains a TSensor IP and a TSensor controller.



2.10.3 FEATURE

- $\pm 3^{\circ}\text{C}$ Accuracy Without Trimming
- Supply voltages: VDD 0.72-0.88V, VDDA 1.62-1.98V
- Temperature Measurement Range: $-40 \sim 125^{\circ}\text{C}$

2.10.4 Function Description

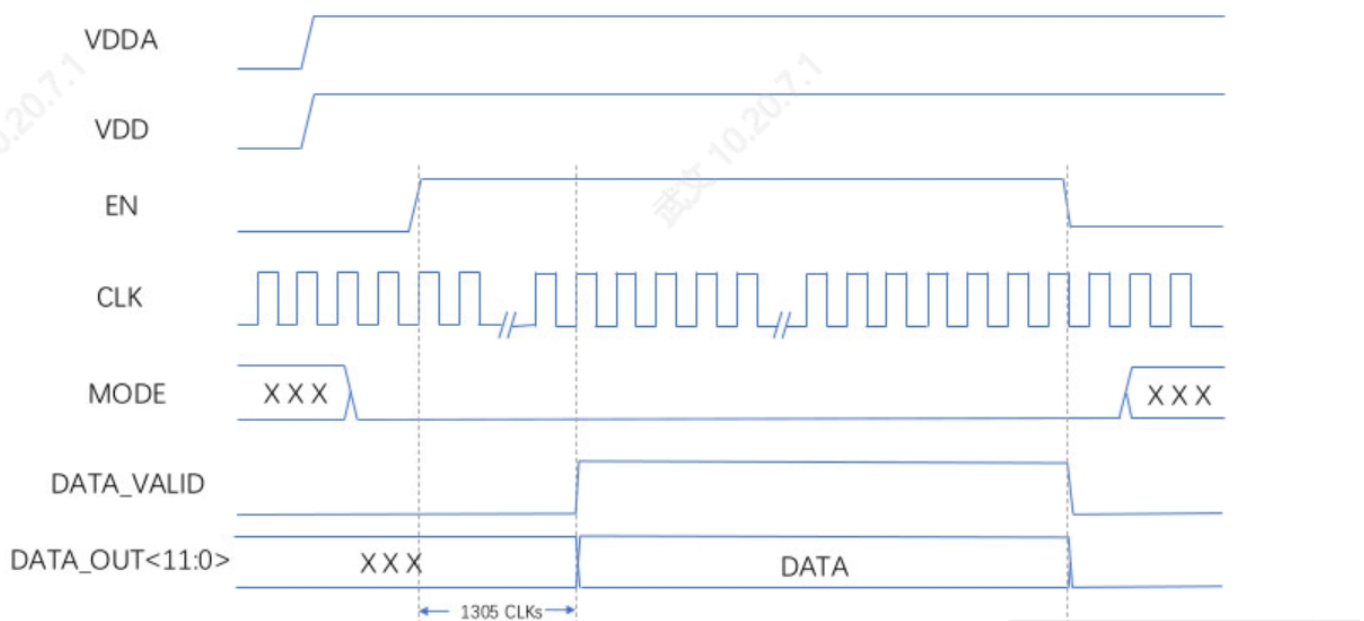
The TSensor IP is configured by The TSensor controller, which is interacted with the APB, and returns the temperature value monitored to The TSensor controller.

There are three output modes: Single output mode, Continuous output mode and Standby mode.

2.10.4.1 Single output mode

Figure 2-10-2 Single Output Mode Sequence

Sequence--Single output mode

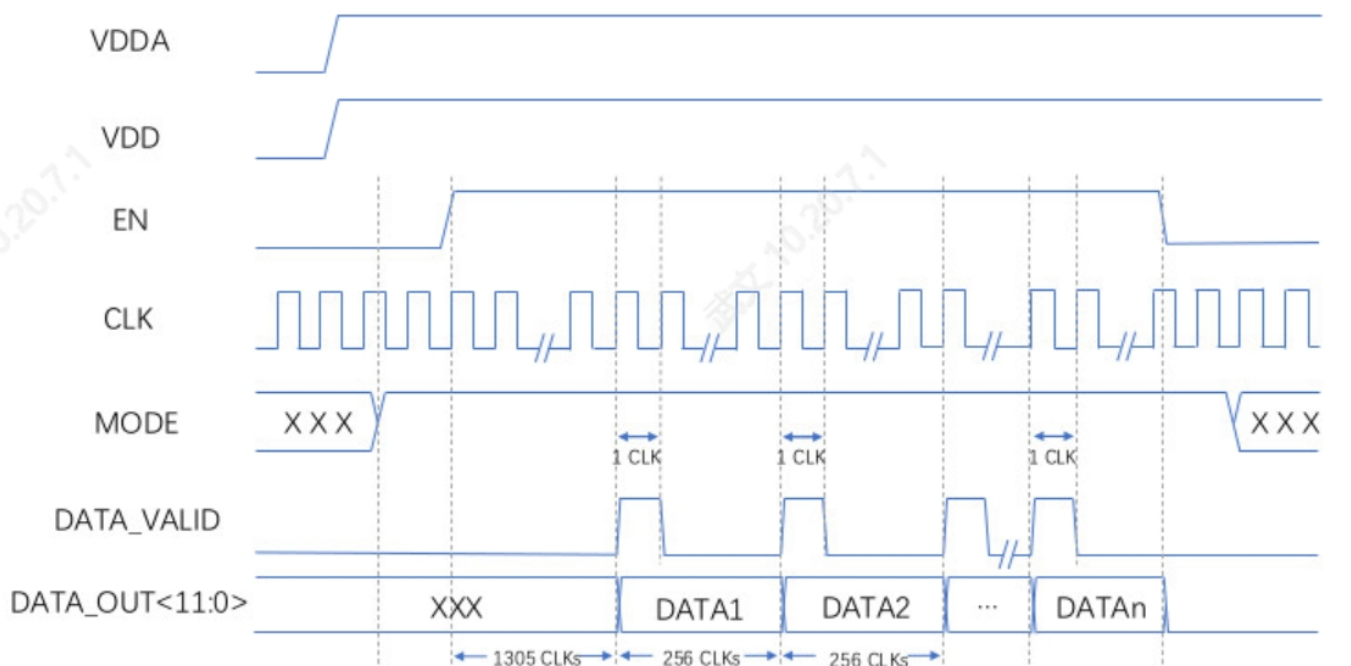


The TSensor IP will be started the temperature monitor after EN signal from 1'b0 to 1'b1. Before EN signal being from 1'b0 to 1'b1, MODE must be 1'b0. After 1305 CLKs, the DATA_VALID signal will go to high level and DATA_OUT data is ready. The Single output is done when EN signal is from 1'b1 to 1'b0 in the meantime DATA_VALID signal is going to low level.

2.10.4.2 Continuous output mode

Figure 2-10-3 Continuous Output Mode Sequence

Sequence--Continuous output mode



The TSensor IP will be started the temperature monitor after EN signal from 1'b0 to 1'b1. Before EN signal being from 1'b0 to 1'b1, MODE must be 1'b1. After 1305 CLKs, the DATA_VALID signal will go to high level for a CLK, and

DATA_OUT data is ready firstly. The DATA_OUT data will not continuously be updated every 256 clock cycles until EN signal from 1'b1 to 1'b0. The Continuous Output is done when EN signal is from 1'b1 to 1'b0.

2.10.4.3 Standby mode

In the Standby mode, the EN signal and TEST_EN signal is 1'b0, and the MODE signal don't care.

2.10.5 Register Description

2.10.5.1 Register Summary

Table 2-1 describes TSENSOR registers.

Table 2-1 Summary of TSENSOR registers (base address: 0x9110_7000)

Offset Address	Register	DESCRIPTIONS
0x000	REG_TSenW	TSensor IP Configuration Register.
0x004	REG_TSenR	Read retured value of TSensor IP Register.

2.10.5.2 Detail Register Description

REG_TSenW

Offset Address: 0x000 Total Reset Value: 0x0000_0020

Bits	Access	Name	Description	Reset
[31:7]	-	reserved	Reserved	-
[6]	RW	ts_test_en	Internal Reference Voltage test enable, active high.	0x0
[5:2]	RW	ts_trim	TSensor Reference Voltage calibration.	0x8
[1]	RW	ts_conv_mode	TSensor Output mode selection. 0: Single output mode 1: Continuous output mode	0x0
[0]	RW	ts_en	TSensor enable, active high.	0x0

REG_TSenR

Offset Address: 0x004 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:13]	-	reserved	Reserved	-
[12]	RC	ts_dout_valid	ts_dout is valid when high.	0x0
[11:0]	RO	ts_dout	TSensor data out.	0x000

2.10.6 Interface Descriptions

IO pads of TSENSOR are listed in the following table.

Table 2-2 TSENSOR interface description

Module Pin	Direction	Pad Name

Module Pin	Direction	Pad Name
VTEST	O	ANA_ts0_VTEST

2.10.7 Application Notes

The sections describe how to carry out temperature conversions using the TSENSOR.

Steps required to carry out a temperature measurement:

Step1. Initialise the IP

Step2. Configure the IP - Register [REG_TSenW](#) bit[1] configured to 0 represents the Single output mode, and [REG_TSenW](#) bit[1] configured to 1 represents the Continuous output mode. Register [REG_TSenW](#) bit[0] configured to 0 and register [REG_TSenW](#) bit[6] configured to 0 represents the Standby mode.

Step3. Start conversion - Register [REG_TSenW](#) bit[0] configured to 1 starts conversion.

Step4. Read Register - The [REG_TSenR](#) bit[11:0] data is valid when [REG_TSenR](#) bit[12] is high. Meanwhile clear the [REG_TSenR](#) bit[12] status. Then the [REG_TSenR](#) bit[11:0] value is converted to temperature value according to the temperature calculation formula below.

3 Processor Subsystem

The Processor Subsystem contains two CPUs, namely small core(cpu0_subsystem) and big core(cpu1_subsystem). Detailed descriptions of cpu0_subsystem and cpu1_subsystem will be introduced in the following chapters.

3.1 cpu0 subsystem

3.1.1 Overview

cpu0_subsystem is used to SoC system services, run LINUX OS.

3.1.2 Function Description

3.1.2.1 Logical Block Diagram

[Figure 3-1-1](#) Logical block diagram of the cpu0 subsystem

3.1.2.2 Features

The cpu0_subsystem uses the T-head C908 processor, which has the following features: 1.Supports maximum 800 MHz working frequency. 2.Cpu contains floating-point units (FPUs). 3.L1 cache contains 32 KB instruction cache (I-cache) and 32 KB data cache (D-cache). 4.Integrates with 128 KB L2 cache. 5.Supports memory management unit (MMU). 6.Integrates with the interrupt controller (PLIC), which can process 208 interrupt external sources. 7.Supports the Joint Test Action Group (JTAG) debugging interface. 8.Supports WFI. 9.Supports T-head TEE(T-head Security extension).

3.1.3 Register Summary

3.1.3.1 System Control Register

Table 3-1 Summary of RISC-V Standard Machine Mode Control Registers

Offset Address	Register	Description
0xF11	MVENDORID	Supplier Number Register
0xF12	MARCHID	Architecture Number Register
0xF13	MIMPID	Machine mode hardware implementation number register
0xF14	MHARTID	Machine mode logical core number register
0xF15	MCONFIGPTR	Machine mode configuration data structure pointer
0x300	MSTATUS	Machine Mode Processor Status Register
0x301	MISA	Machine Mode Processor Instruction Set Feature Register
0x302	MEDELEG	Machine mode exception degradation control register
0x303	MIDELEG	Machine mode interrupt degradation control register
0x304	MIE	Machine mode interrupt enable control register
0x305	MTVEC	Machine mode vector base address register
0x306	MCOUNTEREN	Machine mode counter authorization control register
0x340	MSCRATCH	Machine mode exception temporary data backup register
0x341	MEPC	Machine mode exception retention program counter
0x342	MCAUSE	Machine mode abnormal event cause register
0x343	MTVAL	Machine mode exception event vector register
0x344	MIP	Machine mode interrupt wait status register
0x30A	MENVCFG	Machine mode environment configuration register
0x3A0	PMPCFG0	Physical Memory Protection Configuration Register 0
0x3A2	PMPCFG2	Physical Memory Protection Configuration Register 2
.....	...	
0x3AE	PMPCFG14	Physical Memory Protection Configuration Register 14
0x3B0	PMPADDR0	Physical Memory Protection Base Address Register 0
.....	...	
0x3EF	PMPADDR63	Physical Memory Protection Base Address Register 63
0xB00	MCYCLE	Machine mode cycle counter
0xB02	MINSTRET	Machine mode retirement instruction counter
0xB03	MHPMCOUNTER3	Machine mode counter 3
.....	...	
0xB1F	MHPMCOUNTER31	Machine mode counter 15
0x320	MCOUNTINHIBIT	Machine mode count disable register
0x323	MHPMEVENT3	Machine mode event selection register 3

Offset Address	Register	Description
.....	...	
0x33F	MHPMEVENT31	Machine mode event selection register 31
0x7A0	TSELECT	Debug/trace trigger selection register
0x7A1	TDATA1	Debug/Trace Trigger Data Register 1
0x7A2	TDATA2	Debug/Trace Trigger Data Register 2
0x7A3	TDATA3	Debug/Trace Trigger Data Register 3
0x7A4	TINFO	Debug/Trace Trigger Information Register
0x7A5	TCONTROL	Debug/trace trigger control register
0x7A8	MCONTEXT	Machine mode content register
0x7B0	DCSR	Debug mode control and status registers
0x7B1	DPC	Debug mode program counter
0x7B2	DSCRATCH0	Debug mode temporary data backup register 0
0x7B3	DSCRATCH1	Debug mode temporary data backup register 1

Table 3-2 Summary of RISC-V Standard Super User Mode Control Registers

Offset Address	Register	Description
0x100	SSTATUS	Super User Mode Processor Status Register
0x104	SIE	Super User Mode Interrupt Enable Control Register
0x105	STVEC	Super user mode vector base address register
0x106	SCOUNTEREN	Super User Mode Counter Enable Control Register
0x10A	SENVCFG	Super User Mode Environment Configuration Register
0x140	SSCRATCH	Super user mode exception temporary data backup register
0x141	SEPC	Superuser mode exception retention program counter
0x142	SCAUSE	Super User Mode Exception Event Cause Register
0x143	STVAL	Superuser mode exception event vector register
0x144	SIP	Super User Mode Interrupt Waiting Status Register
0x14D	STIMECMP	Super User Mode Timer Interrupt Compare Value Register
0x180	SATP	Super User Virtual Address Translation and Protection Register
0x5A8	SCONTEXT	Super User Mode Content Register

Table 3-3 Summary of RISC-V Standard User Mode Control Registers

Offset Address	Register	Description

Offset Address	Register	Description
0x001	FFLAGS	Floating point exception accumulation status register
0x002	FRM	Floating-point dynamic rounding mode control register
0x003	FCSR	Floating point control status register
0xC00	CYCLE	User mode cycle counter
0xC01	TIME	User mode time counter
0xC02	INSTRET	User mode retirement command counter
0xC03	HPMCOUNTER3	User mode counter 3
...	...	
0xC1F	HPMCOUNTER31	User mode counter 31
0xC80	CYCLEH	The high 32 bits of the cycle, which exists only under RV32
0xC81	TIMEH	The high 32 bits of time, which exists only under RV32
0xC82	INSTRETH	The high 32 bits of instret, which exists only under RV32
0xC83	HPMCOUNTER3H	The high 32 bits of hpmcounter3, which exists only under RV32
...	...	
0xC9F	HPMCOUNTER31H	The higher 32 bits of hpmcounter31 exist only under RV32
0x008	VSTART	Vector start position register
0x009	VXSAT	Fixed point overflow flag bit register
0x00A	VXRM	Fixed-point rounding mode register
0x00F	VCSR	Vector control and status registers
0xC20	VL	Vector length register
0xC21	VTYPE	Vector data type register
0xC22	VLENB	Vector width in bytes

3.1.3.2 Extended Control Register

Table 3-4 Summary of C908 Extended Machine Mode Control Registers

Register number	Name	Description
0x7C0	MXSTATUS	Machine Mode Extended Status Register
0x7C1	MHCR	Machine mode hardware configuration register
0x7C2	MCOR	Machine mode hardware operation register
0x7C3	MCCR2	Machine mode L2C ache control register
0x7C4	MCER2	Machine mode L2Cache ECC register
0x7C5	MHINT	Machine mode implicit operation register
0x7C7	MRVBR	Machine mode reset vector base address register

Register number	Name	Description
0x7C8	MCER	Machine mode L1Cache ECC register
0x7C9	MCOUNTERWEN	Super User Mode Counter Write Enable Register
0x7CA	MCOUNTERINTEN	Machine mode event interrupt enable register
0x7CB	MCOUNTEROF	Overflow annotation register on machine mode events
0x7F0	MHPMCR	Performance Monitoring Control Register
0x7F1	MHPMSR	Performance Monitor Start Trigger Register
0x7F2	MHPMER	Performance monitoring termination trigger register
0x7F5	MZONEID	zoneid register
0x7F6	MLLCPID	Last-level cache fine-grained ID register
0x7F7	ML2WP	L2 cache fine-grained backfill configuration register
0x7D2	MCINS	Machine Mode C ache Instruction Register
0x7D3	MCINDEX	Machine Mode Cache Access Index Register
0x7D4	MCDATA0	Machine mode Ca che Data register 0
0x7D5	MCDATA1	Machine mode Ca che Data register 1
0x7D6	MEICR	L1Cache hardware error injection register
0x7D7	MEICR2	L2Cache hardware error injection register
0x7D8	MBEADDR	L1 LD BUS ERR Address Register
0xFC0	MCPUID	Machine mode processor model register
0xFC1	MAPBADDR	On-chip bus base address
0x7F3	MSMPR	Snoop Listening Enable Register
0xFE0	MHALTCAUSE	HALT Cause Register
0xFE1	MDBGINFO	Debug Information Register
0xFE2	MPCFIFO	pcfifo registers
0xFE3	MDBGINFO2	Debug information register 2

Table 3-5 Summary of C908 Extended Super User Mode Control Registers

Offset Address	Register	Description
0x5C0	SXTATUS	Super User Mode Extended Status Register
0x5C1	SHCR	Super User Mode Hardware Control Register
0x5C2	SCER2	Super User Mode L2Cache ECC Register
0x5C3	SCER	Super user mode L1Cache ECC register
0x5C4	SCOUNTERINTEN	Super User Mode Event Interrupt Enable Register
0x5C5	SCOUNTEROF	Overflow annotation register on superuser mode events

Offset Address	Register	Description
0x5C8	SHPMINHIBIT	Super User Mode Time Disable Count Register
0x5C9	SHPMCR	Super User Mode Performance Monitoring Control Register
0x5CA	SHPMSR	Super user mode monitoring start trigger register
0x5CB	SHPMER	Super user mode monitoring termination trigger register
0x5CC	SLLCPID	Superuser mode end-level cache fine-grained ID register
0x5CD	SL2WP	Superuser mode L2 cache fine-grained backfill configuration register
0x5CE	SBEADDR	Super User Mode L1 LD BUS ERR Address Register
0x5E0	SCYCLE	Super User Mode Cycle Counter
...	...	
0x5FF	SHPMCOUNTER31	Super User Mode Counter 31
0x9C0	SMIR	Super User Mode MMU Index Register
0x9C1	SMEL	Super User Mode MMU EntryLo Register
0x9C2	SMEH	Super User Mode MMU EntryHi Register
0x9C3	SMCIR	Super User Mode MMU Control Register

Table 3-6 Summary of C908 Extended User Mode Control Registers

Offset Address	Register	Description
0x800	fxcrr	User Mode Extended Floating Point Control Register

3.1.3.3 CLIC Register

Base Addr:0x0f_0000_0000 Table 3-7 Summary of C908 CLIC Registers

Offset Address	Register	Description
0x4000000	MSIP0	Core 0 Machine mode software interrupts
0x4000004	MSIP1	Core 1 Machine mode software interrupts
...	...	...
0x4000000+4*n	MSIPN	n=hart_id, n<256
0x4004000	MTIMECMPL0	Core 0 Machine mode clock timer,Compare Value Register (low 32 bits)
0x4004004	MTIMECMP0	Core 0 Machine mode clock timer,Compare Value Register (high 32 bits)
0x4004008	MTIMECMPL1	Core 1 Machine mode clock timer,Compare Value Register (low 32 bits)
0x400400c	MTIMECMP1	Core 1 Machine mode clock timer,Compare Value Register (high 32 bits)
...	...	...
0x4004000+8*n	MTIMECMPLN	n=hart_id, n<256

Offset Address	Register	Description
0x4004000+8*n+4	MTIMECMPN	n=hart_id, n<256
0x400bff8	MTIME	Machine Mode Clock Timer
0x400bffc	MTIMEH	Machine Mode Clock Timer
0x400c000	SSIP0	Core 0 super user mode software interruption
...	...	...
0x400c000+4*n	SSIPN	n=hart_id , n<256
0x400d000	STIMECMPL0	"Core 0 Super User Mode Clock Timer Comparison Value Register
(lower 32 bits)"		
0x400d004	STIMECMPH0	"Core 0 Super User Mode Clock Timer Comparison Value Register
(high 32 bits)"		
0x400d000+8*n	STIMECMPLN	n=hart_id , n<256
0x400d000+8*n+4	STIMECMPHN	n=hart_id , n<256
0x400fff8	STIMEL	Super user mode clock timer
0x400fffc	STIMEH	Super user mode clock timer

3.1.3.4 PLIC Register

Base Addr:0x0f_0000_0000 Table 3-8 Summary of C908 PLIC Registers

Offset Address	Register	Description
0x0000004	PLIC_PRIO1	Interrupt source 1 priority configuration register
0x0000008	PLIC_PRIO2	Interrupt source 2 priority configuration register
...	...	...
0x0000FFC	PLIC_PRIO1023	Interrupt source 1023 priority configuration register
0x0001000	PLIC_IP0	"Interruption No. 1~31
Interrupt Wait Register"		
...	...	...
0x000107C	PLIC_IP31	Interruption No. 992 ~ 1023Interrupt Wait Register
0x0002000	PLIC_H0_MIE0	Core 0 1~31 machine mode interrupt enable register
...	...	...
0x000207C	PLIC_H0_MIE31	Core 0 992 ~ 1023 machine mode interrupt enable register
0x01FFFFC	PLIC_CTRL	PLIC authority control register
0x0200000	PLIC_H0_MTH	Core 0 machine mode interrupt threshold register
0x0200004	PLIC_H0_MCL	Core 0 machine mode interrupt response/completion register

Offset Address	Register	Description
0x0201000	PLIC_H0_STH	Core 0 superuser mode interrupt threshold register
0x0201004	PLIC_H0_SCLA	Core 0 super user mode interrupt response/completion register

3.1.4 Register Description

3.1.4.1 System Control Register

MVENDORID

The Machine Mode Vendor Number Register (MVENDORID) stores the vendor number information for Pinto Semiconductor Ltd. and is currently undefined with an all-zero value. The bit length of this register is 64 bits, and the read/write permission of the register is machine mode read-only, i.e., both non-machine mode access and machine mode write operation access will result in an illegal instruction exception.

Offset:0xF11 Total Reset Value:0x5b7

MARCHID

The Machine Mode Architecture Number Register (MARCHID) stores the architecture number of the processor core and is used to store the internal number of the Pinto Semiconductor Ltd. product, whose reset value is determined by the product itself. The bit length of this register is 64 bits, and the read/write permission of the register is machine mode read-only, i.e. both non-machine mode access and machine mode write operation access will result in an illegal instruction exception.

Offset:0xF12 Total Reset Value:0x8000_0000_0914_0d00

MIMPID

The Machine Mode Hardware Implementation Number Register (MIMPID) stores the hardware implementation number of the processor core. The bit length of this register is 64 bits, and the read/write permission of the register is machine mode read-only, i.e. both non-machine mode access and machine mode write operation access will result in an illegal instruction exception.

Offset:0xF13 Total Reset Value:0x5_0000

MHARTID

The Machine Mode Logical Core Number Register (MHARTID) stores the hardware logical core number of the processor core. The bit length of this register is 64 bits, and the read/write permission of the register is machine mode read-only, i.e. both non-machine mode access and machine mode write operation access will result in an illegal instruction exception.

Offset:0xF14 Total Reset Value:0x0

MCONFIGPTR

The machine mode configuration data structure pointer (MCONFIGPTR) stores the physical address corresponding to the configuration data structure. When the value of this register is 0, it means that no such data structure exists. The bit length of this register is 64 bits, and the read/write permission of the register is machine mode read-only, i.e. both non-machine mode access and machine mode write operation access will result in an illegal instruction exception.

Offset:0xF15 Total Reset Value:0x0

MSTATUS

The Machine Mode Processor Status Register (MSTATUS) stores the status and control information of the processor in machine mode, including global interrupt valid bits, exception reserved interrupt valid bits, exception reserved privileged mode bits, etc.

Offset:0x300 Total Reset Value:0x0000_0022_0000_1900

Bits	Access	Name	Description	Reset
63	R/W	SD	SD-floating point, vector and extended cell dirty state sum bits. When SD=1, it means that the floating point or vector or extension unit is in dirty state. When SD=0, it means that the floating point and vector and expansion cells are not in the dirty state.	0x0
62:36	R	RESERVED	RESERVED	0x0
35:34	R	SXL	register bit width. means that the bit width of the register is 64bit in the S state.	0x2
33:32	R	UXL	means that the bit width of the register is 64bit in the U state.	0x2
31:25	R	RESERVED	RESERVED	0x0
24:23	R	RESERVED	RESERVED	0x0
22	R/W	TSR	trap sret. When TSR=1, the super user mode executes the sret instruction and generates an illegal instruction exception. When TSR=0, the sret command is allowed to be executed in superuser mode.	0x0
21	R/W	TW	Timeout Waiting. When TW=1, the super user mode executes the low power instruction wfi, generating an illegal instruction exception. When TW=0, the super user mode executes the low power command wfi.	0x0
20	R/W	TVM	Trap Virtual Memory. When TVM=1, super user mode reads and writes satp control registers and executes sfence instruction, generating illegal instruction exception. When TVM=0, the super user mode can read and write satp control registers and execute the sfence instruction.	0x0
19	R/W	MXR	Allow load requests to access memory space marked as executable. When MXR=1, the load request is allowed to access the virtual memory space marked as executable or readable. When MXR=0, the load request is allowed to access only the virtual memory space marked as readable.	0x0
18	R/W	SUM	Allows access to U-state virtual memory space in superuser mode. When SUM=1, load, store, and fetch requests in superuser mode can access the virtual memory space marked as user state. When SUM=0, load, store, and fetch requests in superuser mode do not have access to virtual memory space marked as user state.	0x0
17	R/W	MPRV	Modified Privilege Mode. When MPRV=1, load and store requests are executed according to the privileged state in the MPP. When MPRV=0, load and store requests are executed according to the privileged mode the processor is currently in.	0x0
16:15	R/W	XS	Expansion cell status bit	0x0

Bits	Access	Name	Description	Reset
14:13	R/W	FS	Floating point unit status bits. Based on the floating point status bits, you can determine whether the floating point related registers need to be saved when the context is switched. When FS is 2'b00, the floating point unit is off, and accessing the floating point related registers at this time will generate an exception. When FS is 2'b01, the floating-point unit is in the initialized state. When FS is 2'b10, the floating-point unit is in the clean state. When FS is 2'b11, the floating point unit is in the dirty state, indicating that the floating point register and control register have been modified.	0x0
12:11	R/W	MPP	Machine mode reserved privileged status bits. This bit is used to save the privileged state of the processor before the machine mode enters the exception service procedure. When MPP is 2'b00, it indicates that the processor is in user mode before entering the exception service procedure. When the MPP is 2'b01, it indicates that the processor is in superuser mode before entering the exception service procedure. When MPP is 2'b11, the processor is in machine mode before entering the exception service procedure; this bit will be reset to 2'b11.	0x3
10:9	R/W	VS	vector unit status bits. Based on the vector status bits, you can determine whether the vector-related registers need to be saved when the context is switched. When VS is 2'b00, the vector unit is off, and accessing the vector-related registers at this time will generate an exception. When VS is 2'b01, the vector unit is in the initialized state. When VS is 2'b10, the vector unit is in the clean state. When VS is 2'b11, the vector unit is in the dirty state, indicating that the vector register and vector control register have been modified. The VS bit is valid only when the vector execution unit is configured and is constant to 0 when not configured.	0x1
8	R/W	SPP	Superuser mode retains privileged status bits. This bit is used to save the privileged state of the processor before it enters the exception service procedure in superuser mode. When SPP is 2'b00, it indicates that the processor is in user mode before entering the exception service procedure. When SPP is 2'b01, the processor is in super user mode before entering the exception service program; this bit will be reset to 2'b01.	0x0
7	R/W	MPIE	Machine mode reserved interrupt enable bit. This bit is used to hold the value of the MIE bit before the processor responds to an interrupt in machine mode. This bit is cleared by reset and is set to 1 when the processor exits the interrupt exception service routine.	0x0
6	R	RESERVED	RESERVED	0x0

Bits	Access	Name	Description	Reset
5	R/W	SPIE	Super User Mode Reserved Interrupt Enable Bit. This bit is used to store the value of the SIE bit before the processor responds to an interrupt in superuser mode. This bit is cleared by reset and is set to 1 when the processor exits the interrupt service program.	0x0
4	R	RESERVED	RESERVED	0x0
3	R/W	MIE	Machine mode interrupt enable bit. When MIE is 0, the machine mode interrupt is invalid. When MIE is 1, the machine mode interrupt is active. This bit is cleared by reset and is cleared when the processor responds to an interrupt in machine mode; it is set to the value of MPIE when the processor exits the interrupt service program.	0x0
2	R	RESERVED	RESERVED	0x0
1	R/W	SIE	Super User Mode Interrupt Enable bit. When the SIE is 0, the superuser interrupt is invalid. When the SIE is 1, the superuser interrupt is active.	0x0
0	R	RESERVED	RESERVED	0x0

MISA

The Machine Mode Processor Instruction Set Characteristics Register (MISA) stores the instruction set architectural characteristics supported by the processor. The bit length of this register is 64 bits, and the read/write permission of the register is machine-mode read/write, i.e., any non-machine mode access will result in an illegal instruction exception.
Offset:0x301 Total Reset Value:0x8000_0000_00b4_112f

MEDELEG

The Machine Mode Exception Degradation Control Register (MEDELEG) can degrade exceptions that occur in Super User and User modes to Super User mode responses. The MEDELEG low 16 bits correspond to the exception vector table, and you can choose which exceptions to demote to the superuser mode response. The bit length of this register is 64 bits, and the read/write permission of the register is machine-mode read/write, i.e., any non-machine mode access will result in an illegal instruction exception.
Offset:0x302 Total Reset Value:0x0

MIDELEG

The Machine Mode Interrupt Degradation Control Register (MIDELEG) can degrade a super user mode interrupt to a super user mode response.
Offset:0x303 Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset
63:18	R	RESERVED	RESERVED	0x0
17	R/W	MOIE_DELEG		0x0
16	R/W	MCIE_DELEG		0x0
15:10	R	RESERVED	RESERVED	0x0
9	R/W	SEIE_DELEG		0x0
8:6	R	RESERVED	RESERVED	0x0
5	R/W	STIE_DELEG		0x0

Bits	Access	Name	Description	Reset
4:2	R	RESERVED	RESERVED	0x0
1	R/W	SSIE_DELLEG		0x0
0	R	RESERVED	RESERVED	0x0

MIE

The machine mode interrupt enable control register (MIE) is used to control the enable and mask of different interrupt types. The bit length of this register is 64 bits and the read/write permission of the register is machine mode readable and writable, i.e. any non-machine mode access will result in an illegal instruction exception.

Offset:0x304 Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset
63:18	R	RESERVED	RESERVED	0x0
17	R/W	MOIE	Machine mode event counter overflow interrupt enable bit. When MOIE is 0, the machine mode counter overflow interrupt is invalid. When MOIE is 1, the machine mode counter overflow interrupt is valid.	0x0
16	R/W	MCIE	Machine mode ECC interrupt enable bit. When MCIE is 0, the machine mode ECC interrupt is invalid. When MCIE is 1, the machine mode ECC interrupt is active.	0x0
15:12	R	RESERVED	RESERVED	0x0
11	R/W	MEIE	Machine mode external interrupt enable bit. When MEIE is 0, the machine mode external interrupt is invalid. When MEIE is 1, the machine mode external interrupt is valid.	0x0
10	R	RESERVED	RESERVED	0x0
9	R/W	SEIE	Super User Mode External Interrupt Enable bit. When SEIE is 0, the super user mode external interrupt is invalid. When SEIE is 1, the Super User Mode external interrupt is valid.	0x0
8	R	RESERVED	RESERVED	0x0
7	R/W	MTIE	Machine Mode Timer Interrupt Enable bit. When MTIE is 0, the machine mode timer interrupt is invalid. When MTIE is 1, the machine mode timer interrupt is active.	0x0
6	R	RESERVED	RESERVED	0x0
5	R/W	STIE	Super User Mode Timer Interrupt Enable bit. When STIE is 0, the Super User Mode timer interrupt is invalid. When STIE is 1, the Super User Mode Timer external interrupt is valid.	0x0
4	R	RESERVED	RESERVED	0x0
3	R/W	MSIE	Machine mode software interrupt enable bit. When MSIE is 0, the machine mode software interrupt is invalid. When MSIE is 1, the machine mode software interrupt is active.	0x0
2	R	RESERVED	RESERVED	0x0

Bits	Access	Name	Description	Reset
1	R/W	SSIE	Super User Mode Software Interrupt Enable bit. When SEIE is 0, the Super User Mode software external interrupt is invalid. When SEIE is 1, the Super User Mode software external interrupt is valid.	0x0
0	R	RESERVED	RESERVED	0x0

MTVEC

The machine mode vector base address register (MTVEC) is used to configure the entry address of the exception service program. The bit length of this register is 64 bits, and the read/write permission of the register is machine-mode read/write, i.e., any non-machine mode access will result in an illegal instruction exception.

Offset:0x305 Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset
63:2	R/W	base	vector base address bits. The vector base address bit indicates the high 62 bits of the exception service program entry address, which is stitched 2'b00 to obtain the exception service program entry address. This bit will be cleared by reset.	0x0
1:0	R/W	mode	vector entry mode bit. When MODE[1:0] is 2'b00, both exceptions and interrupts use the BASE address as the exception entry address. When MODE[1:0] is 2'b01, the exception uses the BASE address as the entry address and the interrupt uses BASE + 4*cause.	0x0

MCOUNTEREN Machine mode counter access authorization register.

Offset:0x306 Total Reset Value:0x

MSCRATCH The Machine Mode Exception Temporary Data Backup Register (MSCRATCH) is used by the processor to backup temporary data in the exception service routine. It is generally used to store the entry pointer value of the machine mode local context space.

Offset:0x340 Total Reset Value:0x0

MEPC

The Machine Mode Exception Hold Program Counter (MEPC) is used to store the program counter value (i.e., the PC value) when a program exits from an exception service routine.

Offset:0x341 Total Reset Value:0x0

MCAUSE

The Machine Mode Exception Event Vector Register (MCAUSE) is used to store the vector number of the exception event that triggered the exception and is used to process the corresponding event in the exception service program.

Offset:0x342 Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset

Bits	Access	Name	Description	Reset
63	R/W	interrupt	Interrupt flag bit. When the Interrupt bit is 0, it means that the source of the triggered exception is not an interrupt, and the Exception Code is resolved according to the exception. When the Interrupt bit is 1, it means that the source of the exception is an interrupt and the Exception Code is resolved according to the interrupt.	0x0
62:5	R	RESERVED	RESERVED	0x0
4:0	R/W	Exception code	Exception vector number bit. When the processor enters an exception, the exception vector bits are updated to the corresponding values of the exception source.	0x0

MIP

The Machine Mode Interrupt Waiting Status Register (MIP) is used to store the processor's interrupt waiting status. When the processor has a situation where an interrupt cannot be responded to immediately, the corresponding bit in the MIP register will be set. Write the MSIP and SSIP registers in CLINT to start the corresponding interrupt, and after the interrupt is valid, the corresponding bit in MIP can be used as MSIP bit.and SSIP bit for querying. The bit length of this register is 64 bits, and the read/write permission of the register is machine-mode read/write, i.e., any non-machine mode access will result in an illegal instruction exception.

Offset:0x344 Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset
63:18	R	RESERVED	RESERVED	0x0
17	R/W	MOIP	Machine mode event counter overflow interrupt wait bit. When MOIP is 0, the processor currently has no machine mode counter overflow interrupts in the waiting state. When MOIP is 1, the processor currently has a machine mode counter overflow interrupt in the wait state.	0x0
16	R/W	MCIP	Machine mode ECC interrupt wait bit. When MCIP is 0, the processor currently has no machine mode ECC interrupts in the waiting state. When MCIP is 1, the processor currently has a machine mode ECC interrupt in the waiting state.	0x0
15:12	R	RESERVED	RESERVED	0x0
11	R/W	MEIP	External Interrupt Waiting Bit. When MEIP is 0, the processor currently has no machine mode external interrupts in the waiting state. When MEIP is 1, the processor currently has a machine mode external interrupt in the waiting state.	0x0
10	R	RESERVED	RESERVED	0x0
9	R/W	SEIP	Super User Mode External Interrupt Waiting Bit. When SEIP is 0, the processor currently has no super user mode external interrupts in the waiting state. When SEIP is 1, the processor currently has a super user mode external interrupt in the wait state.	0x0

Bits	Access	Name	Description	Reset
8	R	RESERVED	RESERVED	0x0
7	R/W	MTIP	Machine mode timer interrupt wait bit. When MTIP is 0, the processor currently has no machine mode timer interrupts in the waiting state. When MTIP is 1, the processor currently has a machine mode timer interrupt in the waiting state.	0x0
6	R	RESERVED	RESERVED	0x0
5	R/W	STIP	Super User Mode Timer Interrupt Wait bit. When STIP is 0, the processor currently has no super user mode timer interrupts in the waiting state. When STIP is 1, the processor currently has a super user mode timer interrupt in the waiting state.	0x0
4	R	RESERVED	RESERVED	0x0
3	R	MSIP	Machine mode software interrupt wait bit. When MSIP is 0, the processor currently has no machine mode software interrupts in the waiting state. When MSIP is 1, the processor currently has a machine mode software interrupt in the waiting state. This bit is read-only.	0x0
2	R	RESERVED	RESERVED	0x0
1	R	SSIP	Super User Mode Software Interrupt Waiting Bit. When SSIP is 0, the processor currently has no superuser mode software interrupts in the waiting state. When SSIP is 1, the processor currently has a superuser mode software interrupt in the waiting state.	0x0
0	R	RESERVED	RESERVED	0x0

MENVCFG The machine mode environment configuration register (MENVCFG) is used to control the features in the execution environment when it is below machine mode.

Offset:0x30A Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset
63	R/W	STCE	Super User Mode Timer Interrupt Compare Value Register (STIMECMP) function enable bit. •When STCE is 0 When STCE is 0, CLINT The Super User Mode Timer interrupt is generated when the Super User Mode Timer Compare Value Register High/Low (STIMECMPH/STIMECMPL) in the interrupt controller is less than or equal to the system timer. STIMECMPH/STIMECMPL are memory address mapping registers. •When STCE is 1, the Super User Mode Timer Interrupt Compare Value Register (STIMECMP) is less than or equal to the System Timer to generate a Super User Mode Timer Interrupt. STIMECMP is the CSR register.	0x0
62:1	R	RESERVED	RESERVED	0x0

Bits	Access	Name	Description	Reset
0	R/W	FIOM	IO fence contains memory access. For the RV standard definition, with <code>menvcfg.FIOM = 1</code>, IO sync requests below machine mode need to also include memory RW sync. Note: All fence synchronization to IO in the Xantra C908 will include RW synchronization, so it is no longer controlled by this bit. Regardless of the value of this bit, the IO The fence sync will include the RW sync.	0x0

PMPCFG0~PMPCFG14 The Machine Mode Physical Memory Protection Configuration Register (PMPCFG) is used to configure the access rights, address matching mode of physical memory.

Offset:0x3A0~0x3AE Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset
63:56	R/W	entry7_cfg	The physical memory protection setting register provides permission settings for PMP table entries.	0x0
55:48	R/W	entry6_cfg	The physical memory protection setting register provides permission settings for PMP table entries.	0x0
47:40	R/W	entry5_cfg	The physical memory protection setting register provides permission settings for PMP table entries.	0x0
39:32	R/W	entry4_cfg	The physical memory protection setting register provides permission settings for PMP table entries.	0x0
31:24	R/W	entry3_cfg	The physical memory protection setting register provides permission settings for PMP table entries.	0x0
23:16	R/W	entry2_cfg	The physical memory protection setting register provides permission settings for PMP table entries.	0x0
15:8	R/W	entry1_cfg	The physical memory protection setting register provides permission settings for PMP table entries.	0x0
7:0	R/W	entry0_cfg	The physical memory protection setting register provides permission settings for PMP table entries.	0x0

PMPADDR0~PMPADDR63

The machine mode physical memory address register (PMPADDR) is used to configure the address interval of each table entry of physical memory.

Offset:0x3B0~0x3EF Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset
63:38	R	RESERVED	RESERVED	0x0
37:9	R/W		RISC-V stipulates that the PMP address register stores [39:2] bits of the physical address. Because the C908 PMP table entry granularity supports at least 4KB Bit [8:0] will not be used for address authentication logic.	0x0
8:0	R	RESERVED	RESERVED	0x0

MCYCLE

The machine mode cycle counter (MCYCLE) is used to store the number of cycles that the processor has executed. When the processor is in an execution state (i.e., not in a low-power state), the MCYCLE register is self-incrementing at each execution cycle.

Offset:0xB00 Total Reset Value:0x0000_0000_0000_0000

MINSTRET

The Machine Mode Retired Instruction Counter (MINSTRET) is used to store the number of instructions that the processor has retired. The MINSTRET register will self-increment the count for each instruction as it is retired.

Offset:0xB02 Total Reset Value:0x0000_0000_0000_0000

MHPMCOUNTER3~MHPMCOUNTER31

The machine mode event counter (MHPMCOUNTERn) is used to count events.

Offset:0xB02~B1F Total Reset Value:0x0000_0000_0000_0000

COUNTINHIBIT

The machine mode inhibit count register (mcountinhibit) can disable the machine mode counter counting. When performance analysis is not required, turn off the counter to reduce the power consumption of the processor.

Offset:0x320 Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset
31:3	R/W	MHPMn	The mhpmpcounter register prohibits counting bits: 0: Normal count 1: Suppress counting	0x0
2	R/W	MIR	The minstret register prohibits counting: 0: Normal count 1: Suppress counting	0x0
1	R	RESERVED	RESERVED	0x0
0	R/W	MCY	The mcycle register prohibits counting bits: 0: Normal count 1: Suppress counting	0x0

MHPMEVENT3~MHPMEVENT31

The machine mode counter configuration register (MHPMEVENTn) is used to give the events selected for the machine mode event counter, mhpmevent3-31 and mhpmpcounter3-31 correspond one to the other.

Offset:0x323~0x33F Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset
63:0	R/W	event index	Index of performance monitoring events: 0: No event; 0x1~0xB4: performance monitoring events realized by hardware, see Table for details; >0xB4: Undefined performance monitoring event of hardware, used for software custom event.	0x0

TSELECT

The debug/trace trigger selection register (TSELECT) is used to select one of multiple triggers for the next read/write of that trigger's register.

Offset:0x7A0 Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset
63:4	R	RESERVED	RESERVED	0x0
3:0	R/W	SELECT	Debug/Trace Trigger Selection. Records the debug/trace trigger number that is currently selected. For example, to configure trigger number 2, SELECT writes 0x2.	0x0

TDATA1

Debug/trace trigger data register 1.

Offset:0x7A1 Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset
63:60	R/W	TYPE	debug/trace trigger type selection. TYPE = 2, indicating that the current trigger type is mcontrol. TYPE = 3, indicating that the current trigger type is icount. TYPE = 4, indicating that the current trigger type is ittrigger. TYPE = 5, indicating that the current trigger type is etrigger. c908 supports two types of triggers. 1.mcontrol trigger, with the TYPE field hardwired to 0x2. 2.ittrigger/etrigger/icount configurable triggers, with the TYPE field configurable as 0x3, 0x4, 0x5 and a reset value of 0x5.	0x2/0x5
59	R/W	DEMODE	Controls write access to debug/trace trigger data register 123 (TDATA1/TDATA2/TDATA3). When DEMODE is 0, debug mode and machine mode can be written to the debug/trace trigger data register. When DEMODE is 1, only debug mode can write to the debug/trace trigger data register.	0x0
58:0	R/W	DATA	debug/trace trigger register 1 control. The specific meaning of the DATA field is determined by the TYPE (see section 5.2 of the RISC-V debug spec v0.13.2 for a detailed description).	0x0

TDATA2

Debug/trace trigger data register 2.

Offset:0x7A2 Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset
63:0	R/W	DATA	Debug/Trace Trigger Data Register 2 Data. Used to set the trigger value, the exact meaning of which is determined by the TYPE field in TDATA1.	0x0

TDATA3

Debug/trace trigger data register 3.

Offset:0x7A3 Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset
63:51	R/W	MVALUE	Used to set the value of the machine mode content you wish to match.	0x0

Bits	Access	Name	Description	Reset
50	R/W	MSELECT	When MSELECT is 0: Close the trigger machine pattern content matching	0x0
49:36	R	RESERVED	When MSELECT is 1: This trigger can match the processor information when the machine mode content register (MCONTEXT) is equal to the trigger machine mode content match data (MVALUE).	0x0
35:2	R/W	SVALUE	Used to set the value of the superuser mode content that you want to match.	0x0
1:0	R/W	SSELECT	When SSELECT is 0: Close the trigger super user mode content matching When SSELECT is 1: This trigger can match the processor information when the Super User Mode Content Register (SCONTEXT) is equal to the trigger Super User Mode Content Matching Data (SVALUE).	0x0

TINFO

Debug/trace trigger data register 3.

Offset:0x7A4 Total Reset Value:0x0000_0000_0000_0004

Bits	Access	Name	Description	Reset
63:16	R	RESERVED	RESERVED	0x0
15:0	R/W	INFO	A bit[n] of 1 means that the TYPE field of this trigger TDATA1 can be configured to n. The C908 supports two types of triggers. 1.mcontrol trigger, INFO is hardwired to 0x100, indicating that the TYPE field of TDATA1 can only be 2. 2.itrigger/etrigger/icount configurable trigger, INFO is hardwired to 0x111000, indicating that the TYPE field can be configured to 0x3, 0x4, 0x5.	0x100 or 0x111000

TCONTROL

Debug/Trace Trigger Control Register.

Offset:0x7A5 Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset
63:9	R	RESERVED	RESERVED	0x0
8:7	R/W	MPTE	When MTE is 0: the trigger behaviour is such that the trigger generating the breakpoint exception cannot be triggered in machine mode. When MTE is 1: The trigger can be triggered in machine mode. The hardware sets MTE to 0 when entering the machine mode exception/interrupt handler; when returning from the machine mode exception/interrupt handler, the hardware sets MTE is set to the value of MPTE.	0x0
6:4	R	RESERVED	RESERVED	0x0
3	R/W	MTE	When entering the machine mode exception/interrupt handler, the hardware stores the value of the MTE into the MPTE	0x0
2:0	R	RESERVED	RESERVED	0x0

MCONTEXT Machine mode content register.

Offset:0x7A8 Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset
63:0	R/W	MCONTEXT	The machine mode software can write a specific context that combines the debug/trace trigger data register 3 (TDATA3) in the MSELECT and MVALUE, which controls the trigger to be triggered only in a specific machine mode context.	0x0

MCONTEXT Machine mode content register.

Offset:0x7B0 Total Reset Value:0x4000_0000

Bits	Access	Name	Description	Reset
31:28	R/W	XDEBUGVER	0: No commissioning system 4: There is a debugging system, debugging system support riscv debug spec v0.13.2 15: With debug system, debug system does not support riscv debug spec v0.13.2	0x4
27:16	R	RESERVED	RESERVED	0x0
15	R/W	EBREAKM	0: Machine mode execution of the ebreak instruction generates a breakpoint exception 1: Execute the ebreak command in machine mode to enter debug mode	0x0
14	R	RESERVED	RESERVED	0x0
13	R/W	EBREAKS	0: The ebreak instruction in superuser mode generates a breakpoint exception. 1 : Execute ebreak command in super user mode to enter debug mode	0x0
12	R/W	EBREAKU	0: The execution of the ebreak instruction in user mode generates a breakpoint exception 1: Execute the ebreak command in user mode to enter debug mode	0x0
11	R/W	STEPIE	0: No interrupt response during single-step debugging 1: Respond to interrupt during single-step debugging.	0x0
10	R/W	STOPCOUNT	0: The performance monitoring counter counts normally in debug mode 1: Performance monitoring counters do not count in debug mode,including MCYCLE nuclear MINSTRET .	0x0
9	R/W	STOPTIME	0: The processor core's private clock counter counts normally in debug mode 1: The private clock counter of the processor core in debug mode does not count	0x0

Bits	Access	Name	Description	Reset
8:6	R/W	CAUSE	1: Indicates that the reason for entering debug mode is the execution of the ebreak instruction 2: Indicates that the reason for entering debug mode is trigger triggering 3: Indicates that the reason for entering debug mode is to synchronize debug requests 4: Indicates that the reason for entering debug mode is a single-step debug request 5: Indicates that the reason for entering debug mode is to reset the debug request	0x0
5	R	RESERVED	RESERVED	0x0
4	R/W	MPRVEN	0: MPRV field in MSTATUS register is disabled in debug mode 1: The MPRV field in the MSTATUS register is valid in debug mode, and the processor handles the address translation and protection of access instructions according to the MPRV core MPP setting	0x0
3	R/W	NMIP	0: indicates that no NMI interrupts exist in the processor 1: indicates an NMI interrupt in the processor	0x0
2	R/W	STEP	0: No single-step debugging 1: Initiate single-step debugging mode	0x0
1:0	R/W	PRV	When entering debug mode, the privileged mode of the processor is stored in the PRV; after exiting debug mode, the privileged mode of the processor is set according to the PRV field	0x0

DPC

Debug mode program counter.

Offset:0x7B1 Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset
63:0	R/W	DPC	When entering debug mode, the hardware writes the address of the next instruction to the DPC; when exiting debug mode, the processor starts to take the finger from the address saved in the DPC for execution.	0x0

DSCRATCH0

Debug Mode Temporary Data Backup Register 0.

Offset:0x7B2 Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset
63:0	R/W	DSCRATCH0	Hardware for debugging the data exchange between the system and the processor core.	0x0

DSCRATCH1

Debug Mode Temporary Data Backup Register 1.

Offset:0x7B3 Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset

Bits	Access	Name	Description	Reset
63:0	R/W	DSCRATCH1	Hardware for debugging the data exchange between the system and the processor core.	0x0

SSTATUS

The Super User Mode Processor Status Register (SSTATUS) stores the status and control information of the processor in Super User Mode, including global interrupt valid bits, exception reserved interrupt valid bits, exception reserved privileged mode bits, etc. It is part of the MSTATUS mapping.

Offset:0x100 Total Reset Value:0x0000_0022_0000_1900 Please refer to the Machine Mode Processor Status Register (MSTATUS) for more information.

SIE

The Super User Mode Interrupt Enable Control Register (SIE) is used to control the enable and mask of different interrupt types and is a partial mapping of the MIE. The bit length of this register is 64 bits, and the read/write permission of the register is super user mode readable, and the write permission in super user mode is determined by the mideleg of the corresponding bit, and user mode access will result in illegal instruction exception.

Offset:0x104 Total Reset Value:0x0000_0000_0000_0000 Please refer to the Machine Mode Interrupt Enable Control Register (MIE) for more information.

STVEC

The Super User Mode Vector Base Address Register (STVEC) is used to configure the entry address of the exception service program. The bit length of this register is 64 bits, the register read and write permission is super user mode can read and write, user mode access will lead to illegal instruction exception.

Offset:0x105 Total Reset Value:0x0000_0000_0000_0000 Please refer to the Machine Mode Vector Base Address Register (MTVEC) for more information.

SCOUNTEREN

The super user mode counter access authorization register (scounteren) is used to authorize whether the user mode can access the user mode counter.

Offset:0x106 Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset
31:3	R/W	HPMn	Hpmcountern register U-mode access bit: Illegal instruction exception will occur when 0: U-mode accesses h pmcounter 1: When the corresponding bit of mcounteren is 1, the U-mode can normally access the hpmcoun term, Otherwise, an illegal instruction exception will occur	0x0
2	R/W	IR	Instret register U-mode access bit: 0: Illegal instruction exception will occur when U-mode accesses instret register 1: When the corresponding bit of mcount eren is 1, U-mode can normally access the ins tret register. No Then an illegal instruction exception will occur	0x0

Bits	Access	Name	Description	Reset
1	R/W	TM	Time register U-mode access bit: 0: Illegal instruction exception will occur when U-mode accesses the time register 1: When the corresponding bit of mco unteren is 1, U-mode can normally access the time register, otherwise Illegal instruction exception will occur	0x0
0	R/W	CY	cycle register U-mode access bit: 0: Illegal instruction exception will occur when U-mode accesses the cycle register 1: When the corresponding bit of mcou teren is 1, the U-mode can normally access the cycle register, otherwise an illegal instruction exception will occur	0x0

SENVCFG

The Super User Mode Environment Configuration Register (SENVCFG) is used to control the features in the execution environment when it is below Super User Mode. The bit length of this register is 64 bits, the register read and write permission is super user mode can read and write, user mode access will lead to illegal instruction exception.

Offset:0x10A Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset
63:1	R	RESERVED	RESERVED	0x0
0	R/W	FIOM	The fence synchronization to IO will include RW synchronization regardless of the value of this bit. Note: All fence syncs to IO in the GenTek 908 will include RW syncs, so they are no longer controlled by this bit. For the definition of the RV standard, the When senvcfg.FIOM = 1, IO sync requests below superuser mode need to also include memory RW sync.	0x0

SSCRATCH

The Super User Mode Exception Temporary Data Backup Register (SSCRATCH) is used by the processor to backup temporary data in the exception service routine. It is generally used to store the entry pointer value of the Super User Mode local context space. The bit length of this register is 64 bits, and the read/write permission of the register is super user mode read/write, user mode access will cause illegal instruction exception.

Offset:0x140 Total Reset Value:0x0

SEPC

The Super User Mode Exception Hold Program Counter (SEPC) is used to store the program counter value (i.e., PC value) when a program exits from an exception service routine. the C920 supports 16-bit wide instructions, and the SEPC value is aligned 16 bits wide with the lowest bit being zero. The bit length of this register is 64 bits, and the read/write permission of the register is super user mode read/write, user mode access will cause illegal instruction exception.

Offset:0x141 Total Reset Value:0x0

SCAUSE

The Super User Mode Exception Event Vector Register (SCAUSE) is used to store the vector number of the exception event that triggered the exception and is used to process the corresponding event in the exception service program. The bit length of this register is 64 bits, the register read and write permission is super user mode can read and write,

user mode access will lead to illegal instruction exception.
Offset:0x142 Total Reset Value:0x0

SIP

The Super User Mode Interrupt Waiting Status Register (SIP) is used to store the processor's interrupt waiting status. The corresponding bit in the SIP register is set when the processor has an interrupt that cannot be responded to immediately. The bit length of this register is 64 bits, and the read/write permission of the register is readable in super user mode, and the write permission is determined by the mideleg of the corresponding bit, and any user mode access will result in an illegal instruction exception.
Offset:0x144 Total Reset Value:0x0000_0000_0000_0000 Please refer to the MIP for more information.

STIMECMP

The Super User Mode Timer Interrupt Comparison Value Register (STIMECMP) is used to store the timer comparison value. The value of STIMECMP is compared with the current value of the system timer to determine if a Super User Mode Timer interrupt is generated. No interrupt is generated when STIMECMP is greater than the system timer value; a super user mode timer interrupt is generated when STIMECMP is less than or equal to the system timer value and the STCE field of the Machine Mode Environment Configuration Register (MENVCFG) is 1. Software can clear the Super User Mode Timer interrupt caused by STIMECMP by rewriting the value of STIMECMP. The bit width of this register is 64 bits. Read and write privileges are available in super user mode when the STCE field of the Machine Mode Environment Configuration Register (MENVCFG) is 1 and the TM field of the Machine Mode Counter Access Authorization Register (MCOUNTEREN) is 1. Otherwise, it results in an illegal instruction exception. User mode access causes an illegal instruction exception.
Offset:0x14D Total Reset Value:0xffff_ffff_ffff_ffff

Bits	Access	Name	Description	Reset
63:0	R/W	STIMECMP	Super User Mode Timer Interrupt Compare Value Register	0xffffffffffff

SATP

The Super User Mode Address Translation Register (SATP) is used to control the mode switching, hardware backfill base address and process number of the MMU unit. The bit length of this register is 64 bits, and the read/write access to the register is super user mode readable and writeable; user mode access will result in an illegal instruction exception.
Offset:0x180 Total Reset Value:0x0000_0000_0000_0000_0000

Bits	Access	Name	Description	Reset
63:60	R/W	MODE	The Mode field indicates the virtual address translation mode. 0:Bare 1-7:Reserved 8:Sv39 9:Sv48 10:Sv57 Reserved for page-based 57-bit virtual addressing 11:Sv64 Reserved for page-based 64-bit virtual addressing 12-13:Reserved for standard use 14-15:Designated for custom use	
59:44	R/W	ASID	Current process number	
43:0	R/W	PPN	Hardware backfill root PPN	

SCONTEXT

Super User Mode Content Register
Offset:0x5A8 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
63:0	R/W	SCONTEXT	The superuser mode software can write specific contexts in combination with the debug/trace trigger data register 3 (TDATD3) for SSELECT and SVALUE, you can control the trigger to be triggered only in a specific superuser mode context.	

FFLAGS

The Floating Point Exception Accumulation Status Register (FFLAGS) is an exception accumulation field mapping of the Floating Point Control Status Register (FCSR), refer to the Floating Point Control Status Register (FCSR) for more information.

Offset:0x001 Total Reset Value:0b000000

FRM

The Floating Point Dynamic Rounding Register (FRM) is a mapping of the rounding mode fields of the Floating Point Control Status Register (FCSR), see Floating Point Control Status Register (FCSR) for more information.

Offset:0x002 Total Reset Value:0b0000

FCSR

The Floating Point Control Status Register (FCSR) is used to record the exception accumulation and rounding mode control of the floating point. The bit length of this register is 64 bits and the register can be read or written in any mode.

Offset:0x003 Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset
63:11	R	RESERVED	RESERVED	0x0
10:9	R/W	VXRM	Vector rounding mode bits. The mapping of VXRM to the corresponding bits.	0x0
8	R/W	vXSAT	Vector overflow flag bit. the mapping of the corresponding bit of VXSAT.	0x0
7:5	R/W	RM(FRM)	rounding mode. When RM=0, the RNE rounding mode rounds to the nearest even number. When RM=1, RTZ rounding mode, rounding to 0. When RM=2, RDN rounding mode, rounding to negative infinity. When RM=3, RUP rounding mode, rounding to positive infinity. When RM=4, the RMM rounding mode, rounds to the nearest.	0x0
4	R/W	NV(FFLAGS4)	Invalid operand exception. When NV=0, no invalid operand exception is thrown. When NV=1, an invalid operand exception is thrown.	0x0
3	R/W	DZ(FFLAGS3)	except 0 Exception. When DZ=0, no divide by 0 exception is thrown. When DZ=1, a divide by 0 exception is generated.	0x0
2	R/W	OF(FFLAGS2)	overflow anomaly. When OF=0, no overflow exception is thrown. When OF=1, an overflow exception is generated.	0x0
1	R/W	UF(FFLAGS1)	underflow anomaly. When UF=0, no underflow exception is thrown. When UF=1, an underflow exception is generated.	0x0

Bits	Access	Name	Description	Reset
0	R/W	NX(FFLAGS0)	Non-precision anomalies. When NX=0, no non-exact exception is generated. When NX=1, a non-exact exception is generated.	

CYCLE

The user mode cycle counter (CYCLE) is used to store the number of cycles that have been executed by the processor when the processor is in the execution state (i.e., non-low power state) The CYCLE register is then self-incrementing in count each execution cycle. The cycle counter is 64-bit, and the cycle counter is reset to zero.

Offset:0xC00 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
63:0	R	CYCLE	cycle counter	0x0

TIME

The user mode time counter (TIME) is a read-only mapping of MTIME.

Offset:0xC01 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
63:0	R	TIME	timer	0x0

INSTRET The user mode retired instruction counter (MINSTRET) is used to store the number of instructions that the processor has retired. The MINSTRET register will self-increment the count for each instruction as it is retired.

Offset:0xC02 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
63:0	R	INSTRET	instructions-retired counter	0x0

HPMCOUNTER3~HPMCOUNTER11

The user mode event counter (HPMCOUNTERn) is a mapping of the machine mode event counter (MHPMCOUNTERn)

Offset:0xC03~0xC1F Total Reset Value:0x0

Bits	Access	Name	Description	Reset
63:0	R	HPMCOUNTERn	performance-monitoring counter	0x0

VSTART

The vector start position register specifies the starting element position when executing vector instructions, and VSTART will be cleared after each vector instruction is executed.

Offset:0x008 Total Reset Value:0x0

VXSAT

The fixed-point overflow flag bit register indicates whether a fixed-point instruction produces an overflow result.

Offset:0x009 Total Reset Value:0x0

VXRM

The fixed-point rounding mode register specifies the rounding mode to be used for fixed-point instructions.

Offset:0x00A Total Reset Value:0x0

Bits	Access	Name	Description	Reset
63:2		Base		0x0
1:0	R/W	RM	Fixed-point rounding mode. When RM=0, the RNU rounding mode, rounds to large numbers. When RM=1, the RNE rounding mode rounds to an even number. When RM=2, the RDN rounding mode, rounding to zero. When RM=3, the ROD rounding mode rounds to odd numbers.	0x0

VL

The vector length register specifies the range in which the vector instruction updates the destination register. The vector instruction updates the elements in the destination register whose element number is less than VL and clears the elements in the destination register whose element number is greater than or equal to VL. In particular, when $VSTART \geq VL$ or VL is 0, all elements of the destination register are not updated. This register is a read-only register in any mode, but the vsetvli, vsetvl, and fault-only-first instructions can update the value of this register.

Offset:0xC20 Total Reset Value:0x0

VTYPE

The VTYPE register specifies the data type of the vector register set and the element composition of the vector register.

Offset:0xC21 Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset
63	R	VILL	Illegal operation flag bit. This position starts when the vsetvli/vsetvl instruction updates the VTYPE register with a value not supported by the C920, otherwise it is 0. When this position starts, execution of the vector instruction generates an illegal instruction exception.	0x0
62:7	R	RESERVED	RESERVED	0x0
6:5	R	VEDIV	EDIV extended enable bit.	0x0
4:2	R/W	VSEW	vector element bit width setting bit. 000:Elemental bit width 8 001:Elemental bit width 16 010:Elemental bit width 32 011:Elemental bit width 64	0x0
1:0	R/W	VLMUL	Vector register grouping setting bit. 00:LMUL 1 01:LMUL 2 10:LMUL 4 11:LMUL 8	0x0

VLENB

The VLENB register indicates the vector bit width of the processor, in bytes. The vector bit width of the C908 is 128 bits (i.e., $VLEN = 128$), so $VLENB = 128/8 = 16$.

Offset:0xC22 Total Reset Value:0x10

3.1.4.2 Extended Control Register

MXSTATUS

The Machine Mode Extended Status Register (MXSTATUS) stores the privileged mode the processor is currently in

and the C908 extended function switch bits. The bit length of this register is 64 bits, and the read/write permission of the register is machine-mode read/write, i.e., any non-machine mode access will result in an illegal instruction exception.

Offset:0x7C0 Total Reset Value:0x0000_0000_8000_8000

Bits	Access	Name	Description	Reset
63:32	R	RESERVED	RESERVED	0x0
31:30	R/W	PM	Processor is in privileged mode When PM=2'b00, it indicates that the processor is currently running in user mode. When PM=2'b01, it indicates that the processor is currently running in super user mode. When PM=2'b11, it indicates that the processor is currently running in machine mode. (Machine mode after Reset)	0x1
29:23	R	RESERVED	RESERVED	0x0
22	R/W	THEADISAEE	Enabling Extended Instruction Set. When THEADISAEE is 0, an illegal instruction exception is thrown when using the C908 extended instruction set. When THEADISAEE is 1, the C908 extended instruction set can be used.	0x0
21	R/W	MAEE	Extension MMU Address Attributes. When MAEE is 0, the MMU address attribute is not extended. When MAEE is 1, the address attribute bit is extended in the pte of the MMU and the user can configure the address attribute of the page.	0x0
20	R	RESERVED	RESERVED	0x0
19	R/W	INSDE	Disable Icache snoop Dcache function. When INSDE is 0, the Icache will snoop Dcache when it is missing. when INSDE is 1, the Icache will not snoop Dcache when it is missing.	0x0
18	R/W	MHRD	Close hardware backfill. When MHRD is 0, the hardware will perform a hardware backfill when the TLB is missing. When MHRD is 1, the hardware does not perform a hardware backfill when the TLB is missing.	0x0
17	R/W	CLINTEE	Clint Timer/Software Interrupt Super User Extended Enable Bit. When CLINTEE is 0, superuser software interrupts and timer interrupts initiated by CLINT are not responded to. When CLINTEE is 1, superuser software interrupts and timer interrupts initiated by CLINT can be responded to.	0x0
16	R/W	UCME	U state executes the extended cache instruction. When UCME is 0, the user mode cannot execute extended cache operation instructions and generates an illegal instruction exception. When UCME is 1, user mode can execute extended cache operation instructions.	0x0

Bits	Access	Name	Description	Reset
15	R/W	MM	Non-aligned access enable bit When MM is 0, non-aligned accesses are not supported and non-aligned accesses will generate non-aligned exceptions. When MM is 1, non-aligned accesses are supported and hardware handles non-aligned accesses. (C908 default value is 1)	0x1
14	R/W	PMP4K	PMP minimum granularity control bit The C908 currently only supports PMP with a minimum granularity of 4K, which is not affected by this bit.	0x0
13	R/W	PMDM	Machine Mode Performance Monitoring Count enable bit. When PMDM is 0, performance counter counting is allowed in machine mode. When PMDM is 1, performance counter counting is disabled in machine mode.	0x0
12	R	RESERVED	RESERVED	0x0
11	R/W	PMDS	Super User Mode Performance Monitoring Count enable bit. When PMDS is 0, performance counter counting is allowed in Super User mode. When PMDS is 1, performance counter counting is disabled in Super User mode.	0x0
10	R/W	PMDU	User mode performance monitoring count enable bit. When PMDU is 0, performance counter counting is allowed in user mode. When PMDU is 1, performance counter counting is disabled in user mode.	0x0
9:8	R	RESERVED	RESERVED	0x0
7:0	R	RESERVED	RESERVED	0x0

MHCR

The Machine Mode Hardware Configuration Register (MHCR) is used to configure the processor, including performance and features. The bit length of this register is 64 bits, and the read/write permission of the register is machine-mode read/write, i.e., any non-machine mode access will result in an illegal instruction exception.

Offset:0x7C1 Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset
63:19		RESERVE	RESERVE	0x0
18	R/W	SCK	System to processor clock ratio. This bit is used to indicate the clock ratio between the system and the processor, which is calculated as: Clock ratio = SCK+1, with the corresponding pin on the CPU. SCK is set at reset It is configured at the time of the change and cannot be changed later.	0x0
17:13		RESERVE	RESERVE	0x0
12	R/W	LOBTB	First level branch target prediction enable bit. When LOBTB=0, the first level branch target prediction is off. When LOBTB=1, the first level branch target prediction is on.	0x0
11:9		RESERVE	RESERVE	0x0

Bits	Access	Name	Description	Reset
8	R/W	WBR	Write Burst Transmit Enable Bit. When WBR=0, write burst transfer is not supported. When WBR=1, write burst transfer is supported. C908 Default is 1 and cannot be set.	0x0
7	R/W	IBPE	Indirect Branch Jump Predictive Jump Enable bit. When IBPE=0, Indirect Branch Jump Prediction is off. When IBPE=1, indirect branch jump prediction is on.	0x0
6	R/W	BTB	Branch Target Prediction enable bit. When BTB=0, branch target prediction is off. When BTB=1, branch target prediction is on.	0x0
5	R/W	BPE	Allow Predictive Jump setting bit. When BPE=0, the predictive jump is off. When BPE=1, predictive jumping is on.	0x0
4	R/W	RS	address return stack setting bit. When RS=0, the return stack is closed. When RS=1, the return stack is on.	0x0
3	R/W	WB	Cache write back setting bit. When WB=0, the data cache is in write straight mode. When WB=1, the data cache is in write back mode. c908 only supports write back mode, WB is fixed to 1.	0x0
2	R/W	WA	Cache Write Allocation setting bit. When WA=0, the data cache is in write non-allocate mode. When WA=1, the data cache is in write allocate mode.	0x0
1	R/W	DE	Dcache enable bit. When DE=0, Dcache is off. When DE=1, Dcache is on.	0x0
0	R/W	IE	Icache enable bit. When IE=0, Icache is off. When IE=1, Icache is on.	0x0

MCOR The Machine Mode Hardware Operation Register (MCOR) is used to perform related operations on the cache and branch prediction components. The bit length of this register is 64 bits, and the read/write permission of the register is machine-mode read/write, i.e., any non-machine mode access will result in an illegal instruction exception.

Offset:0x7C2 Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset
63:19		RESERVED	RESERVED	0x0
18		IBP_INV	IBP Invalid setting bit. When IBP_INV=0, the data predicted by the indirect jump branch is not invalidated. When IBP_INV=1, the data predicted by the indirect jump branch is invalidated. All the above invalidation operations and clear dirty expression operations are set high when writing and clear 0 when the operation is completed.	0x0

Bits	Access	Name	Description	Reset
17		BTB_INV	BTB Invalid setting bit. When BTB_INV=0, the data in the branch target buffer is not invalidated. When BTB_INV=1, the data in the branch target buffer is invalidated.	0x0
16		BHT_INV	BHT Invalid setting bit. When BHT_INV=0, the data in the branch history table is not invalidated. When BHT_INV=1, the data in the branch history table is invalidated.	0x0
15:6		RESERVED	RESERVED	0x0
5		CLR	Cache dirty performance clear setting bit. When CLR=0, table entries that are marked as dirty by the cache are not written off-chip. When CLR=1, table entries that are marked as dirty by the cache are written off-chip.	0x0
4		INV	Cache Invalidation Setting Bit. When INV=0, the cache is not invalidated. When INV=1, the cache is invalidated.	0x0
3:2		RESERVED	RESERVED	0x0
1:0		CACHESEL	Cache selection bit. When CACHE_SEL=2'b01, the instruction cache is selected. When CACHE_SEL=2'b10, the data cache is selected. When CACHE_SEL=2'b11, the instruction and data cache is selected.	0x0

MCCR2 The Machine Mode L2Cache Control Register (MCCR2) is used to configure the access latency of each memory in the shared L2 cache, L2 cache valid/invalid, instruction prefetch capability and TLB prefetch enable, and ECC checksum enable. The bit length of this register is 64 bits, and the read/write permission of the register is machine-mode read/write, i.e., any non-machine mode access will result in an illegal instruction exception. Offset:0x7C3 Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset
63:32		RESERVED	RESERVED	0x0
31	R/W	TPRF	L2Cache TLB Prefetch Enable. When TPRF is 0, the L2Cache TLB prefetching function is off; when TPRF is 1, the L2Cache TLB prefetching function is on.	0x0
30:29	R/W	IPRF	L2Cache command prefetch capability. Indicated fetch means the number of cache lines prefetched when requesting access to an L2Cache miss. When IPRF is 0, L2Cache command prefetching is disabled; when IPRF is 1, 1 cache line is prefetched. When the IPRF is 2, 2 cache lines are prefetched; when the IPRF is 3, 3 cache lines are prefetched.	0x0
28:26		RESERVED	RESERVED	0x0

Bits	Access	Name	Description	Reset
25	R/W	TSETUP	L2 CACHE TAG RAM setup configuration bits. When TSETUP is 0, the TAG RAM does not require an additional setup cycle; when TSETUP is 1, the TAG RAM requires an additional 1 setup cycle.	0x0
24:22	R/W	TLTNCY	L2Cache TAG RAM access cycle configuration bits. When TLTNCY is 0, the TAG RAM access period is 1; when TLTNCY is 1, the TAG RAM access period is 2; when TLTNCY is 2, the TAG RAM access period is 3; when TLTNCY is 3, the TAG RAM access period is 4; when TLTNCY is 4, the TAG RAM access period is 5.	0x0
21:20		RESERVED	RESERVED	0x0
19	R/W	DSETUP	L2Cache DATA RAM setup configuration bits. When DSETUP is 0, the DATA RAM does not require an additional setup cycle; when DSETUP is 1, the DATA RAM requires an additional setup cycle.	0x0
18:16	R/W	DLTNCY	L2Cache DATA RAM access cycle configuration bit. When DLTNCY=0, the DATA RAM access period is 1. When DLTNCY=1, the DATA RAM access period is 2. When DLTNCY is 2, the DATA RAM access period is 3. When DLTNCY is 3, the DATA RAM access period is 4. When DLTNCY is 4, the DATA RAM access period is 5. When DLTNCY is 5, the DATA RAM access period is 6. When DLTNCY is 6, the DATA RAM access period is 7. When DLTNCY is 7, the DATA RAM access period is 8.	0x0
15:4		RESERVED	RESERVED	0x0
3	R/W	L2EN	L2Cache enable bit. When L2EN=0, L2Cache is turned off. When L2EN=1, L2Cache is turned on. (C908 fixed to 1)	0x0
2		RESERVED	RESERVED	0x0
1	R/W	ECCEN	ECC enable bit. When ECCEN=0, L2Cache ECC is off. When ECCEN=1, L2Cache ECC is on.	0x0
0	R/W	RFE	Data Access Read Assignment Enable bit. When RFE=0, the L2Cache is not backfilled when the data access L2Cache is missing. when RFE=1, the L2Cache is backfilled when the data access L2Cache is missing.	0x0

MCER2

The Machine Mode L2 Cache ECC Control Register (MCER2) is used to configure the L2 Cache ECC. The L2 cache supports configurable ECC for 1-bit error correctable and 2-bit error detectable functionality. When a 2-bit error is detected, the hardware automatically sets the ERR_VLD bit of the MCER2 register and the location information of the error for software to query. Software can clear the ERR_VLD bit by writing a 0, but cannot set it to 1. The bit length of this register is 64 bits, and the read/write permission of the register is machine-mode read/write, i.e., any non-machine mode access will result in an illegal instruction exception.

Offset:0x7C4 Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset

Bits	Access	Name	Description	Reset
63	R/W	ERR_VLD	L2 CACHE Checksum error bit. When ERR_VLD is 0, no abnormalities occur in L2 CACHE. When ERR_VLD is 1, there is an ECC error or a parity error in L2 CACHE. The software can clear this error bit in the exception service routine, but it cannot be set high.	0x0
62	R/W	ECC_FATAL	L2 CACHE Fatal error bit. When ECC_FATAL is 0, L2 CACHE does not have 2-bit ECC errors. When ECC_FATAL is 1, a 2-bit ECC error occurs in L2 CACHE.	0x0
61	R	RESERVED	RESERVED	0x0
60:56	R/W	FIX_CNT	Number of ECC Errors fixed. Records the number of ECC Errors that have been fixed.	0x0
55	R	RESERVED	RESERVED	0x0
54:53	R/W	RAMID	SRAM ID number where the ECC error occurred. Record the SRAM ID number where the ECC error occurred. id=0: L2 CACHE TAG RAM id=1: L2 CACHE DATA RAM id=2: L2 CACHE DIRTY RAM	0x0
52:25	R	RESERVED	RESERVED	0x0
24:21	R/W	ERR_WAY	L2 CACHE checksum error road location information: Records the road location of the first 2-bit checksum error in L2 CACHE.	0x0
20:3	R/W	ERR_INDEX	L2 CACHE checksum error index information. Record the index location of the first 2-bit checksum error in L2 CACHE.	0x0
2:0	R	RESERVED	RESERVED	0x0

MHINT

The Machine Mode Implicit Operation Register (MHINT) is used for cache multi-function switching control. The bit length of this register is 64 bits, and the read/write permission of the register is machine-mode read/write, i.e., any non-machine mode access will result in an illegal instruction exception.

Offset:0x7C5 Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset
63:24		RESERVED	RESERVED	0x0
23	R/W	FENCEI_BROAD_DIS	fence.i Operation broadcast cancellation bit. When FENCEI_BROAD_DIS is 0, the fence.i instruction operation is broadcast to other cores; when FENCEI_BROAD_DIS is 1, the fence.i instruction operation is not broadcast. The bit is not present in the single-core case.	0x0

Bits	Access	Name	Description	Reset
22	R/W	FENCERW_BROAD_DIS	fence Operation broadcast cancellation bit. When FENCERW_BROAD_DIS is 0, the fence instruction operation is broadcast to other cores; when FENCERW_BROAD_DIS is 1, the fence instruction operation is not broadcast. The bit is not present in the single-core case.	0x0
21	R/W	TLB_BROAD_DIS	TLB fence Operation broadcast cancellation bit. When TLB_BROAD_DIS is 0, the sfence.vma instruction operation is broadcast to other cores; when TLB_BROAD_DIS is 1, the sfence.vma instruction operation is not broadcast. The bit is not present in the single-core case.	0x0
20	R/W	L2STPLD	L2 Cache Store Prefetch enable bit. When L2STPLD is 0, L2 CACHE store prefetching is off; when L2STPLD is 1, L2 CACHE store prefetching is on.	0x0
19	R/W	ECC	L1 CACHE ECC checksum enable bit. When ECC is 0, L1Cache checksum is off; when ECC is 1, L1Cache checksum is on.	0x0
18	R/W	NO_SPEC	SPEC FAIL Predictive function enable bit. When NO_SPEC is 0, spec fail prediction is off; when NO_SPEC is 1, spec fail prediction is on.	0x0
17:16	R/W	L2_DIS	L2CACHE Number of prefetch cache lines. When L2_DIS is 0, 8 cache lines are prefetched; when L2_DIS is 1, 16 cache lines are prefetched; when L2_DIS is 2, 32 cache lines are prefetched; when L2_DIS is 3, 64 cache lines are prefetched. The L2Cache prefetching is based on the L1Cache prefetching.	0x0
15	R/W	L2PLD	L2CACHE Prefetch enable bit. When L2PLD is 0, L2CACHE prefetching is off; when L2PLD is 1, L2CACHE prefetching is on.	0x0
14:13		RESERVED	RESERVED	0x0
12	R/W	D_DIS	DCACHE Number of prefetched cache lines. When DPLD is 0, 2 cache lines are prefetched. When DPLD is 1, 4 cache lines are prefetched. When DPLD is 2, 8 cache lines are prefetched. When DPLD is 3, 16 cache lines are prefetched. The default is 0.	0x0
11	R/W	SRE	Single Retirement Mode Bits. When SRE is 0, Single Retirement mode is off; when SRE is 1, Single Retirement mode is on.	0x0
10	R/W	IWPE	ICACHE Road Prediction Enable bit. When IWPE is 0, ICACHE road prediction is off; when IWPE is 1, ICACHE road prediction is on.	0x0

Bits	Access	Name	Description	Reset
9	R/W	LPE	Cyclic acceleration enable bit. When LPE is 0, cycling is turned off. When LPE is 1, cyclic acceleration is on.	0x0
8		RESERVED	RESERVED	0x0
7:6	R/W	IPLD	ICACHE Prefetch enable bit. When IPLD is 0, ICACHE prefetching is off. When IPLD is 1, ICACHE prefetching is on.	0x0
5		RESERVED	RESERVED	0x0
4:3	R/W	AMR	L1 Cache Write Allocation Policy Auto-Adjust Enable bit. When the AMR is 0, the write allocation policy is determined by the page attribute WA of the access address. When the AMR is 1, the L1 Cache is no longer written to the subsequent consecutive addresses when there are 3 consecutive cache line storage operations. when the AMR is 2, the L1 Cache is no longer written to the subsequent consecutive addresses when there are 15 consecutive cache line storage operations. when the AMR is 3, the L1 Cache is no longer written to the subsequent consecutive addresses when there are 63 consecutive cache line storage operations.	0x0
2	R/W	DPLD	DCACHE Prefetch enable bit. When DPLD is 0, DCACHE prefetching is off; when DPLD is 1, DCACHE prefetching is on.	0x0
1:0		RESERVED	RESERVED	0x0

MRVBR

The Machine Mode Reset Register (MRVBR) is used to hold the base address of the reset exception vector. Each C908 core has a separate MRVBR register. The bit length of this register is 64 bits, and the read/write access to the register is machine mode read-only, meaning that any non-machine mode access will result in an illegal instruction exception.

Offset:0x7C7 Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset
63:10	R/W		Reset vector base-Reset base address. The reset base address of the control core.	0x0
9:0		RESERVED	RESERVED	0x0

MCER

The Machine Mode L1 Cache ECC Control Register (MCER) is used to configure the L1 Cache ECC. The first-level cache supports configurable ECC for 1-bit error correctability and 2-bit error detectability. When an error of 2 bits or more is detected, the hardware automatically sets the ERR_FATAL bit of the MCER register and the location information of the error for software query, and the software can clear the ERR_FATAL bit by writing a 0, but cannot set it to 1. The bit length of this register is 64 bits, and the read/write permission of the register is machine-mode read/write,

i.e., any non-machine mode access will result in an illegal instruction exception.

Offset:0x7C8 Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset
63	R/W	ECC_ERR	ECC message valid bit. When ERR_VLD is high, the L1 CACHE ECC message is invalid when ECC_ERR is 0. When ERR_VLD is high, the L1 CACHE ECC message is valid when ECC_ERR is 1.	0x0
62	R/W	BUS_ERR	Bus exception valid bit. When ERR_VLD is high, when BUS_ERR is 0, indicating that no ld bus exception has occurred. When ERR_VLD is high, when BUS_ERR is 1, it indicates an ld bus exception.	0x0
61:32		RESERVED	RESERVED	0x0
31	R/W	ERR_VLD	Asynchronous exception valid bit. When ERR_VLD is 0, it means that no bus exception or ECC err was generated. When ERR_VLD is 1, it means that a bus exception is generated or an ECC err occurs.	0x0
30	R/W	ERR_FATAL	L1 CACHE Checksum error bit. (a) When ERR_FATAL is 0, the hardware can repair the ERR. When ERR_FATAL is 1, it means that an ECC error of more than 2 bits has occurred and this bit can only be cleared by software.	0x0
29:24	R/W	FIX_CNT	Fixed ERROR Counting Bits.	0x0
23:21	R/W	RAMID	RAM with an ECC FATAL error. L1 ICACHE TAG RAM checksum error when RAMID is 0. L1 ICACHE DATA RAM checksum error when RAMID is 1. L1 DCACHE TAG RAM checksum error when RAMID is 2. L1 DCACHE DATA RAM checksum error when RAMID is 3. JTLB TAG RAM checksum error when RAMID is 4. When RAMID is 5, JTLB DATA RAM checksum error.	0x0
20:19		RESERVED	RESERVED	0x0
18:17	R/W	ERR_WAY	The location of the way when the ECC FATAL error first occurs. Records the location of the road when the first ECC FATAL ERROR occurs in the error RAM, and does not update this information until the first error is handled by the software and occurs again.	0x0
16:0	R/W	ERR_INDEX	Location of the index at which the first ECC FATAL error occurred. Records the index location of the first occurrence of an ECC FATAL ERROR in the error RAM, and does not update this information for further errors until the first error is handled by the software.	0x0

MCOUNTERWEN

The Machine Mode Reset Register (MRVBR) is used to hold the base address of the reset exception vector. Each C908 core has a separate MRVBR register. The bit length of this register is 64 bits, and the read/write access to the register is machine mode read-only, meaning that any non-machine mode access will result in an illegal instruction

exception.

Offset:0x7C9 Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset
63:31		RESERVED	RESERVED	0x0
31:3	R/W	HPM31~HPM3	When mcounterwen.bit[n] is 1, the corresponding shpmcounter is allowed to be written in superuser mode. When mcounterwen.bit[n] is 0, the corresponding shpmcounter is not allowed to be written in superuser mode, otherwise an illegal instruction exception is thrown.	0x0
2		IR		0x0
1		TW		0x0
0		CY		0x0

MCOUNTERINTEN

Machine mode event interrupt enable register (MCOUNTERINTEN), used to enable interrupt generation on overflow on each event count. The bit length of this register is 64 bits, and the read/write permission of the register is machine-mode read/write, i.e., any non-machine mode access will result in an illegal instruction exception.

Offset:0x7CA Total Reset Value:0x0000_0000_0000_0000 When mcounterinten.bit[n] is 1, the interrupt is triggered when the overflow on the corresponding mhpmcounter. When mcounterinten.bit[n] is 0, no interrupt is triggered when mhpmcounter overflows.

MCOUNTEROF

Machine mode event overflow annotation register (MCOUNTEROF) to indicate whether an overflow has occurred for each event count. The bit length of this register is 64 bits, and the read/write permission of the register is machine-mode read/write, i.e., any non-machine mode access will result in an illegal instruction exception.

Offset:0x7CB Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset
63:32		RESERVED	RESERVED	0x0
31:3		HPM30~HPM3	When mcounterof.bit[n] is 1, the corresponding mhpmcounter has overflowed. When mcounterof.bit[n] is 0, the corresponding mhpmcounter is not overflowed.	0x0
2		IR		0x0
1		TW		0x0
0		CY		0x0

MHPMSR/MHPMER

MHPMSR and MHPMER together perform the trigger monitoring task for performance monitoring. Both registers are 64 bits wide and hold the PC values of the start and stop counts respectively. The processor will use the virtual address as the match, or the physical address if the MMU is not enabled or is in M mode. When MHPMCR.TME is 2'b01, it is TRIGGER/STOP mode: the program runs to the PC specified by MHPMSR and the counter starts counting until the program runs to the termination trigger register (MHPMER) to end the count. When MHPMCR.TME is 2'b10, it is START/END mode: the counter triggers and counts when the program running PC is within the range of the start trigger register (MHPMSR) and the end trigger register (MHPMER), and all counters stop counting once the instruction retirement address is not in the range. Compared to TRIGGER/STOP mode, START/END mode has stricter conditions for counting, i.e., when a subroutine call occurs in a program segment that causes an instruction to run outside of

START and END, it will cause the counter to stop.

Offset:0x7F1/0x7F2 Total Reset Value:0x

MZONEID

Processor ZONE ID register.

Offset:0x7C7 Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset
63:4		RESERVED	RESERVED	0x0
3:0	R/W	ZoneID	The processor can configure the ZoneID number that it is currently in. Different ZoneIDs correspond to different permissions.	0x0

MLLCPID

Processor Last Level Cache partition ID register.

Offset:0x7F6 Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset
63:3		RESERVED	RESERVED	0x0
2:0	R/W	PARTID	processor last-level cache partition access ID. The field access is MRW, which determines the cache way group available to the current processor by matching the PARTID in the last level cache.	0x0

ML2WP

Processor ZONE ID register.

Offset:0x7F7 Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset
63:56		RESERVED	RESERVED	0x0
55:48	R/W	group3	group 3 has access to the ID. Contains 8 ids. When the bit corresponding to the processor MLLCPID number is 1, it means that this processor can access the cache way of group3.	0x0
47:40		RESERVED	RESERVED	0x0
39:32	R/W	group2	group 2 has access to the ID. contains 8 ids, when the bit corresponding to the processor MLLCPID number is 1, it means that this processor can access the cache way of group2.	0x0
31:24		RESERVED	RESERVED	0x0
23:16	R/W	group1	group 1 has access to the ID. Contains 8 ids. When the bit corresponding to the processor MLLCPID number is 1, it means that this processor can access the cache way of group1.	0x0
15:8		RESERVED	RESERVED	0x0

Bits	Access	Name	Description	Reset
7:0	R/W	group0	group 0 has access to the ID. Contains 8 ids. When the bit corresponding to the processor MLLCPID number is 1, it means that this processor can access the cache way of group0.	0x0

MCINS

The Machine Mode Cache Instruction Register (MCINS) is used to initiate read requests to the L1 or L2 cache. The bit length of this register is 64 bits, and the read/write permission of the register is machine-mode read/write, i.e., any non-machine mode access will result in an illegal instruction exception.

Offset:0x7D2 Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset
63:1		RESERVED	RESERVED	0x0
0		R	Cache Read Access. When R is 0, no Cache read request is initiated. When R is 1, a Cache read request is initiated.	0x0

MCINDEX

The Machine Mode Cache Access Index Register (MCINDEX) is used to configure the Cache location information accessed by read requests. The bit length of this register is 64 bits, and the read/write permission of the register is machine-mode read/write, i.e., any non-machine mode access will result in an illegal instruction exception.

Offset:0x7D3 Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset
63:32		RESERVED	RESERVED	0x0

Bits	Access	Name	Description	Reset
31:28		RID	RAM flag bit. Indicates the RAM information to be accessed. When the RID is 0, it means that the ICACHE TAG RAM is accessed. When the RID is 1, it means that the ICACHE DATA RAM is accessed. When the RID is 2, it means that the DCACHE TAG RAM is accessed. When the RID is 3, it means that the DCACHE DATA RAM is accessed. When RID is 4, it means that the L2CACHE TAG RAM is accessed. When RID is 5, it means that the L2CACHE DATA RAM is accessed. When RID is 6, it means that the L2CACHE ECC RAM is accessed. When the RID is 7, it means that the ICACHE DATA ECC RAM is accessed. When the RID is 8, it means that the DCACHE ST TAG ECC RAM is accessed. When the RID is 9, it means that the DCACHE DATA ECC RAM is accessed. When RID is 10, it means that the L2CACHE TAG ECC RAM is accessed. When RID is 11, it means that the L2CACHE DATA ECC RAM is accessed. When the RID is 12, it means that the DCACHE LD TAG RAM is accessed. When RID is 13, it means that the DCACHE LD TAG ECC RAM is accessed.	0x0
27:25		RESERVED	RESERVED	0x0
24:21		WAY	Cache Road Information. Indicates the road location information for RAM access.	0x0
20:0		INDEX	Cache Index. Indicates the index location information for RAM access.	0x0

MCDATA0/1

The Machine Mode Cache Data Register (MCDATA0/1) is used to record the data read from the L1 or L2 cache. The bit length of this register is 64 bits, and the read/write permission of the register is machine-mode read/write, i.e., any non-machine mode access will result in an illegal instruction exception.

Offset:0x7D4~0x7D5 Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset
63:0	R/W	DATA	Machine Mode Cache Data Register	0x0

MEICR

The Machine Mode L1Cache Hardware Error Injection Register (MEICR) is used to inject ECC errors into the L1cache. The bit length of this register is 64 bits, and the read/write permission of the register is machine-mode read/write, i.e., any non-machine mode access will result in an illegal instruction exception.

Offset:0x7D6 Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset
63:32		RESERVED	RESERVED	0x0

Bits	Access	Name	Description	Reset
31:29		RAMID	ECC RAM Index. When RAMID is 0, ICACHE TAG RAM is injected; when RAMID is 1, ICACHE DATA RAM is injected; when RAMID is 2, DCACHE TAG RAM is injected; when RAMID is 3, DCACHE DATA RAM is injected. When RAMID is 4, JTLB TAG RAM is injected; when RAMID is 5, JTLB DATA RAM is injected.	0x0
28:2		RESERVED	RESERVED	0x0
1		FATAL_INJ	ECC ERROR Injection selection bit. When FATAL_INJ is 1, 2bit error is injected; when FATAL_INJ is 0, 1bit error is injected.	0x0
0		INJ_EN	ECC error Inject enable bit. When INJ_EN is 1, L1cache ECC error injection is on; when INJ_EN is 0, L1cache ECC error injection is off.	0x0

MEICR2

The Machine Mode L2Cache Hardware Error Injection Register (MEICR2) is used to inject ECC errors into the L2cache. The bit length of this register is 64 bits, and the read/write permission of the register is machine-mode read/write, i.e., any non-machine mode access will result in an illegal instruction exception.

Offset:0x7D7 Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset
63:32		RESERVED	RESERVED	0x0
31:29	R/W	L2_RAMID	ECC RAM Index. When RAMID is 0, L2 CACHE TAG RAM is injected; when RAMID is 1, L2 CACHE DATA RAM is injected; when RAMID is 2, L2 CACHE DIRTY RAM is injected.	0x0
28:2		RESERVED	RESERVED	0x0
1	R/W	FATAL_INJ	ECC ERROR Injection selection bit. When FATAL_INJ is 1, 2bit error is injected; when FATAL_INJ is 0, 1bit error is injected.	0x0
0	R/W	L2_INJ_EN	L2 ECC ERROR Inject enable bit. When L2_INJ_EN is 1, L2Cache ECC error injection is on; when L2_INJ_EN is 0, L2Cache ECC error injection is off.	0x0

MBEADDR

L1 LD BUS ERR address register.

Offset:0x7C7 Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset
63:0		MBEADDR	L1 LD BUS ERR address register. This register stores the bus exception physical address when the processor has an ld bus exception.	0x0

MCPUID

The Machine Mode Processor Model Register (MCPUID) stores the processor model information, as defined in the C-

SKY Product ID Definition Specification (Version 4.0). Its reset value is determined by the product itself.

Offset:0xFC0 Total Reset Value:0914_0b0d

MAPBADDR

On-chip bus base address register. This register reflects the base address of the processor's on-chip registers (CLINT, PLIC). The value of this register is determined by hardware integration.

Offset:0xFC1 Total Reset Value:0x

MSMPR

The listen enable register controls whether the core can handle listen requests. Each core independently configures whether the core can handle listen requests, and the top-level consistency bus controls the sending of listen requests based on the listen status of each core. The read/write permission is machine mode read/write. The width of this register is 64 bits, only bit 0 is defined, the rest is Reserved.

Offset:0x7F3 Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset
63:1		RESERVED	RESERVED	0x0
0		SMPEN	core listen enable bit. When SMPEN is 1'b0, the core is not able to process listen requests and the top-level mask sends listen requests to the core. (Reset value) When SMPEN is 1'b1, the core is able to process listen requests and the top layer sends listen requests to the core	0x0

MHALTCAUSE

Genro debug reason register.

Offset:0xFE0 Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset
63:4		RESERVED	RESERVED	0x0
3:0	R/W	MHALTCAUSE	Indicates the reason for entering debug mode. When MHALTCAUSE is 1: indicates that the reason for entering debug mode is the execution of the ebreak instruction When MHALTCAUSE is 2: indicates that the reason for entering debug mode is trigger triggering When MHALTCAUSE is 3: indicates that the reason for entering debug mode is a synchronous debug request When MHALTCAUSE is 4: indicates that the reason for entering debug mode is a single-step debug request When MHALTCAUSE is 5: indicates that the reason for entering debug mode is to reset the debug request When MHALTCAUSE is 8: indicates that the reason for entering debug mode is an asynchronous debug request	0x0

MDBGINFO

Genro debug information register.

Offset:0xFE1 Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset

Bits	Access	Name	Description	Reset
63:0		MDBGINFO	Used to record debug information of processor cores during asynchronous debugging.	0x0

MPCFIFO

Genro branch target address record register.

Offset:0xFE2 Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset
63:0		MPCFIFO	Record the target address of the branch/jump instruction.	0x0

MDBGINFO2

Genro debug information register 2.

Offset:0xFE3 Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset
63:0		MDBGINFO2	Used to record debug information of CIU and L2-Cache during asynchronous debugging.	0x0

SHCR

The Super User Mode Hardware Control Register (SHCR) is a mapping of the Machine Mode Hardware Control Register (MHCR), refer to the Machine Mode Hardware Configuration Register (MHCR) for more information . The bit length of this register is 64 bits, and the read/write permission of the register is super user mode readable; user mode access will result in an illegal instruction exception.

Offset:0x5C1 Total Reset Value:108

SCER2

The Super User Mode L2Cache ECC register (SCER2) is a mapping of the Machine Mode L2Cache ECC register (MCER2), please refer to the Machine Mode L2 Cache ECC Control Register (MCER2) for more information. The bit length of this register is 64 bits, and the read/write permission of the register is super user mode readable; user mode access will result in an illegal instruction exception.

Offset:0x5C2 Total Reset Value:0x0

SCER

The Super User Mode L1Cache ECC register (SCER) is a mapping of the Machine Mode L1Cache ECC register (MCER), refer to the Machine Mode L1 Cache ECC register (MCER) for more information. The bit length of this register is 64 bits, and the read/write permission of the register is super user mode readable; user mode access will result in an illegal instruction exception.

Offset:0x5C3 Total Reset Value:0x0

SCOUNTERINTEN

The Super User Mode Overflow Interrupt Enable Register (SCOUNTERINTEN) is an event overflow interrupt enable register (MCOUNTER-) in machine mode. (INTEN) is mapped, please refer to the Machine Mode Event Interrupt Enable Register (MCOUNTERINTEN) for more information. The bit length of this register is 64 bits, and the read/write permission of the register is super user-mode readable; user-mode access will result in an illegal instruction exception. When mcounterwen.bit[n] is 1, scounterinten.bit[n] determines whether an interrupt is generated when the corresponding shpmcounter overflows.

Offset:0x5C4 Total Reset Value:0x0

SCOUNTEROF

The Super User Mode Event Overflow Labeling Register (SCOUNTEROF) is a mapping of the Machine Mode Event

Overflow Labeling Register (MCOUNTEROF), refer to the Machine Mode Event Overflow Labeling Register (MCOUNTEROF) for more information.

Offset:0x5C5 Total Reset Value:0x

SHPMCR

Super User Performance Monitoring Control Register.

Offset:0x5C9 Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset
63	R	TS		0x0
62:14		RESERVED	RESERVED	0x0
13	R	PMDM	Machine mode performance monitoring count enable bit, mapped for the mxstatus.pmdm bit. When PMDM is 0, the performance counter is allowed to count in machine mode. When PMDM is 1, performance counter counting is disabled in machine mode.	0x0
12		RESERVED	RESERVED	0x0
11		PMDS		0x0
10		PMDU		0x0
9:2		RESERVED	RESERVED	0x0
1:0		TIME		0x0

SHPMSR/SHPMER

SHPMSR/SHPMER is the mapping of MHPMSR/MHPMER, when MHPMCR.sce is high, super user mode can control the start and stop of triggering through this set of registers. The processor will use the virtual address as the match, or the physical address if the MMU is not enabled or is in M mode.

Offset:0x5C9~0x5CA Total Reset Value:0x

SLLCPID

When mxstatus.bit[SPCE] is 1, the superuser state can configure the partition access id number of the L2 cache by configuring SLLCPID.

Offset:0x5CC Total Reset Value:0x

SL2WP

When mxstatus.bit[SPCE] is 1, the superuser state can configure partition access to the L2 cache by configuring SL2WP.

Offset:0x5CD Total Reset Value:0x

SBEADDR

The field definition and function of this register are the same as MBEADDR.

Offset:0x5CE Total Reset Value:0x0

SCYCLE

The Super User Mode Cycle Counter (SCYCLE) is used to store the number of cycles the processor has executed.

When the processor is in an execution state (i.e., not in a low-power state), the SCYCLE register is self-incrementing at each execution cycle. The cycle counter is 64-bit, and the cycle counter is reset to zero. Please refer to Event Counter for more information.

Offset:0x5E0 Total Reset Value:0x0000_0000_0000_0000

SHPMCOUNTER3~SHPMCOUNTER31

The Super User Mode Event Counter (SHPMCOUNTERn) is a mapping of the Machine Mode Event Counter (MHPMCOUNTERn).

Offset:0x5FF Total Reset Value:0x0 SHPMCOUNTERn is the corresponding machine mode Mapping of event counters.

SMIR

The bit length of this register is 64 bits and the read/write permission of the register is super user mode readable, user mode access will result in an illegal instruction exception.

Offset:0x9C0 Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset
63:32		RESERVED	RESERVED	0x0
31	R/W	P	Probe Failure. 0: TLBP query matching hit. 1: The TLBP query failed.	0x0
30	R/W	Tfatal	Probe multiple. Whether multiple matches occur when the TLBP instruction is executed. 0: No multiple matches. 1: Multiple matches occurred.	0x0
29:16		RESERVED	RESERVED	0x0
15	R/W	Iutlb	Instruction Micro TLB. Represents a query instruction uTLB.	0x0
14	R/W	Dutlb	Data Micro TLB. Represents a query instruction dTLB.	0x0
13:10		RESERVED	RESERVED	0x0
9:0	R/W	Index	TLB Index. 512 entry configuration: Index [8:7] is the way index, and Index [6:0] is the set/entry index; (4 way , 128 entries) 1024 entry configuration: Index [9:8] is the way index, and Index [7:0] is the set/entry index; (4 way , 256 entries)	0x0

SMEL

The bit length of this register is 64 bits and the read/write permission of the register is super user mode readable, user mode access will cause illegal instruction exception.

Offset:0x9C1 Total Reset Value:0x

Bits	Access	Name	Description	Reset
63	R/W	SO	The main content of SMEL is the same as that of page table. See page table format for relevant bit definitions.	
62	R/W	C	The main content of SMEL is the same as that of page table. See page table format for relevant bit definitions.	
61	R/W	B	The main content of SMEL is the same as that of page table. See page table format for relevant bit definitions.	

Bits	Access	Name	Description	Reset
60	R/W	SH	The main content of SMEL is the same as that of page table. See page table format for relevant bit definitions.	
59	R/W	SEC	The main content of SMEL is the same as that of page table. See page table format for relevant bit definitions.	
58		RESERVED	RESERVED	
57:55	R/W	PMP	PMP	
54:38		RESERVED	RESERVED	
37:10	R/W	PPN	PPN	
9:8	R/W	RSW	RSW	
7	R/W	D	D	
6	R/W	A	A	
5	R/W	G	G	
4	R/W	U	U	
3	R/W	X	X	
2	R/W	W	W	
1	R/W	R	R	
0	R/W	V	V	

SMEH

The bit length of this register is 64 bits and the read/write permission of the register is super user mode readable, user mode access will cause illegal instruction exception. The SMEH register is used to save the virtual address information accessed by TLB. The hardware is updated when the TLB is read, and the software writes the TLB table entry in advance write in.

Offset:0x9C2 Total Reset Value:0x

Bits	Access	Name	Description	Reset
63	R/W	512G	Under SV48, it indicates that the page size is 512G.	
62:59	R/W	ZID	In the TEE configuration, it represents the ZoneID corresponding to the page. This field is updated by the hardware when the TLB is read, and written by the software before the TLB table entry is written by the software.	
58:46/55		RESERVED	RESERVED	
45/54:19	R/W	VPN	Virtual page frame number. This field is updated by the hardware during TLB reading, and is pre written by the software before the software writes TLB table entries.	
18:16	R/W	PageSize	Page size, Onehot from low to high means 4K, 2M, 1G page size. The hardware of this field is updated during TLB reading, and it is pre written by the software before the software writes TLB table entries.	
15:0	R/W	ASID	Address space identification.	

SMCIR

The bit length of this register is 64 bits and the read/write permission of the register is super user mode readable, user mode access will cause illegal instruction exception. For more information, please refer to virtual_mem_manage_smcir. Offset:0x9C3 Total Reset Value:0x

Bits	Access	Name	Description	Reset
63:32		RESERVED	RESERVED	
31	R/W	TLBP	TLB Query. Query the TLB according to the EntryHi register. When the query hits, update the Index register with the serial number of the TLB.	
30	R/W	TLBR	TLB Read. Read the values of the corresponding TLB table entries according to the Index register index, and use these values to update the SMEH and SMEL registers.	
29	R/W	TLBWI	TLB index write. Write the registers SMEH and SMEL into the corresponding TLB table entries according to the index value of the Index register.	
28	R/W	TLBWR	TLB random write. Write the registers SMEH and SMEL into the TLB corresponding table entries according to the index value of Random register.	
27	R/W	TLBIASID	TLB is invalid according to ASID. All TLB table entries matching the ASID are invalid.	
26	R/W	TLBIALL	TLB initialization. Invalidate all TLB table entries and initialize them.	
25	R/W	TLBII	The TLB is invalid according to the index. According to the index value of the Index register, the corresponding TLB table entry is invalid.	
24:16		RESERVED	RESERVED	
15:0	R/W	TLBASID	ASID. The TLBIASID operation uses this ASID for matching.	

FXCR

The user mode floating point expansion control register (FXCR) is used for the floating point expansion function switch and the floating point exception accumulation bit.

Offset:0x9C3 Total Reset Value:0x0000_0000_0000_0000

Bits	Access	Name	Description	Reset
63:27		RESERVED	RESERVED	0x0
26:24	R/W	RM	rounding mode. The mapping of the FCSR to the corresponding bits.	0x0

Bits	Access	Name	Description	Reset
23	R/W	DQNaN	Output QNaN mode bits. When DQNaN is 0, the output QNaN value is calculated as the default fixed value. When DQNaN is 1, the calculated output QNaN value is consistent with the IEEE754 standard.	0x0
22:6		RESERVED	RESERVED	0x0
5	R/W	FE	Floating point anomaly accumulation bit. This bit will be set to 1 when any of the floating point exceptions occur.	0x0
4	R/W	NV	Invalid operand exception. The mapping of the FCSR to the corresponding bits.	0x0
3	R/W	DZ	except 0 Exception. The mapping of the FCSR to the corresponding bits.	0x0
2	R/W	OF	overflow anomaly. The mapping of the FCSR to the corresponding bits.	0x0
1	R/W	UF	underflow anomaly. The mapping of the FCSR to the corresponding bits.	0x0
0	R/W	NX	Non-precision anomalies. The mapping of the FCSR to the corresponding bits.	0x0

3.1.4.3 CLIC Register

MSIPO

Machine mode software interrupt waiting bit.

Offset:0x4000000 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:1		RESERVED	RESERVED	0x0
0	R/W	MSIP	Machine mode software interrupt waiting bit. This bit indicates the interrupt status of the machine mode software interrupt. When the MSIP is at position 1, there is currently a valid machine mode software interrupt request. When the MSIP is at 0, there is currently no valid machine mode software interrupt request.	0x0

MSIPN

Machine mode software interrupt waiting bit.

Offset:0x4000000+4*n Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:1		RESERVED	RESERVED	0x0

Bits	Access	Name	Description	Reset
0	R/W	MSIP	Machine mode software interrupt waiting bit. This bit indicates the interrupt status of the machine mode software interrupt. When the MSIP is at position 1, there is currently a valid machine mode software interrupt request. When the MSIP is at 0, there is currently no valid machine mode software interrupt request.	0x0

MTIMECMPL0

Machine Mode Timer Interrupt Comparison Value Register (Low).

Offset:0x4004000 Total Reset Value:0xffff_ffff

MTIMECMP0

Machine Mode Timer Interrupt Comparison Value Register (High).

Offset:0x4004004 Total Reset Value:0xffff_ffff

Bits	Access	Name	Description	Reset
63:32	R/W	MTIMECMPH	Timer comparison value high 32 bits	0xffff
31:0	R/W	MTIMECMPL	Timer comparison value low 32 bits	0xffff

MTIME

Machine Mode Clock Timer(low).

Offset:0x400bff8 Total Reset Value:0x0000_0000

MTIMEH

Machine Mode Clock Timer(High).

Offset:0x400bffc Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
63:32	R/W	MTIMEH	Clock timer high 32 bits	0x0
31:0	R/W	MTIMEL	Clock timer low 32 bits	0x0

SSIP0

Core 0 super user mode software interruption.

Offset:0x400c000 Total Reset Value:0x0000_0000 Refer to machine mode for super user mode.

3.1.4.4 PLIC Register

PLIC_PRI01

Interrupt source 1 priority configuration register.

Offset:0x0000004 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:5		RESERVED	RESERVED	0x0

Bits	Access	Name	Description	Reset
4:0	R/W	PRIO	The lower 5 bits of the priority configuration register are writable and support 32 priorities of different levels. The priority setting is 0, which means the interrupt is invalid. The priority of machine mode interrupt is unconditionally higher than that of superuser mode interrupt. When the modes are the same, priority 1 is the lowest priority and priority 31 is the highest. When the priorities are the same, further compare the interrupt source IDs. The ones with smaller IDs have higher priorities.	0x0

PLIC_IP0

Interruption No. 1~31Interrupt Wait Register.

Offset:0x0001000 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31	R/W	IP ID(32x+31)	This bit indicates the interrupt waiting status of the corresponding interrupt source. – When the IP bit is 1, it indicates that the external interrupt source currently has interrupts waiting for response. This bit can be set to 1 by memory storage instruction. This position 1 will also be set after the sampling logic of the corresponding interrupt source samples to the effective level or pulse interrupt. When the IP bit is 0, it indicates that the external interrupt source is not waiting for a response interrupt. This bit can be cleared to 0 by memory storage instruction. When the interrupt is responded, PLIC will clear the corresponding IP bit.	0x0
30	R/W	IP ID(32x+30)	Same as the first line	0x0
29:2	R/W	...	...	0x0
1	R/W	IP ID(32x+1)	Same as the first line	0x0
0	R/W	IP ID(32x)	Same as the first line	0x0

PLIC_H0_MIE0

Core 0 1~31 machine mode interrupt enable register.

Offset:0x0002000 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31	R/W	IE ID(32x+31)	This bit indicates the interrupt enable status of the corresponding interrupt source. When the IE bit is 1, it indicates that the interrupt is enabled for the target. When the IE bit is 0, it indicates that the interrupt shields the target.	0x0
30	R/W	IE ID(32x+30)	Same as the first line	0x0
29:2	R/W	...	...	0x0

Bits	Access	Name	Description	Reset
1	R/W	IE ID(32x+1)	Same as the first line	0x0
0	R/W	IE ID(32x)	Same as the first line	0x0

PLIC_CTRL

PLIC authority control register.

Offset:0x01FFFFC Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:1		RES	RES	0x0
0	R/W	S_PER	Access control bit: At S_When PER is 0, only machine mode has access to all the registers of PLIC. Super user mode has no access to PLIC permission control register, interrupt priority configuration register, interrupt waiting register and interrupt enable register. Only super user mode interrupt threshold register and super user mode interrupt response/completion register can be accessed. Normal user mode does not have access to any PLIC registers. At S_When PER is 1, the machine mode has all permissions. Super user mode has all the PLIC register permissions except for the PLIC permission control register. Normal user mode does not have access to any PLIC registers.	0x0

PLIC_H0_MTH

Core 0 machine mode interrupt threshold register.

Offset:0x0200000 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:5		RES	RES	0x0
4:0	R/W	PRIOTHRESHOLD	Priority threshold: Indicates the interrupt threshold for the current interrupt target. The threshold value is set to 0, indicating that all interrupts are allowed.	0x0

PLIC_H0_MCL

Core 0 machine mode interrupt response/completion register.

Offset:0x0200004 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:10		RES	RES	0x0

Bits	Access	Name	Description	Reset
9:0	R/W	CLAIM_ID	Interrupt request ID: Reading the register: returns the ID value currently stored in the register. The read operation indicates that the interrupt of the corresponding ID has started processing. PLIC starts interrupt response processing. Write operation on this register: indicates that the interrupt corresponding to the ID of the write value has been processed, and this write operation will not update the interrupt response/completion register. PLIC starts interrupt completion processing.	0x0

PLIC_H0_STHL

Core 0 superuser mode interrupt threshold register.

Offset:0x0201000 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:5		RES	RES	0x0
4:0	R/W	PRIOTHRESHOLD	Priority threshold: Indicates the interrupt threshold for the current interrupt target. The threshold value is set to 0, indicating that all interrupts are allowed.	0x0

PLIC_H0_SCLA

Core 0 super user mode interrupt response/completion register.

Offset:0x0201004 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:10		RES	RES	0x0
9:0	R/W	CLAIM_ID	Interrupt request ID: Reading the register: returns the ID value currently stored in the register. The read operation indicates that the interrupt of the corresponding ID has started processing. PLIC starts interrupt response processing. Write operation on this register: indicates that the interrupt corresponding to the ID of the write value has been processed, and this write operation will not update the interrupt response/completion register. PLIC starts interrupt completion processing.	0x0

3.2 cpu1 subsystem

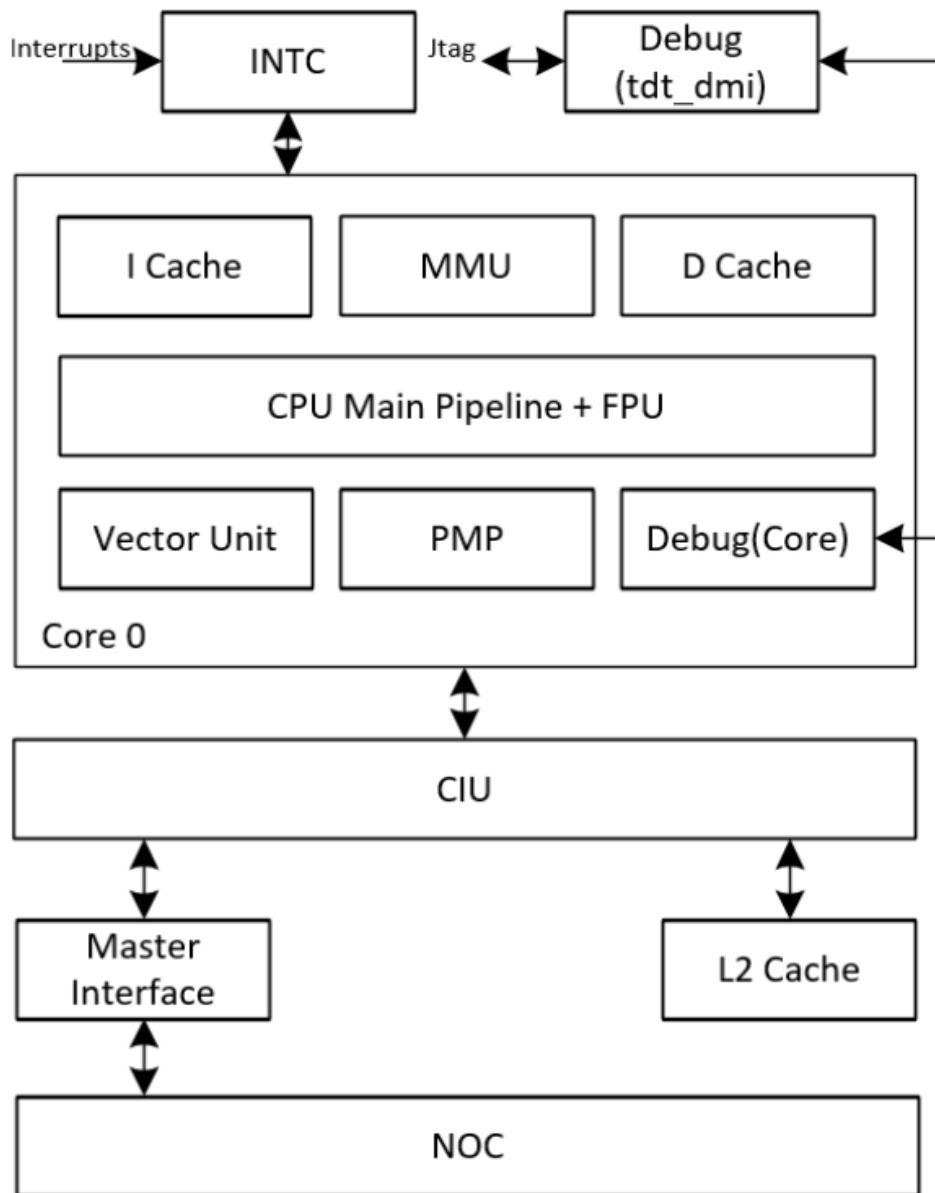
3.2.1 Overview

cpu1_subsystem is used to AI services, Multimedia services and run RT-SMART.

3.2.2 Function Description

3.2.2.1 Logical Block Diagram

Figure 3-2-1 Logical block diagram of the cpu1 subsystem



3.2.2.2 Features

The cpu1_subsystem uses the T-head C908 processor, which has the following features: 1.Supports maximum 1.6 GHz working frequency. 2.Cpu contains floating-point units (FPUs),128bits vpu with RVV1.0. 3.L1 cache contains 32 KB instruction cache (I-cache) and 32 KB data cache (D-cache). 4.Integrates with 256 KB L2 cache. 5.Supports memory management unit (MMU). 6.Integrates with the interrupt controller (PLIC), which can process 208 interrupt external sources. 7.Supports the Joint Test Action Group (JTAG) debugging interface. 8.Supports WFI. 9.Supports T-head TEE(T-head Security extension).

3.2.3 Register Summary

Table 3-2-1 Summary of cpu1_sys registers The registers of cpu1_sys are the same as those of cpu0_sys.

3.2.4 Register Description

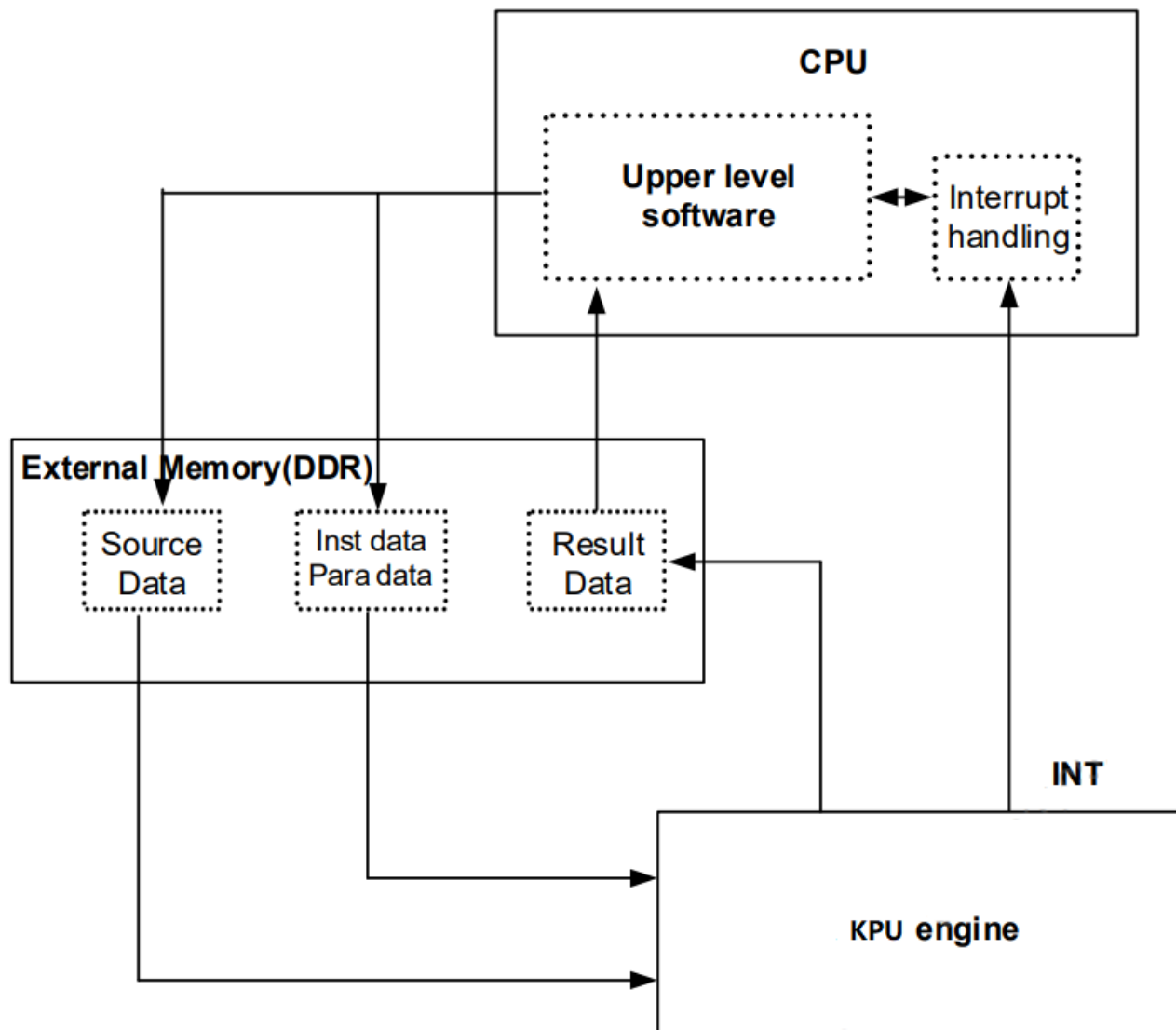
The registers of cpu1_sys are the same as those of cpu0_sys.

4 KPU

4.1 Overview

Based on the structure of neural network, KPU is a special acceleration engine for deep learning. Supports various basic operators of neural networks, which can be used in scenarios such as object detection, image classification, and liveness detection. Figure 4-1 shows the location of the KPU in the system.

Figure 4-1-1 The location of the KPU in the system



4.2 Function Description

- Support weight decompression (compression is done offline by compiler)
- Support convolution operator data precision INT8/INT16
- Support some non-convolutional operators to support fp16
- Support ordinary convolution (sliding window maximum width 8, unlimited height)
- Support group convolution, dw convolution
- Support dilation convolution, deconvolution
- Support matrix multiplication
- Support FC
- Support MatMul
- Support Shuffle_channel
- Support pooling (max, min, avg, add operations)
- Support LSTM, GRU

- Support Transpose
- Support Memoryset/Memorycopy
- Support Upsample
- Support Map/Broadcast
- Support REDUCE (max, min, add, sub, mult operations)
- Support batch normalization (BN)
- Support ReLU, ReLU6, LeakyReLU, PReLU, Swish/HardSwish, Mish, sigmoid
- Support Concat
- Support stride slice
- Support slice padding
- Support Reshape, Convert
- Support Cast
- Support Quant/Dequant
- Support SpaceToBatch/BatchToSpace
- Support Permute

NOTE

- Note1: Other operators such as element-wise, softmax, and Gather are calculated by CPU software and can be accelerated by SIMD.
- Note2: Crop, resize, and padding can be completed by the AI-2D acceleration engine.

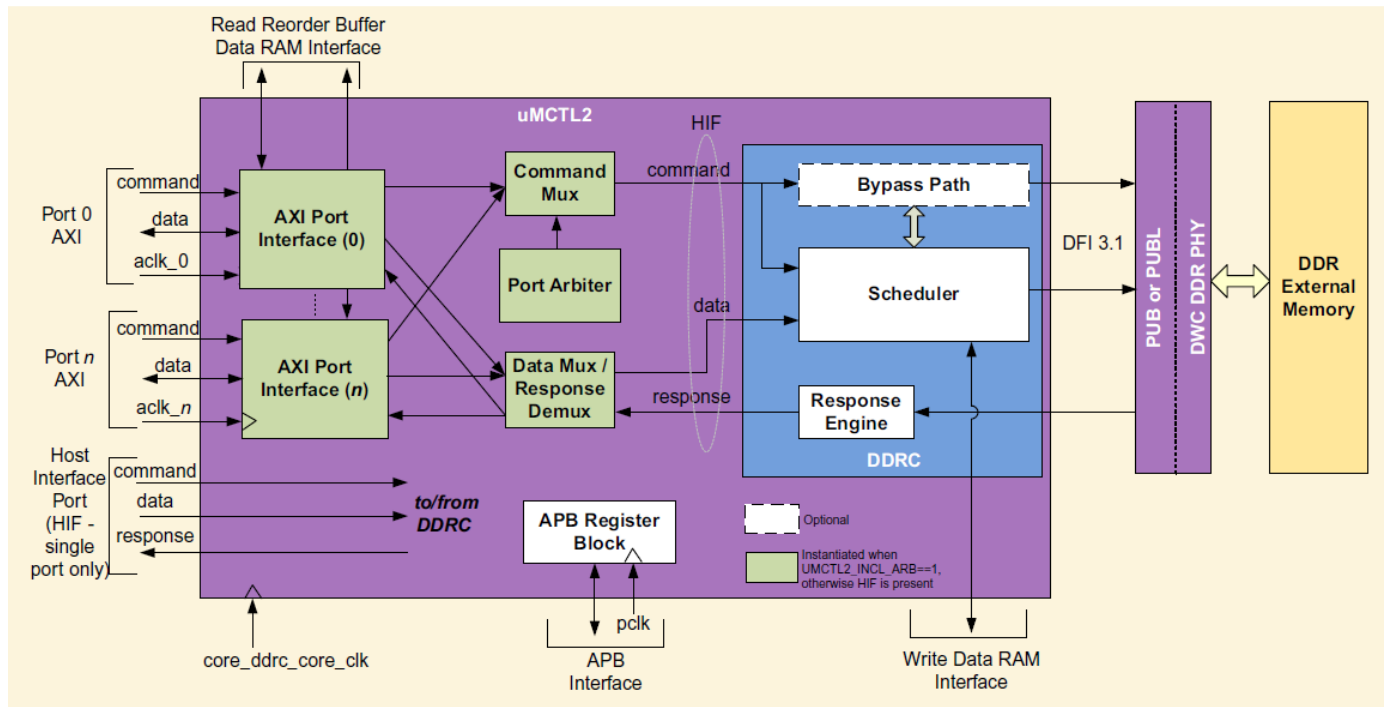
5 Memory Interfaces

5.1 DDRC

5.1.1 Overview

The DDRC support LPDDR3 and LPDDR4 SDRAMs, the controller is a high performance memory controller including scheduler and port arbiter, the PHY which embedded microprocessor based training is mainly phy independent, and the PUB in the phy is mainly for training, BIST, configuration registers, etc. The system block diagram as below.

Figure 5-1-1 ddr system diagram



For the PHY, it have below advantages:

- Compatible with JEDEC standard LPDDR4 or LPDDR3 SDRAM
- PHY independent, embedded microprocessor based training
- mbedded DFT and 2D eye-mapping diagnostic capability
- Up to 4 trained states / frequencies —<5us switching time
- Delay lines for DQS centering, write leveling, per bit deskew
- PHY Utility Block (soft IP component) for training, BIST, configuration registers, etc.
- I/O cells embedded in the hard IPs

5.1.2 Features

- 16-bit per channel LPDDR4, two channels
- 32-bit per channel LPDDR3 supported
- DDR PHY Interface (DFI) support for easy integration with industry standard DFI 3.1-compliant
- Scalable 1:1/1:2 frequency ratio architecture. We setting 1:2 frequency ratio.
- For LPDDR3 configurations, direct software request control or programmable internal control for ZQ short calibration cycles
- For LPDDR3 configurations, support for ZQ long calibration after self-refresh exit
- For LPDDR4 configurations, direct software request control or programmable internal control for ZQ calibration cycles
- For LPDDR4 configurations, support for ZQ calibration after SR-Powerdown exit
- Dynamic scheduling to optimize bandwidth and latency
- For maximum SDRAM efficiency, commands are executed out-of-order:
- Hardware configurable and software programmable Quality of Service (QoS) support:
- Configurable maximum SDRAM data-bus width (denoted as “full data-bus width”)
- Programmable support for all of the following SDRAM data-bus widths: Full data-bus width or Half of the full data-bus width
- Supports max to 2 ranks
- Control options to avoid starvation of lower priorities
- Guaranteed coherency for write-after-read (WAR) and read-after-write (RAW) hazards (always on the HIF interface and on the AXI interface only if appropriate hardware configuration parameter and software register are

set)

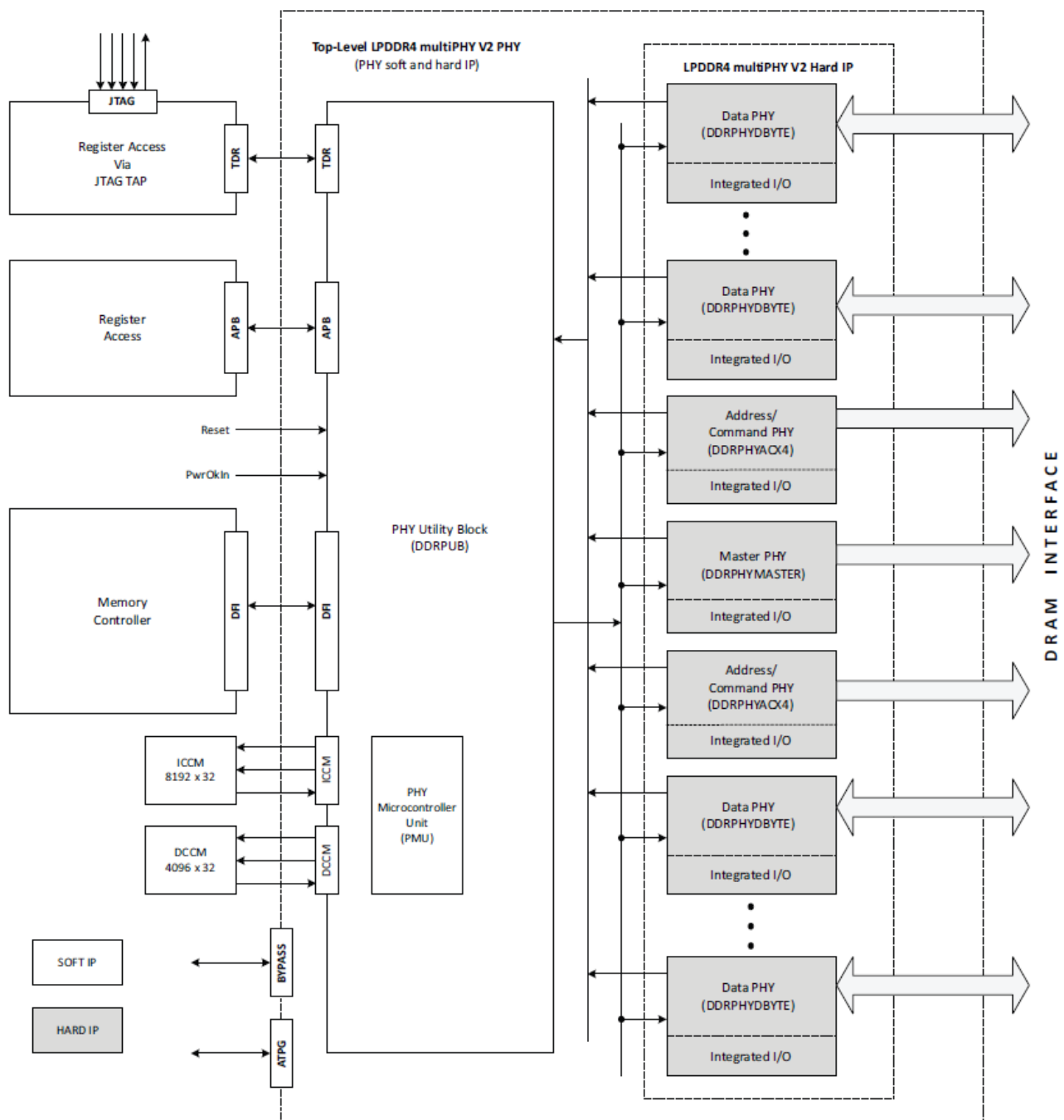
- Write combine to allow multiple writes to the same address to be combined into a single write to SDRAM; supported for same starting address
- Supports automatic SDRAM power-down entry and exit caused by lack of transaction arrival for a programmable time
- Supports automatic Clock Stop (LPDDR3/LPDDR4) entry and exit caused by lack transaction arrival
- Supports automatic uMCTL2 low power mode operation caused by lack of transaction arrival for a rogrammable time through the Hardware Low Power Interface
- Support for dynamically changing clock frequency while in self-refresh
- Shadow timing registers provided to allow fast frequency changing
- Flexible address mapper logic to allow application specific mapping of row, column, bank, and rank bits
- Leverages out of order requests with CAM to maximize throughput
- APB interface for the uMCTL2 software accessible registers
- 5 host ports using AMBA AXI
- Automatic periodic retraining through PHY MASTER Interface
- Support for:
 1. (LPDDR3, LPDDR4) Command Bus eye training relative to CK
 2. Write Leveling to compensate for CK-DQS timing skew
 3. Write Training: DQS to DQ Two modes of training are available in LPDDR4: - Command-based FIFO WR/RD with user patterns - An internal DQS clock tree oscillator, to determine the need for, and the magnitude of, required periodic training
 4. Data bus VREFDQ training
 5. Read training(DQ bit deskew training using DRAM mode registers and DQS to DQ eye centering training using DRAM array)
- LVSTL IO calibration and ODT calibration
- Support for a SW controllable DQ bit and AC bit swizzling
- Test support:
 1. At-speed loopback testing on both the address and data channels
 2. Delay line BIST
 3. MUX-scan ATPG (stuck-at SCAN)
 4. RX and TX asynchronous paths support a maximum bit rate of 400 Mb/s (200MHz)
 5. PLL lock test
 6. ZQ calibration test
- JTAG register interface for easy test access
- Firmware-based 2D eye mapping diagnostic tool allows measuring 2D eye for every bit of the bus at both DRAM and host receivers
- Direct override programming available for all VREF, ODT, drive strength, and timing delays to facilitate debug and characterization

5.1.3 Function description

5.1.3.1 block diagram

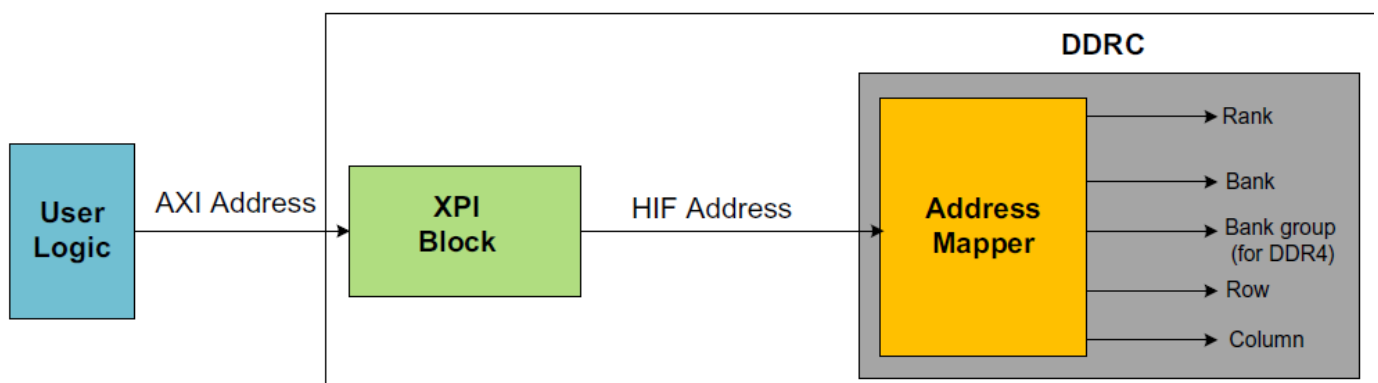
the ddr solution is a combination of hard and soft IP that is used to provide a physical interface to JEDEC standard LPDDR3/LPDDR4 SDRAM memories. This solution provides logical support for all JEDEC DRAM devices compatible with the system design and PHY hard macro components. The following figure shows how the solution is used in a typical SDRAM system.

Figure 5-1-2 system overview



5.1.3.2 address mapping

the ddrc converts this system address to a physical address. It maps the system address to the SDRAM rank, bank, row, and column addresses,



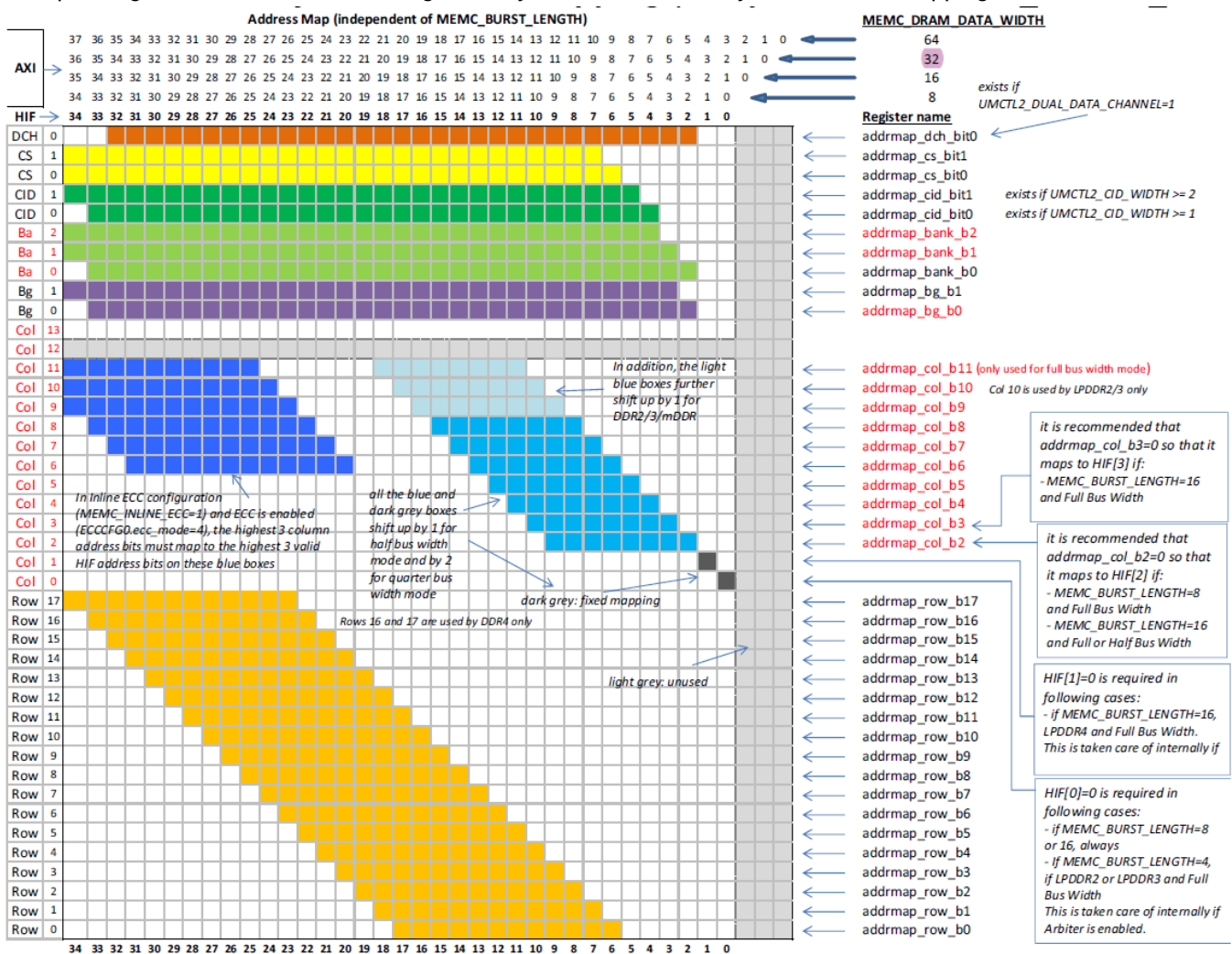
The address mapper maps HIF word addresses to SDRAM addresses by selecting the HIF address bit that maps to each and every applicable SDRAM address bit. While it is possible to map HIF address bits to a SDRAM address in any desired manner, the full available address space is accessible only when no two SDRAM address bits are determined by the same HIF address bit. Each SDRAM address bit has an associated register vector to determine its source. Registers ADDRMAPx (x = 0 to 11) are used to program the address mapper. The HIF address bit number is determined by adding the internal base of the ADDRMAPx (x = 0 to 11) register to the programmed value for that register, as described in the following equation: $\text{HIF address bit number} = [\text{internal base}] + [\text{register value}]$

For example, for ADDRMAP3.addrmap_col_b7, the internal base is 7. When full data bus is in use, column bit 7 is determined by the following equation: $7 + [\text{register value}]$

If this register is programmed to 2, the HIF address bit can be calculated by using the following equation:

$[\text{HIF address bit number}] = 7 + 2 = 9$

In other words, the column address bit 7 sent to SDRAM would always be equal to hif_cmd_addr[9] of the corresponding HIF source address. the figure of System Address to Physical Address Mapping



Note:

- All of the column bits shift up 1 bit when only half of the data bus is in use. In this case, you need to look at ADDRMAP3.addrmap_col_b6 instead to determine the value of column address bit 7.
- All of the column bits shift up 2 bits when only a quarter of the data bus is in use. In this case, you need to look at ADDRMAP2.addrmap_col_b5 instead to determine the value of column address bit 7.
- The addressing for column bits is independent of MEMC_BURST_LENGTH. The column bits do not shift up by an additional 1 bit when uMCTL2 is configured for BL8 mode (MEMC_BURST_LENGTH = 8). Similarly, the column bits do not shift up by an additional 2 bits when the uMCTL2 is configured for BL16 mode (MEMC_BURST_LENGTH = 16).

- For Arbiter configurations (UMCTL2_INCL_ARB = 1) with MEMC_BURST_LENGTH = 16, Full bus width (MSTR.data_bus_width = 00) and BL16 (MSTR.burst_rdw = 1000), it is recommended to program ADDRMAP2.addrmap_col_b3 = 0.
- For Arbiter configurations (UMCTL2_INCL_ARB = 1) with MEMC_BURST_LENGTH = 16, it is required to program ADDRMAP2.addrmap_col_b2 = 0
- For Arbiter configurations (UMCTL2_INCL_ARB = 1) with MEMC_BURST_LENGTH = 8, it is required to program ADDRMAP2.addrmap_col_b2 = 0
- If LPDDR2 or LPDDR3, SDRAM column address [0] does not exist and hence, is not addressable. If UMCTL2_INCL_ARB = 1, this is automatically the case. Otherwise, the HIF master must generate hif_cmd_addr[0] = 0 if MEMC_BURST_LENGTH = 4 and Full Bus Width
- In LPDDR4, SDRAM column address [1:0] does not exist and hence is not addressable. If UMCTL2_INCL_ARB = 1, this is automatically the case. Otherwise, if MEMC_BURST_LENGTH = 16 and Full Bus Width, the HIF master must generate hif_cmd_addr[1:0] = 00 for Reads. For Writes or RMW, hif_cmd_addr[1] = 1 is allowed but hif_cmd_addr[0] = 0 is required.
- The register ADDRMAP5.addrmap_row_b2_10 can map multiple HIF address bits (if the value set is lesser than 15). If set to 15, then ADDRMAP9, ADDRMAP10, and ADDRMAP11 registers are used, which provide individual mapping for the HIF address bits (rows 2 to 10).
- For any address bits which cannot be in use in all cases, all bits of the associated address map register must be set to 1 when the associated SDRAM address bit is not in use.

5.1.3.3 arbitration mechanism

The Port Arbiter (PA) block arbitrates command requests from up to 16 AXI/AHB ports to the HIF of the DDR Controller (DDRC). PA is comprised of multiple tiers of arbitration stages which include:

- Read/write arbitration (when UMCTL2_DUAL_HIF is set to '0')
- 2-priority level arbitration based on port aging and expired-VPR/expired-VPW commands (timeout priority0)
- 2-priority level arbitration for read requests based on DDRC read priorities (HPR/LPR-VPR)
- 32-priority level arbitration based on internal port aging or 16-priority level arbitration based external AXI QoS inputs (selectable by hardware parameter)
- Round-robin arbitration to resolve ports having the same priority after passing all stages of arbitration

5.1.3.3.1 Read/Write Arbitration

This is the first level of arbitration where all requests to the ports are observed and a decision is made based on:

- Read (LPR-VPR, HPR) and write credits
- Timeout/Expired-VPR/Expired-VPW (Priority0) The main goal of the read/write arbiter is to combine reads and writes together as long as the selected direction has available credits and the timeout has not occurred for any port of the opposite direction. If all conditions are equal, reads are prioritized over writes. Maximize direction switches to improve the memory bus efficiency. The algorithm can be summarized as:
 - While executing reads:
 - ☐ Stay on the reads as long as there is a timed-out read port or expired-VPR (which are both Priority0 conditions) with available credit, else
 - ☐ Switch to writes if there is a timed-out write port or expired-VPW (Priority0) with available credit, else ☐ Switch to writes when there is no read credit left and there is a pending write with available credit.
 - While executing writes:
 - ☐ Stay on the writes as long as there is a timed-out write port or expired-VPW (Priority0) with available credit, else ☐ Switch to reads if there is a timed-out read port or expired-VPR (which are both Priority0 conditions) with available credit, else
 - ☐ Switch to reads if there is an HPR read port with available credit, else-
 - ☐ Switch to reads when there is no write credit left and there is a pending read with available credit.

5.1.3.3.2 Port timeout

The aging counters per port, per direction (read/write) count down the time when a port is requesting but not granted. The timeout condition occurs when a port aging counter becomes 0 and the port becomes the highest priority requester (Priority0) to the Port Arbiter. Registers PCFGR_n.rd_port_priority and PCFGW_n.wr_port_priority (see UMCTL2_MP Registers page 1170) determine the initial value of the counters. The aging feature and the timeout can be enabled by PCFGR_n.rd_port_aging_en and PCFGW_n.wr_port_aging_en registers.

5.1.3.3.3 DDRC Read Priorities and Write Priorities for Ports

The read channel of a port can be set to operate as HPR (High Priority Reads), LPR (Low Priority Reads) or VPR (Variable Priority Reads). The write channel of a port can be set as NPW (Normal Priority Writes) or VPW (Variable Priority Writes). Priority mapping is done through the arqos and awqos input based on the PCFGQOS0_n and PCFGWQOS0_n configuration registers.

The PA gives higher priority to an HPR read port than an LPR/VPW read port. NPW and VPW have the same initial priority. If any port is set to issue an HPR, SCHED.lpr_num_entries register must be programmed to a suitable value to reserve locations in the high priority queue of the read CAM. VPR that are not expired are treated as LPR from the arbiter point of view. If a VPR transaction expires in the XPI, it has higher priority than HPR or writes. VPW that are not expired are treated as NPW from the arbiter point of view. If a VPW transaction expires in the XPI, it has higher priority than normal writes. When multiple VPR/VPW commands expire simultaneously, the PA executes them in round-robin order to the DDRC.

5.1.3.3.4 Round-Robin Arbitration

After passing all tiers of arbitration, a tie is resolved by the final round-robin arbitration stage. The round-robin pointer starts from a port which has the lowest port index and after a grant, the pointer is moved to the first active requester after the one which has just received the grant. When page match feature is enabled, the PA locks on a port and grants to that port. However, the pointer of the round-robin arbiter continues to the subsequent ports, but, they are not granted in that round. They are granted in the subsequent rounds.

5.1.3.3.5 Page match

Page Match feature is the ability of Port Arbiter (PA) locking on a port when there are consecutive transactions to the same rank, bank, and row (same page). The grouping increases the DDR efficiency of the controller by adaptively choosing the back-to-back DDRC transactions to be granted at a time from a given port.

5.1.3.4 power saving

5.1.3.4.1 SDRAM Power Saving

SDRAM Power Saving Features are list as below:

- Self-refresh (except LPDDR4)
there are three methods to make the dram enter into self-refresh
 1. automatic enter into Self Refresh through configure PWRCTL.selfref_en
 2. software method enter into self-refresh through configure PWRCTL.selfref_sw
 3. hardware handshake interface((csysreq_ddrc/csysack_ddrc) with cactive_in_ddrc =0) enter into self-refresh
- Self-refresh power down (LPDDR4 only)
same as above
- Deep power-down
through configure PWRCTL.deeppowerdown_en enter into Deep power-down.

5.1.3.4.2 Fast frequency change

The uMCTL2 supports Fast Frequency Change, using up to four sets of timing registers. The alternative sets of timing registers can be found in UMCTL2_REGS_FREQ1, UMCTL2_REGS_FREQ2, and UMCTL2_REGS_FREQ3 registers.

Hardware Fast Frequency Change (HWFFC) functionality in uMCTL2 is intended to support clock frequency change procedure without the software intervention. HWFFC can support up to four sets of timing registers.

The following is a simplified HWFFC procedure that uMCTL2 performs automatically:

1. Blocks all the AXI ports from taking further commands and data
2. Optionally drains commands from the controller or stops sending further transactions to SDRAMs
3. Puts all the SDRAMs into self-refresh without powerdown (LPDDR4 only)
4. Programs several mode registers (MR) in all the SDRAMs as necessary
5. Puts all the SDRAMs into self-refresh (or SR-Powerdown when LPDDR4)
6. Interacts with PHY by using DFI frequency change protocol
7. Exits all the SDRAMs from self-refresh (or SR-Powerdown when LPDDR4)
8. Programs several mode registers (MR) in all the SDRAMs as necessary
9. Exits all the SDRAMs from self-refresh without powerdown (LPDDR4 only)

Note:

- During step (6), when csysack_ddrc becomes 0, clock frequency can be changed to a new target frequency.
- The HWFFC process can be managed by a hardware low-power controller. This interface is extended by the addition of three signals: csysmode_ddrc/ csysdiscamdrain_ddrc /csysfrequency_ddrc
- When the signal csysreq_ddrc is de-asserted, 1 on csysmode_ddrc indicates that a frequency change of the core_ddrc_core_clk is required. The input signal csysdiscamdrain_ddrc indicates whether the controller must be drained before the frequency change takes place. The signal csysfrequency_ddrc indicates which set of timing registers is used after the change of frequency. These signals must be held constant while csysreq_ddrc is de-asserted.
- When either HWFFCCTL.hwffc_en is 0 or both MSTR.lpddr4 are 0, even if the signal csysreq_ddrc is de-asserted and csysmode_ddrc is 1, then the uMCTL2 does not initiate HWFFC procedure.

5.1.3.4.3 Phy low power modes

The PHY power states can be grouped into two categories: Active (Mission Mode) and Standby. The PHY provides a mix of low power states with varying exit latencies so that memory controller / SoC can dynamically take

advantage of memory system idle windows.

Low Power State	PHY Exit Latency	PHY State	DRAM Commands available
DFI_LPDATA_ACK	T _{lp_wakeup}	DATA BYTES stopped Receivers are in stand-by	All commands (available other than data-bearing transactions).
DFI_LPCTRL_ACK	T _{lp_wakeup}	DATA BYTES stopped Receivers are in stand-by Chip selects inactive AC signals tristated/inactive except in LPDDR4 Mode ¹ MEMCLK toggling	DES or Maintain Power-down or Maintain Self-Refresh
DFI_LPCTRL_ACK +DRAMCLKSTOP	T _{lp_wakeup}	DATA BYTES stopped Receivers are in stand-by Chip selects inactive AC signals tristated/inactive except in LPDDR4 Mode ¹ MEMCLK stopped	Maintain Power-down or Maintain Self-Refresh (protocol specific).
LP2 +DFICKLSTOP	3us (3uS + 1280 DfiClks w/retain)	DATA BYTES stopped Receivers powered-down Chip selects inactive AC signals tristated/inactive except in LPDDR4 Mode ¹ MEMCLK stopped DFICKL stopped ZQ/LCDL Calibration off PLL in Standby	Maintain Self-Refresh
LP3 +DFICKLSTOP	33uS (33uS + 1280 DfiClks w/retrain)	DATA BYTES stopped Receivers powered-down Chip selects inactive AC signals tristated/inactive except in LPDDR4 Mode ¹ MEMCLK stopped DFICKL stopped ZQ/LCDL Calibration off PLL in PowerDown	Maintain Self-Refresh

5.1.3.4.4 IO retention mode

The standby power state has the following properties:

- This is the lowest possible power state for the PHY.
- In this power state, the DRAM may either be in self-refresh, or in Reset.
- The core VDD and VAA power supplies can be powered down in this state. If the core VDD power supply is shut off, power consumption is predominantly determined by the device leakage on VDDQ and the number of active CKE lanes.
- The state of CKE and MEMRESET pins will be pReserved at their last values when the core VDD is shut off. VDDQ must remain valid for the value of the pins to be pReserved.
- The Standby power state is entered via a Frequency Change sequence, using the dfi_init_start and dfi_freq[4:0] interface.

5.1.4 operation mode

5.1.4.1 initialization flow

1. Follow the PHYs power up procedure See PUB databook for details
2. Program the DWC_ddr_umctl2 registers
3. De-assert reset signal core_ddrc_rstn
4. Disable self-refresh, power down and assertion of dfi_dram_clk_disable by setting RFSHCTL3.dis_auto_refresh=1, PWRCTL.powerdown_en = 0, PWRCTL.selfref_en = 0, and PWRCTL.en_dfi_dram_clk_disable =0
5. Set SWCTL.sw_done to '0'
6. Set DFIMISC.dfi_init_complete_en to '0' (0x1 (DfiInitCompleteEn): The dfi_init_complete signal can be used to trigger SDRAM initialisation)

7. Set SWCTL.sw_done to '1'
8. Start PHY initialization and training by accessing relevant PUB registers CSR/IMEM/DMEM/micro controller
9. Poll the PUB register APBONLY.UctShadowRegs[0]=1'b0
10. Read the PUB Register APBONLY.UctWriteOnlyShadow for training status
11. Write the PUB Register APBONLY.DctWriteProt = 0
12. Poll the PUB register APBONLY.UctShadowRegs[0]=1'b1
13. Write the PUB Register APBONLY.DctWriteProt= 1
14. Poll the PUB register MASTER.CalBusy=0 (could be ignored about this step)
15. Set SWCTL.sw_done to '0'
16. Set DFIMISC.dfi_init_start to '1'
17. Set SWCTL.sw_done to '1'
18. Poll DFISTAT.dfi_init_complete=1
19. Set SWCTL.sw_done to '0'
20. Set DFIMISC.dfi_init_start to '0'
21. The following registers may need to be updated after training has completed:
 - ■ RANKCTL.diff_rank_wr_gap
 - ■ RANKCTL.diff_rank_rd_gap
 - ■ DRAMTMG2.rd2wr
 - ■ DRAMTMG2.wr2rd
 - ■ DRAMTMG9.wr2rd_s
 - ■ RANKCTL.diff_rank_wr_gap_msb
 - ■ RANKCTL.diff_rank_rd_gap_msb
 - ■ RANKCTL1.wr2rd_dr
 - ■ DFITMG1.dfi_t_wrdata_delay
 Also, the following registers related to VREF setting may need to be updated after 2D Training has completed:
 - ■ INIT7.mr6 - ■ INIT6.mr5
22. Set DFIMISC.dfi_init_complete_en to 1
23. Set PWRCTL.selfref_sw to '0'
24. Set SWCTL.sw_done to '1'
25. Wait for DWC_ddr_umctl2 to move to normal operating mode by monitoring STAT.operating_mode signal
26. Set back registers in step 4 to the original values if desired

5.1.4.2 DFS flow

For LPDDR3, there are no HWFFC function, Pls refer 6.3.3.6 in uMctl2 databook. They are pure software flow. Changing Clock Frequencies Through the DFI Frequency Change Interface When UMCTL2_-FAST_FREQUENCY_CHANGE = 1

The process of changing clock frequencies using UMCTL2_REGS_FREQ1/2/3 timing register set is as follows:

1. Program UMCTL2_REGS_FREQ1/2/3, whichever you prefer, timing register set with the timing settings required for the alternative clock frequency.
2. Write '0' to PCTRL_n.port_en. This blocks AXI ports from taking any transaction (blocks traffic on AXI ports).
3. Poll PSTAT.rd_port_busy_n = 0 and PSTAT.wr_port_busy_n = 0. Wait until all AXI ports are idle (the uMCTL2 controller must be idle).
4. Write '0' to SBRCTL.scrub_en. Disable SBR, required only if SBR instantiated.
5. Poll SBRSTAT.scrub_busy = 0. Indicates that there are no outstanding SBR read commands (required only if SBR instantiated).
6. Set DERATEEN.derate_mr4_pause_fc = 1, if DERATEEN.derate_eanble = 1 and the read latency (RL) value must change after frequency change (LPDDR2/3/4 only).
7. Set DBG1.dis_hif = 1 so that no new commands are accepted by the uMCTL2.

8. Poll DBGCAM.wr_data_pipeline_empty and DBGCAM.rd_data_pipeline_empty1, to ensure all outstanding commands have been sent on the DFI.
9. Poll DBGCAM.dbg_wr_q_empty and DBGCAM.dbg_rd_q_empty to ensure that write and read databuffers are empty. (MRCTRL0.mr_/_MRCTRL1.mr_).
10. Program DFILPCF0.dfi_lp_en_sr1 = 0. Poll DFISTAT.dfi_lp_ack = 0.
11. Wait until STAT.operating_mode[1:0] != 11 indicating that the DWC_ddr_umctl2 controller is in not self-refresh mode.
12. Assert PWRCTL.selfref_sw for the DWC_ddr_umctl2 controller to enter the self-refresh mode.
13. Wait until STAT.operating_mode[1:0] = 11 indicating that the DWC_ddr_umctl2 controller is in self-refresh mode. Ensure transition to self-refresh is due to software by checking that STAT.selfref_type[1:0] = 2'b10.
14. Set DBG1.dis_dq=1 so that no CAM commands are de-queued. This allows step 16 to occur without any Read/Write command executing with incorrect MR values
15. Poll DBGCAM.wr_data_pipeline_empty and DBGCAM.rd_data_pipeline_empty1, to ensure all outstanding commands have been sent on the DFI.
16. Set DFIMISC.dfi_init_complete_en to 0. It ensures that the uMCTL2 initialization state machine is not reset if the PHY must perform some initialization after frequency change.
17. Program DFIMISC.dfi_frequency as required. Refer to PHY databook for more details.
18. Set DFIMISC.dfi_init_start to 1. PHY performs internal sequences to cleanly stop pipelines, protect CKE/MEMCLK/MEMRESET and prepare for clock frequency change.
19. The PHY de-asserts dfi_init_complete. The controller polls DFISTAT.dfi_init_complete=0.
20. Change the clock frequency to the DWC_ddr_umctl2 controller ensuring there are no glitches.
21. Program MSTR2.target_frequency=2'b01 and MSTR.frequency_mode=1'b1 to select UMCTL2_REGS_FREQ1 registers.
22. Toggle RFSHCTL3.refresh_update_level to allow the new refresh-related register values to propagate to the refresh logic.
23. Set DFIMISC.dfi_init_start to 0. The PHY performs internal sequences to relock PLLs and calibrate ZQ/Delay-Lines.
24. The PHY asserts dfi_init_complete. The controller polls DFISTAT.dfi_init_complete=1.
25. Request the DWC_ddr_umctl2 controller to exit self-refresh by de-asserting PWRCTL.selfref_sw. Wait until STAT.operating_mode[1:0] != 11 indicating that the DWC_ddr_umctl2 controller is no longer in self-refresh mode.
26. Restore value DFILPCF0.dfi_lp_en_sr prior to self-refresh entry.
27. If necessary due to timing values changing due to frequency change, perform MRS/MRW commands through the Mode Register Read/Write signals (MRCTRL0.mr_/_MRCTRL1.mr_) to update MR register settings of the DRAM.

5.1.4.3 retention flow

5.1.4.3.1 IO Retention Enter flow

The following is the full process to enter LP3/IO Retention

1. Write '0' to PCTRL_n.port_en to blocks AXI ports from taking anymore transactions
2. Poll PSTAT.rd_port_busy_n = 0 Poll PSTAT.wr_port_busy_n = 0 Waits unit all AXI ports are idle
3. Write '0' to SBRCTL.scrub_en Disables SBR, only required if SBR instantiated
4. Poll SBRSTAT.scrub_busy = 0 Indicates that there are no outstanding SBR read commands, only required if SBR instantiated.
5. Write '1' to PWRCTL.selfref_sw Causes system to move to Self Refresh state For LPDDR4, PWRCTL.stay_in_selfref must be set to '0' to enter Self Refresh Power down.
6. Poll STAT.selfref_type= 2'b10 Poll STAT.selfref_state = 2'b10 (LPDDR4 only) Waits until Self Refresh state is entered If DDR4 retry is enabled by CRCPARCTL1.crc_parity_retry_enable and software intervention is enabled by CRCPARCTL1.alert_wait_for_sw. Also monitor CRCPARSTAT.dfi_alert_err_int and CRCPARSTAT.dfi_alert_err_fatl_int during the polling STAT.selfref_type. If one or more of them are asserted

before the polling is done, complete the retry procedure prior to the subsequent steps. For LPDDR4, wait until Self Refresh Power Down state is entered.

7. Transition the PHY to the LP3/IO Retention state by using a DFI Frequency Change operation.
8. Set the PwrOkIn signal to 0. This enables the data retention feature on the CKE and MEMRESET pins. These pins hold the current values.
 - Remaining PHY input signals must be valid, inactive states for at least 10 ns after the PwrOkIN deasserting edge.
 - The PHY input clocks are not required to toggle after the de-asserting edge of PwrOkIN
9. Turn off the core VDD and VAA supplies. These supplies can be turned off in any order. The VDDQ power supply must remain active during LP3/IO Retention.
10. All PHY input values are now don't cares and can go to X in simulation.

Note:

1. In step1 to step6, it could be entered into self-refresh(or self refresh power down in LPDDR4) by ddrc hardware low power interface handshake. sysctl 3.5.2 chapter.
2. The number of registers that must be saved prior to LP3/IO Retention entry and restored post LP3/IO Retention Exit varies based upon the DDR standard and the training steps run. Phylnit can be used to generate an initialization sequence that automatically includes saving retention registers based on PHY configuration and protocol.
3. Except note2), The following registers in controller may need to be saved before enter into data retention and reloaded after training has completed

RANKCTL.diff_rank_wr_gap
RANKCTL.diff_rank_rd_gap
DRAMTMG2.rd2wr
DRAMTMG2.wr2rd
DRAMTMG9.wr2rd_s
RANKCTL.diff_rank_wr_gap_msb
RANKCTL.diff_rank_rd_gap_msb
RANKCTL1.wr2rd_dr
DFITMG1.dfi_t_wrdata_delay

Also, the following registers related to VREF setting may need to be updated after 2D Training has completed:

INIT7.mr6
INIT6.mr5

4. Prior to the LP3/IO Retention Enter procedure, the values of the PHY configuration and trained registers must be saved as part of the PHY initialization process

5.1.4.3.2 IO Retention exit

The process to exit LP3/IO Retention is similar to the initialization sequence. For DDRPHY, Phylnit can be used to automatically generate a detailed retention exit sequence for a given configuration and DRAM protocol.

1. The VDDQ power supply will already be active; restore the core VDD and VAA power supplies
2. Enable power and reset controller/PHY by driving core_ddrc_rstn = 1'b0, areset_n = 1'b0, presetn = 1'b0
3. Re-enable clocks if clocks were disabled after power removal
4. Remove APB reset, presetn = 1'b1, and re-program the registers to pre-power removal values
5. Program INIT0.skip_dram_init = 2'b11 Skips the DRAM init routine and starts up in self-refresh mode
6. Programs PWRCTL.selfref_sw = 1'b1 Keeps the controller in self-refresh mode
7. Program DFIMISC.dfi_init_complete_en to 1'b0 PHY initialization needs to be rerun so set to '0' until initialization complete
8. Remove the controller reset core_ddrc_rstn = 1'b1 aresetn_n = 1'b1
9. Run PHY initialization/training as required, including removing the IOs from retention mode Synopsys legacy PHYs may require more steps. Refer to the relevant PHY Databook for more information.

- Restore all PHY registers
- Initialize the PHY to mission mode through a DFI initialization request. The detail, pls refer 6.4.3.3 in PUB databook.

10. Program DFIMISC.dfi_init_complete_en to 1'b1 Indicates to controller that PHY has completed re-training/initialization
11. Program PWRCTL.selfref_sw = 1'b0 trigger self-refresh exit
12. Poll STAT.selfref_type = 2'b00 Wait until self-refresh state is exited
13. Poll STAT.operating_mode for Normal Mode entry
14. Write PCTRL.port_en = 1 AXI ports are no longer blocked from taking transactions

5.1.5 register info

for controller registers, please refer chapter 5 in the uMctI2 databook. for phy registers, please refer the chapter 12 in the pub databook.

5.2 Sram

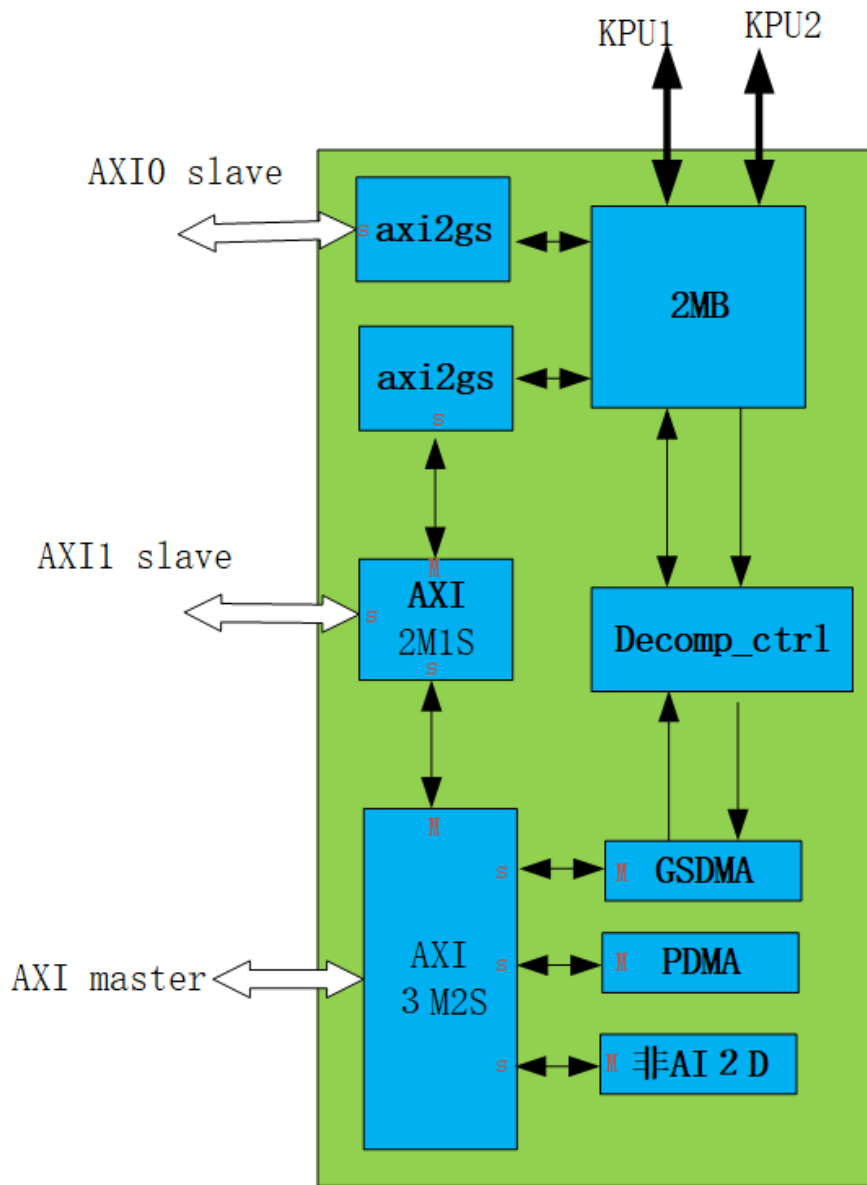
5.2.1 Overview

The shared 2MB sram provides access to CPU/KPU/decompression and other modules. Key features:

- Two independent 128bit AXI4 Slave buses. Burst supports incremental, fixed, and wrap. The maximum burst length is 16. Exclusive transmission is not supported. The maximum outstanding read and write are 16 respectively.
- The 2MB sram is divided into two groups. The first group is shared by KPU0 port, axi slave0, axi slave1, and decompression module. When decompression does not work, KPU0 has the highest priority, followed by axi slave0/axi slave1. The other group of KPU1 ports, axi slave 0, and axi slave 1 are shared. KPU1 has the highest priority.
- The decompression module shares 768KB SRAM to cache the decompression process data.

5.2.2 Block Diagram

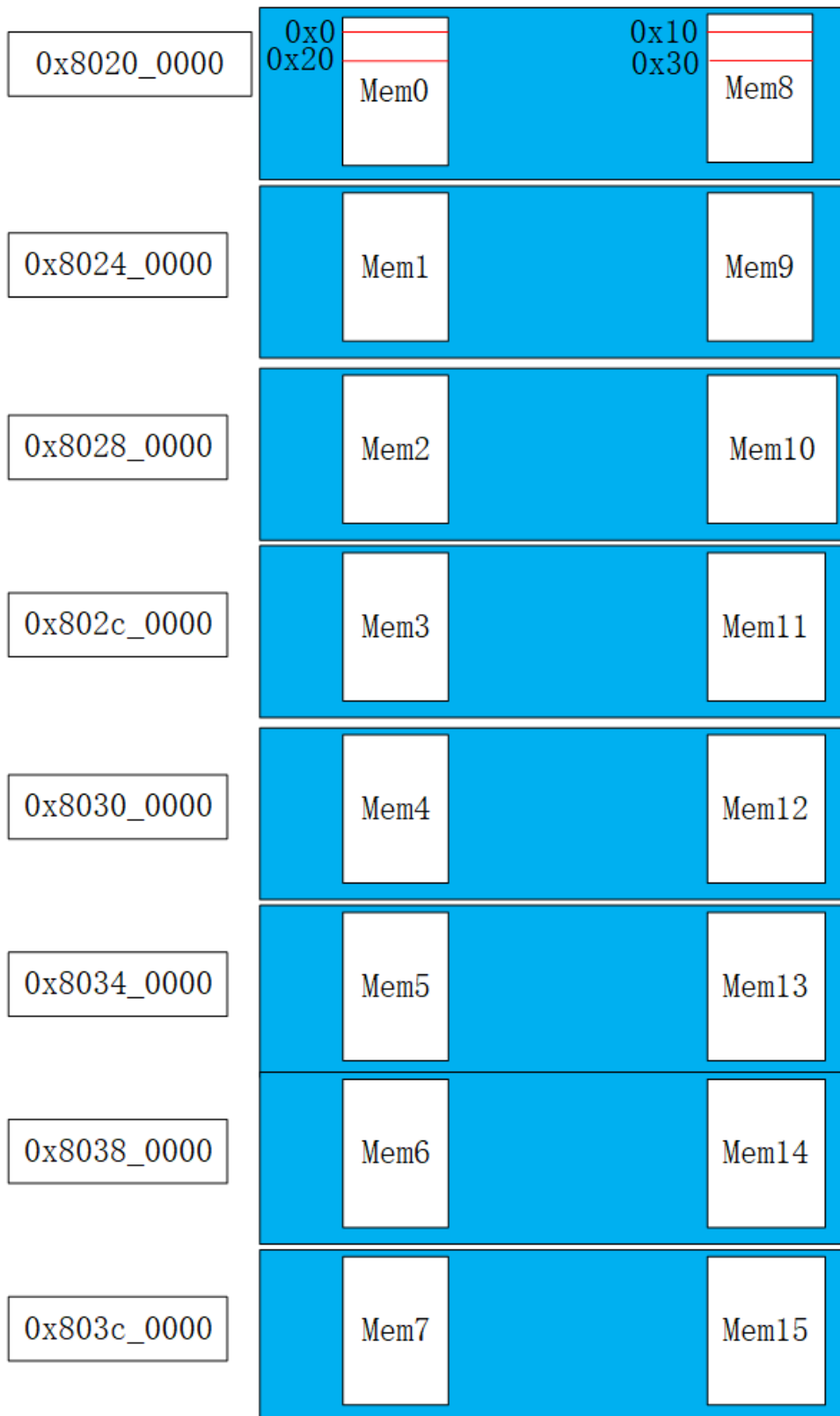
[Figure 5-2-1](#) sram block diagram



5.2.3 Function Description

5.2.3.1 Sram space access description

5.2.3.1.1 Decompression module does not work



A total of 2MB sram is divided into 16 sram blocks, Mem 0~15. Mem 0~7 is accessed by KPU port 0, and Mem 8~15 is accessed by KPU port 1. In order to make the kpu0 port and kpu1 port work in parallel, the address spaces Mem0~7 and Mem8~15 odd and even addresses seen from the bus side are spliced. For example, as shown in the figure above, the 16 bytes of the offset address 0x0 are in Mem0, and the 16 bytes of the offset address 0x10 are in Mem8. The 16 bytes of offset address 0x20 are in Mem0, and the 16 bytes of offset address 0x30 are in Mem8. Mem0 and Mem8 are a pair, Mem1 and Mem9 are a pair, and each pair of sram takes up 256KB of space.

5.2.3.1.2 Decompression working mode



When decompressing, 6

sram (yellow part) are occupied. In order not to affect the continuity of other addresses, Mem0/Mem1/Mem2/Mem8/Mem9/Mem10 is used as the decompression module. When these srams are decompressed, Mem9/Mem10 (green sram) as input buffer, Mem0/Mem1/Mem2/Mem8 (red sram) as output buffer. The rest of the block space mapping is still an interleaving mechanism, regardless of whether the decompression works.

5.2.3.2 Sram Arbitration Access Description

16 sram independent arbitration. When kpu0/1, axi0/1, and decomp access different memory blocks at the same time, they can be accessed at the same time. The arbitration policy when accessing at the same time is as follows:

- When decompressing, Mem0/Mem1/Mem2/Mem8/Mem9/Mem10 are used as bus buffers because decompressing. These blocks have the highest decompression priority, followed by axi0, and axi1.
- When decompressing non working or non decompressed memory blocks KPU has the highest priority, followed by axi0 and axi1.

5.2.3.3 Function description

1. Decompression (configure the decompression enable register (decomp_ctrl_en)) Occupy Mem0/Mem1/Mem2/Mem8/Mem9/Mem10, decompress access space DMA reads sram address output Block0 0x8020_0000~0x8021_ffff output Block1 0x8022_0000~0x8023_ffff output Block2 0x8024_0000~0x8025_ffff output Block3 0x8026_0000~0x8027_ffff DMA write sram address Input Block0 0x8028_0000~0x8029_ffff Input Block1 0x802a_0000~0x802b_ffff Space accessible by two Axi slave ports 0x802c_0000 ~ 0x803f_ffff
2. When KPU and decompression work at the same time The six 128KB sram and kpu used for decompression should be avoided. That is, the kpu can use 0x802c_0000 ~ 0x803f_ffff space.
3. KPU continuous access If KPU accesses the sram continuously and uninterruptedly, in this scenario, in order to access the same sram as the axis slave and cause a hang, the software needs to avoid the kpu and sram accessing the same 128kb sram.

5.3 Flash Memory Controller

5.3.1 Overview

5.3.1.1 General Description

The flash memory controller (FMC) provides memory controller interfaces for connecting to external SPI NAND flash or SPI NOR flash memories to access data.

5.3.1.2 Features

- Supports the SPI NOR flash and SPI NAND flash
- Supports the read and write operations in internal DMA mode for the SPI NAND flash and the SPI NOR flash.
- Support Enhanced (Dual/Quad/Octal) SPI features:
 - Programmable address, instruction, wait cycles, and data frame size
 - Dual Data Rate (DDR) support
 - Read Data Strobe support for DDR transfer
- Data item size (4 to 32 bits) - Item size of each data transfer under control of the programmer
- Execute In Place (XIP) mode support:
 - AHB address is mapped to SPI NOR flash address directly
 - Read and write SPI NOR flash directly
- Read and write SPI NAND flash need to operate according to the NAND flash operation requirement
- Except XIP mode, all operations need to operate according to the flash operation requirement
- Programmable delay on the sample time of received serial data bit (rxd)
- Programmable clock bit rate
- The latency of strb can be adjusted

5.3.2 Function Description

5.3.2.1 Interface

The FMC supports five types of SPIs, including the standard SPI, dual-output/dual-input SPI, quad-output/quad-input SPI, dual I/O SPI, and quad I/O SPI. The standard SPI, dual I/O SPI, and quad I/O SPI support the STR and DDR modes.

Standard SPI

The standard SPI is connected to a 1-bit data input line and a 1-bit data output line. Figure 5-3-1 shows the write timing of the standard SPI. [Figure 5-3-1 Write Operation in Standard SPI Format](#)

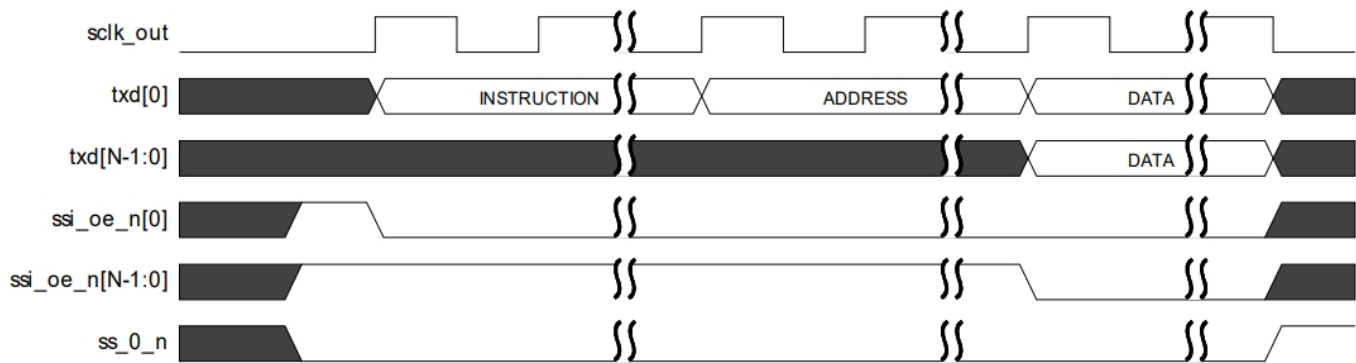
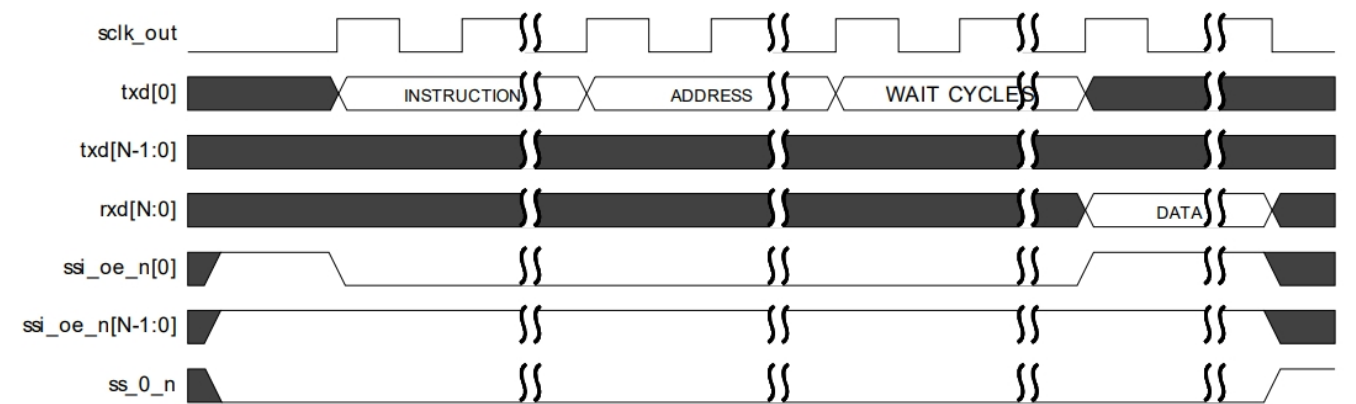


Figure 5-3-2 shows the read timing of the standard SPI. [Figure 5-3-2 Read timing of the standard SPI](#)

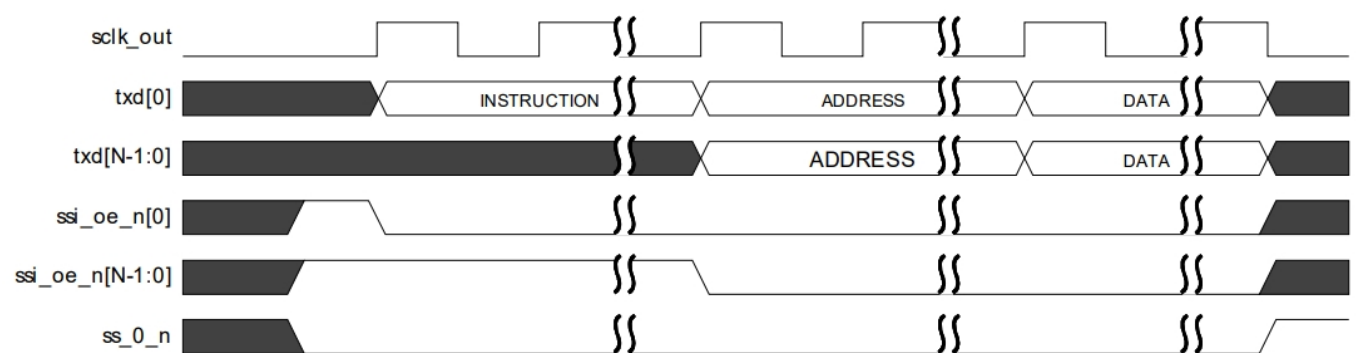


Note the following:

- For these, `SPI_CTRLR0.TRANS_TYPE` field must be set to 00.
- The value of N will be: 7 if `CTRLR0.SPI_FRF` is set to 11, 3 if `CTRLR0.SPI_FRF` is set to 10, and 1 if `CTRLR0.SPI_FRF` is set to 01.

Enhanced(Dual/Quad/Octal) SPI

Figure 5-3-3 and Figure 5-3-4 show Instruction transmitted in standard and address transmitted in Enhanced SPI format. [Figure 5-3-3 Write Operation](#)



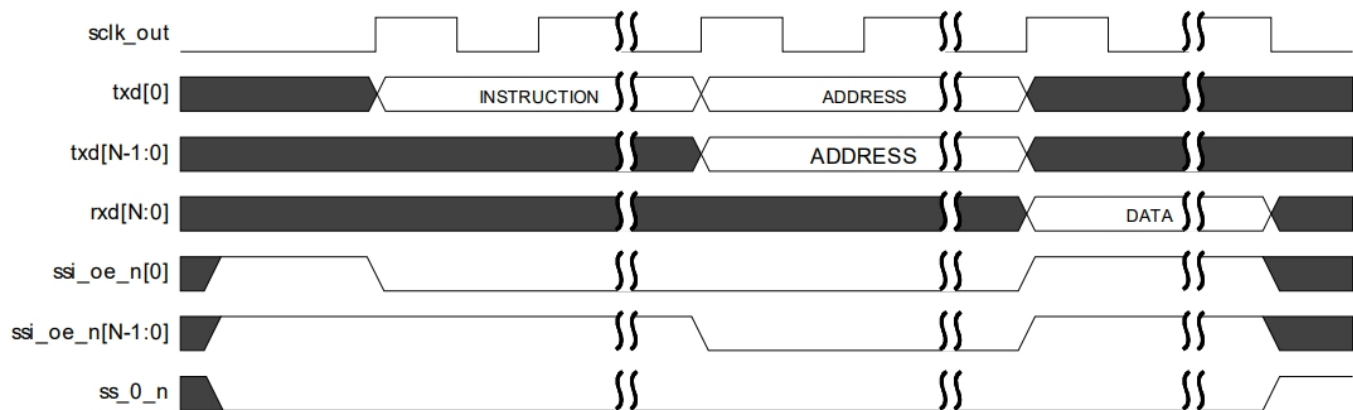


Figure 5-3-4 Read Operation

Note the following:

- For this, SPI_CTRLR0.TRANS_TYPE field should be set to 01.
- The value of N will be: 7 if CTRLR0.SPI_FRF is set to 11, 3 if CTRLR0.SPI_FRF is set to 10, and 1 if CTRLR0.SPI_FRF is set to 01.

Figure 5-3-5 and Figure 5-3-6 show Instruction and Address both transmitted in Enhanced SPI format. [Figure 5-3-5](#) Write Operation

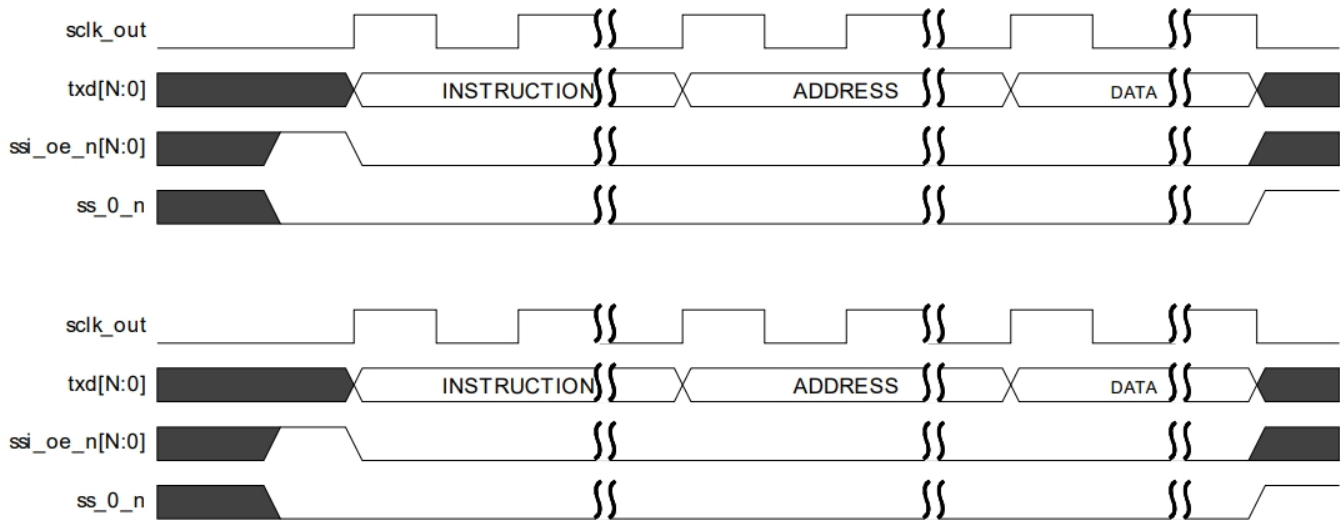
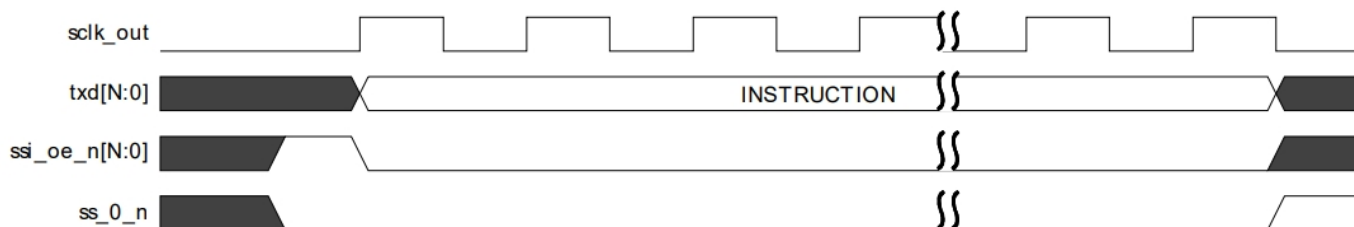


Figure 5-3-6 Read Operation

Note the following:

- For this, SPI_CTRLR0.TRANS_TYPE field must be set to 10.
- Figures show the timing diagram in which instruction and address are transmitted in SPI format specified in the CTRLR0.SPI_FRF field.
- The value of N will be: 7 if CTRLR0.SPI_FRF is set to 11, 3 if CTRLR0.SPI_FRF is set to 10, and 1 if CTRLR0.SPI_FRF is set to 01.

Figure 5-3-7 shows Instruction only transfer in enhanced SPI format. [Figure 5-3-7](#) Instruction only transfer in enhanced SPI format



5.3.2.2 Operations in Internal DMA Mode

The FMC internal DMA feature is responsible for data transfer from an internal memory (or on chip memory) to an external SPI device, and vice versa. The FMC uses AXI master interface on one side and SPI interface on the other, which acts as a source or a destination depending on the programming.

5.3.2.3 RXD Sample Delay

Additional logic can be included in the design in order to delay the default sample time of the rxd signal. This additional logic can help to increase the maximum achievable frequency on the serial bus. The FMC uses RX_SAMPLE_DLY register to change the sampling point of rxd signal.

- If RX_SAMPLE_DLY field of the SE register is set to 0, then SPI delays the sampling point by the programmed number of ssi_clk cycles.
- If RX_SAMPLE_DLY field of the SE register is set to 1, then SPI delays sampling point by programmed number of ssi_clk cycles + 0.5 * (ssi_clk period). Thus, the sampling is done on negative edge of ssi_clk signal as described in the figure Figure 5-3-8.

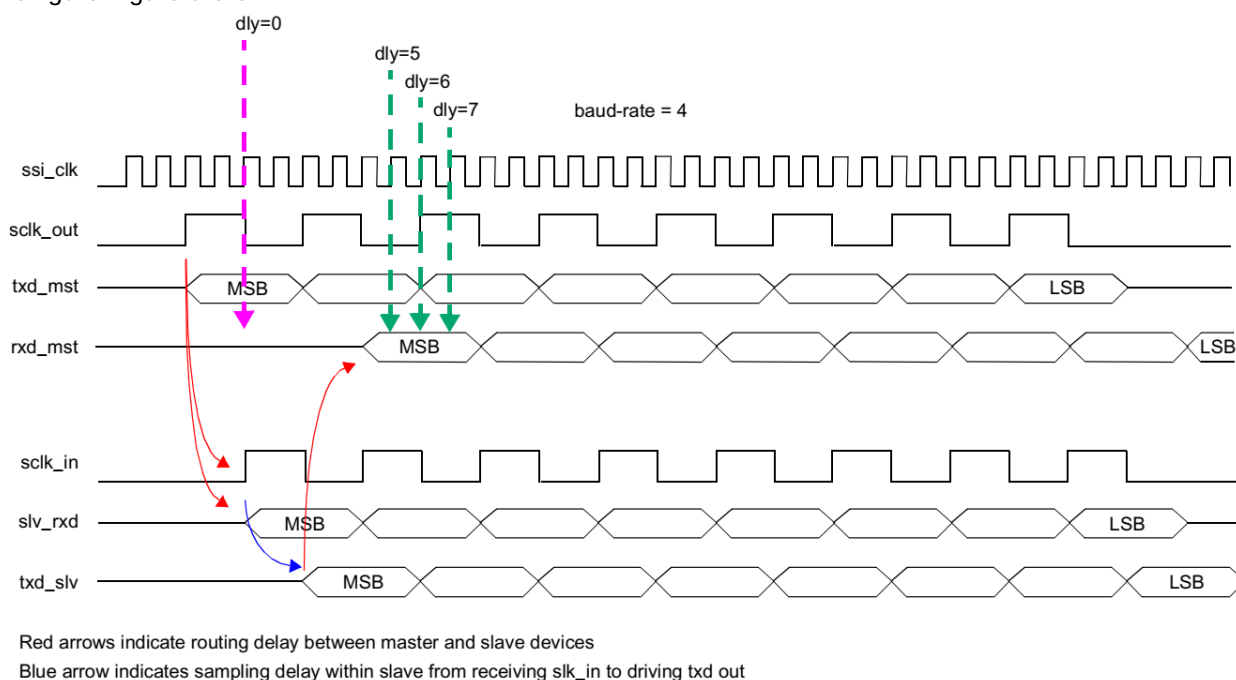


Figure 5-3-8 Effects of Round-Trip Routing Delays on the sclk_out Signal The slave uses the sclk_out signal from the master as a strobe in order to drive rxd signal data onto the serial bus. Routing and sampling delays on the sclk_out signal by the slave device can mean that the rxd bit has not stabilized to the correct value before the master samples the rxd signal. Figure 5-3-8 shows an example of how a routing delay on the rxd signal can result in an incorrect rxd value at the default time when the master samples the port. Note:

- If RX_SAMPLE_DLY field of the SE register is set to 0, then the FMC delays the sampling point by the programmed number of ssi_clk cycles.
- If RX_SAMPLE_DLY field of the SE register is set to 1, then the FMC delays sampling point by programmed number of ssi_clk cycles + 0.5 * (ssi_clk period). Thus, the sampling is done on negative edge of ssi_clk signal.

- RX_SAMPLE_DLY field of the SE register is available only if SSIC_HAS_RX_SAMPLE_DELAY configuration parameter is set to 2 (Value Use Both Positive and Negative Edges).

5.3.2.4 Dual Data-Rate (DDR) Support

For improved throughput, the dual data-rate transfer can be used for reading or writing to the memories. The DDR mode supports the following modes:

- Mode 0 - When the default serial clock phase and default serial clock polarity are not enabled (SCPH = 0 & SCPOL = 0)
- Mode 3 - When the default serial clock phase and default serial clock polarity are enabled (SCPH = 1 & SCPOL = 1) DDR commands enable data to be transferred on both edges of clock. Following are the different types of DDR commands:
- Address and data are transmitted (or received in case of data) in DDR format, while instruction is transmitted in standard format.
- Instruction, address, and data are all transmitted or received in DDR format. The SPI_DDR_EN (SPI_CTRLR0[16]) bit is used to determine if the Address and data have to be transferred in DDR mode and INST_DDR_EN (SPI_CTRLR0[17]) bit is used to determine if Instruction must be transferred in DDR format. These bits are only valid when the CTRLR0.SPI_FRF bit is set to be in Dual, Quad or Octal mode. Figure 5-3-9 describes a DDR write transfer where instruction, address and data all are transferred in DDR format.

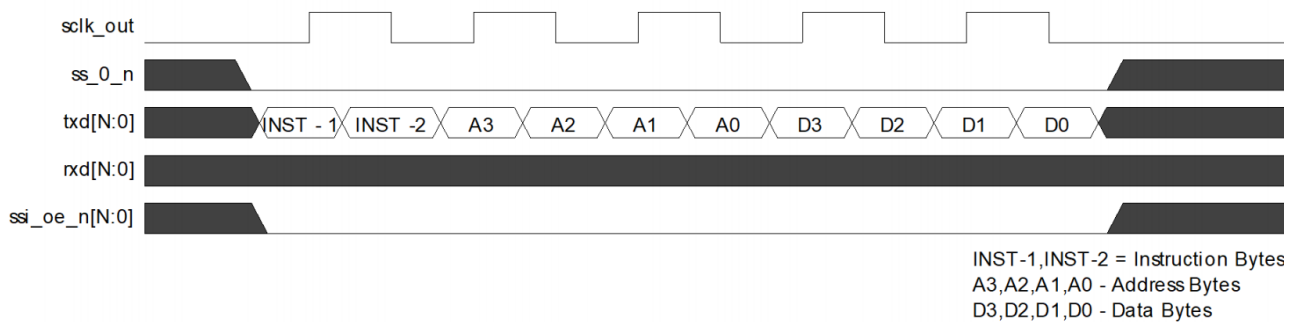


Figure 5-3-9 DDR Transfer with Instruction, Address and Data Transmitted in DDR Format

5.3.2.5 Clock Ratios

The frequency of sclk_out signal can be derived from the following equation. $F_{sclk\_out} = F_{ssi\_clk} / SCKDV$ Where, SCKDV is a programmable register holding any even value in the range 0 – 65,534. If SCKDV = 0, sclk_out is disabled. When in DDR mode, SCKDV needs to be greater than or equal to 2. The sclk_out line only toggles when an active transfer is in progress. At all other time, it is held in inactive state, as define by the serial protocol under which it operates. Figure 5-3-10 illustrates the maximum ratio between sclk_out signal and ssi_clk signal.

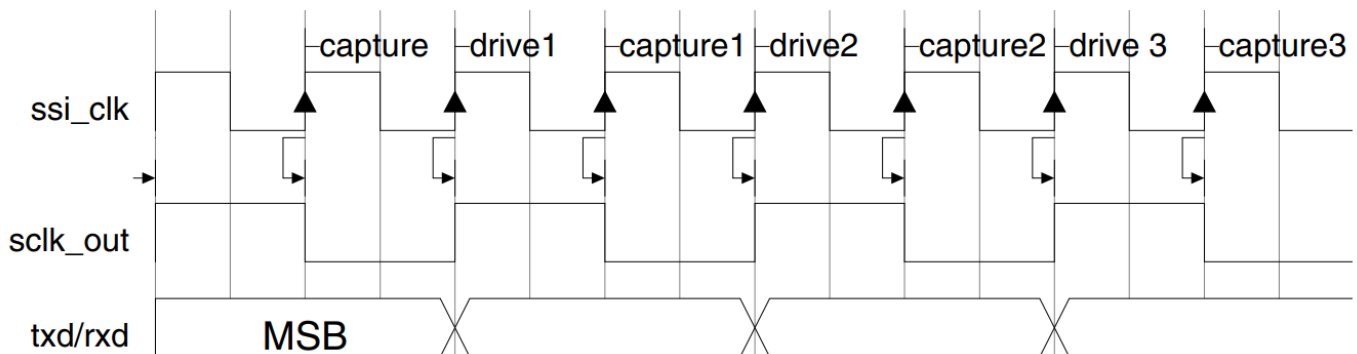


Figure 5-3-10 Maximum sclk_out/ssi_clk Ratio

5.3.2.6 Read Data Strobe Signal

To achieve high frequency, data strobe signaling is used in read operations. The FMC provides the SSIC_HAS_RXDS configurable parameter to include the data strobe signal. When SSIC_HAS_RXDS is set to 1, the SPI_CTRLR0 field of the SPI_RXDS_EN register needs to be programmed to 1 to sample incoming data with respect to the read data strobe signal, as shown in Figure 2-83. If the SPI_CTRLR0.SPI_RXDS_EN signal is not set to 1, the data is sampled based on the existing logic. It is expected that the device uses the incoming clock and feeds it back to The FMC as a data strobe. The FMC transmits only sufficient amount of clock cycles to complete the transfer. If the number of data frames expected is four, The FMC transmits only four clocks as shown in Figure 5-3-11.

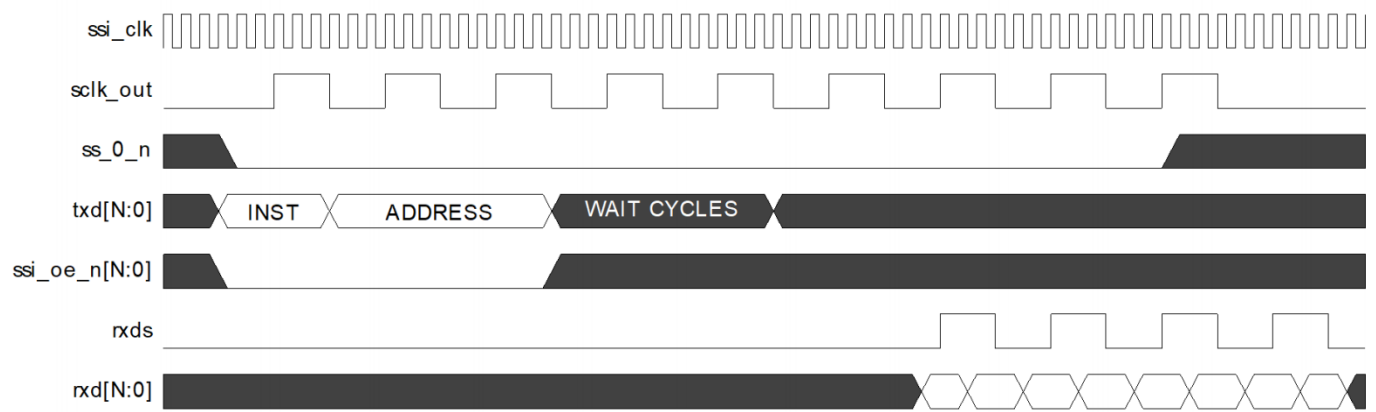


Figure 5-3-11 Example of a Transfer Using Read Data Strobe Signal

5.3.2.7 eXecute In Place (XIP) Mode

SPI provides a function to directly perform memory read operation from AHB transaction. This is called execute in place mode. The XIP mode can be enabled in SPI by selecting the configuration parameter SSIC_XIP_EN or SSIC_XIP_WRITE_EN (for write). This includes an extra sideband signal xip_en on an AHB interface. This signal level decides if the AHB transfers are register read-write or XIP transfers. If xip_en signal is driven to 1, then SPI expects a read request to be made on the AHB interface. This request is translated to SPI read on the serial interface. As soon as the data is received, it is returned to the AHB interface in the same transaction. The haddr signal is used to derive the address to be sent on the SIP interface.

5.3.3 Working Process

5.3.3.1 Process of Reading Data in XIP transfer

The programming flow to set-up an XIP transfer is as follows:

1. Set the SPI frame format value in CTRLR0 register.
2. Set the Address length, Wait cycles, and transaction type in SPI_CTRLR0.

Note that, the maximum address length is 32. The instruction phase can also be included in XIP transfer by using SPI_CTRLR0 field of the XIP_INST_EN register. In this case, the following registers must be set:

1. Set length of instruction in the SPI_CTRLR0 register.
2. Write the instruction opcodes in the XIP_INCR_INST and XIP_WRAP_INST registers.

When SPI_CTLR0 field of the XIP_DFS_HC register is set to 0, AHB control signals are used to derive the values of the data frame size and the number of data frames to be fetched. hsize signal is used to get the value of data frame size for the transfer. Table 5-3-1 shows the data frame size mapping from the HSIXZE value.

Table 5-3-1 hsize to Data Frame Size Decode

hsize	Data frame Size (DFS)
3'b000	8

hsize	Data frame Size (DFS)
3'b001	16
>=3'b010	32

The number of data frames to be fetched is derived from the HBURST signal. Table 5-3-2 shows the decode for the number for data frames decoded from the HBURST value.

Table 5-3-2 HBURST to Number of Data Frames (XIP_DFS_HC = 0)

HBURST	Type	Number of Data Frames (NDF
000	Single	1
001	INCR	Data is fetched until the burst completes
010	WRAP4	4
011	INCR4	4
100	WRAP8	8
101	INCR8	8
110	WRAP16	16
111	INCR16	16

XIP Transfers

Once the CTRLR0 and SPI_CTRLR0 registers are programmed with correct values for transfer, then XIP transfer can be initiated by driving the xip_en signal to 1 and by performing a read operation on the AHB bus. Figure 2-103 shows an XIP transfer on an AHB interface. The transfer is of 4 beats starting from address 0x0. In Figure 2-103, note the following:

- The length of data frame is derived from AHB signal hsize that is 32-bit.
- The xip_en signal must be active for the address phase of the burst.
- The address sent on AHB SPI line is same as the address sent by the AHB interface. The number of bits sent on SPI line is defined by ADDR_L field of the SPI_CTRLR0 register.

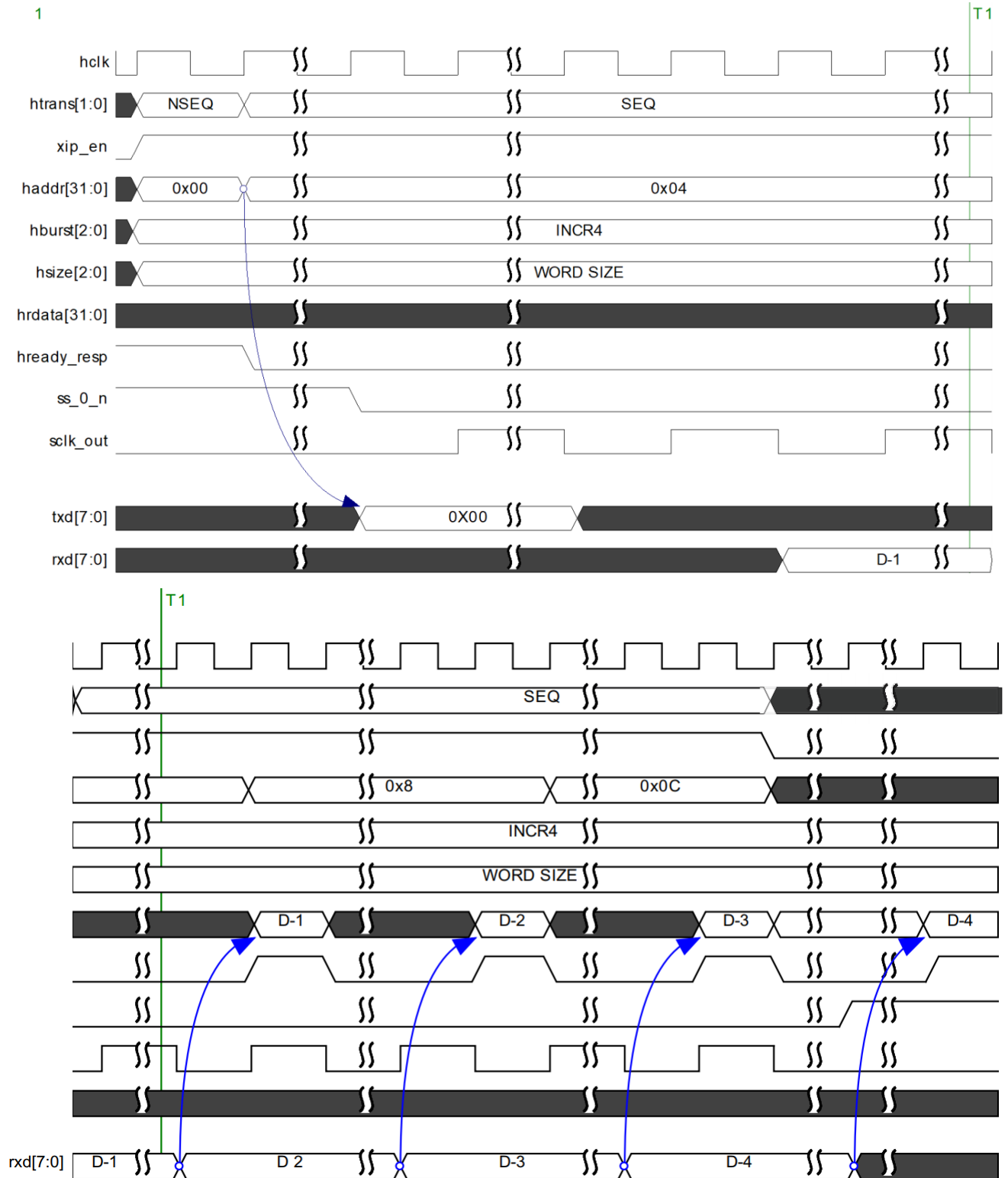


Figure 5-3-12 XIP Transfer

AHB WAIT Transfers

During XIP transfers, AHB master may insert wait states in the transfer. In this condition, the transfer does not break on interface and continues to fetch the data from slave until all the required data frames are fetched for the ongoing burst. This intermediate data is stored in receive FIFO and is transmitted to AHB interface after WAIT cycles are completed.

Mode Bits Support

During a serial transfer, certain devices have the capability to send additional bits that are referred to as Mode bits. The mode bit support can be enabled by setting the SPI_CTRLR0 field of the XIP_MD_BIT_EN register. The actual value of the mode bits to be driven can be programmed by writing into the XIP_MODE_BITS register. The length of the mode

bits can be programmed using SPI_CTRLR0 field of the XIP_MBL register or XIP_CTRL field of the XIP_MBL register. Mode bits occur immediately after the address phase is completed and follow the same rules as for the address bits. Therefore, serial transfers consist of following phases when mode bit support is enabled:

Instruction Phase > Address Phase > Mode Bits > Wait cycles > Data Phase

Based on the CTRLR0.SPI_FRF value, mode bits are driven as follows:

- If SPI_FRF is Dual, the Mode bits are sent on two lanes.
- If SPI_FRF is Quad, the Mode bits are sent on four lanes.
- If SPI_FRF is Octal, the Mode bits are sent on eight lanes.

5.3.3.2 Process of Writing Data in XIP transfer

The FMC can write into the memory without software overhead of writing the instruction and address separately inside the transmit buffer. XIP for write feature executes the write transactions directly without software intervention. Assert the xip_en signal while performing write into the controller. After the controller detects such transfer it inserts the address and instruction (XIP_WRITE_WRAP_INST or XIP_WRITE_INCR_INST when SSIC_XIP_WRITE_REG_EN is enabled and XIP_WRAP_INST or XIP_INCR_INST when SSIC_XIP_WRITE_REG_EN is disabled) into the transfer and initiates a write transaction.

Transfer End Conditions

The transfer ends under the following conditions:

- If IDLE transfer is detected in-between an XIP write transfer.
- Transmit FIFO goes empty. The interrupt(ssi_txu_intr) is raised when the transmit FIFO goes empty. The interrupt is an underflow interrupt, which is used to indicate the empty state of the transmit FIFO. For EBT conditions, the data bits sent until the slave is deselected is considered valid, and are transferred to SPI side. The master needs to resume this transfer as a new transfer including the address and instruction phase.

A typical XIP write transfer is shown in Figure 5-3-13

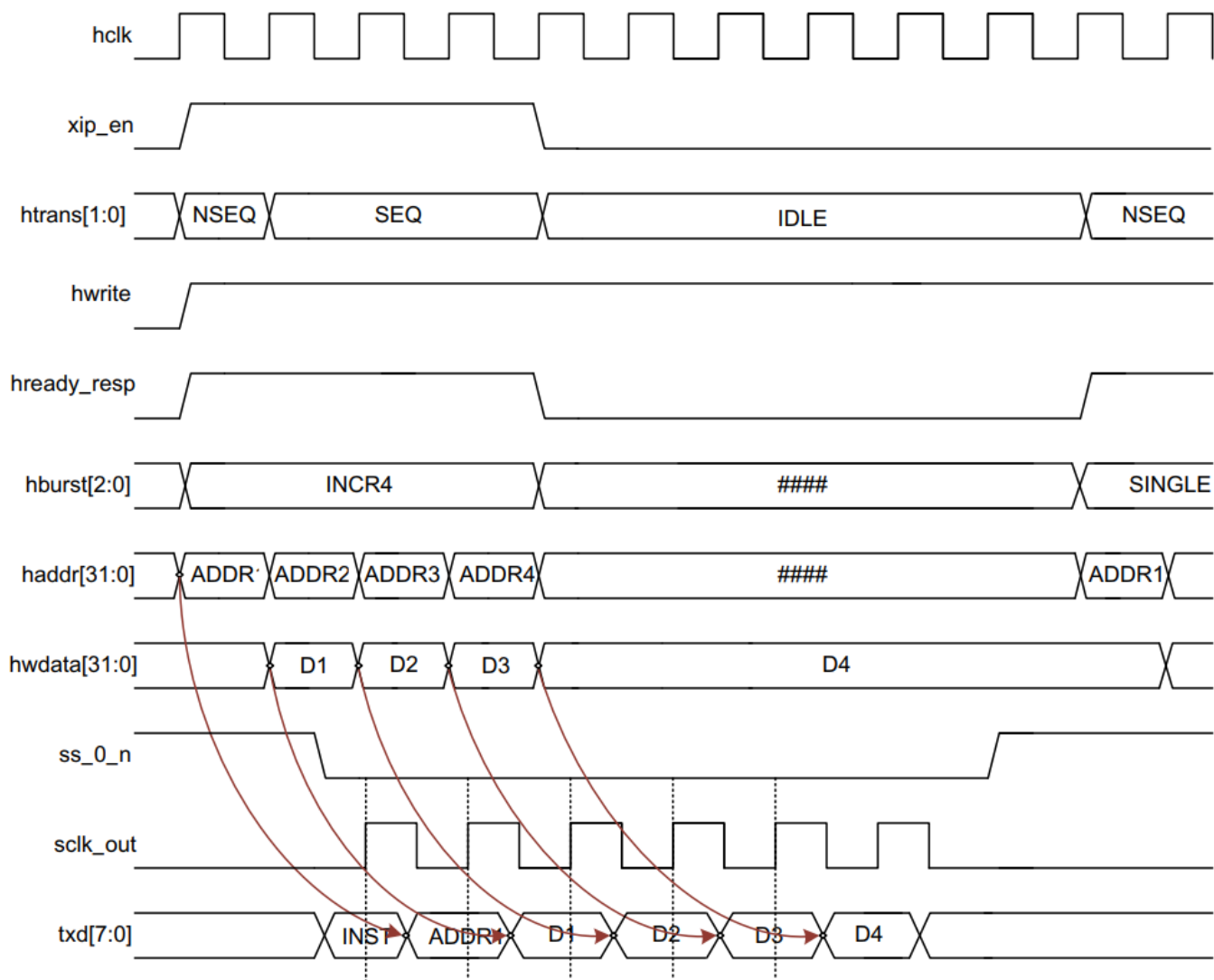


Figure 5-3-13 XIP Write Transfer

Note:

Before issuing a write XIP transfer on AHB interface, ensure that the following conditions are met:

- SSIENR is enabled.
- Tx FIFO is empty.
- Tx underflow interrupt is cleared.

If any of these conditions are not met, the controller issues an error response for write XIP transfer.

5.3.3.3 Transmitting Data in DDR Mode

In DDR mode, data is transmitted on both edges so that it is difficult to sample data correctly. An internal register determines the edge on which the data should be transmitted. This ensures that the receiver is able to get a stable data while sampling. The internal register (DDR_DRIVE_EDGE) determines the edge on which the data is transmitted. The FMC sends data with respect to baud clock, which is an integral multiple of the internal clock ($ssi\_clk * BAUDR$). The data needs to be transmitted within half clock cycle ($BAUDR/2$), therefore the maximum value for DDR_DRIVE_EDGE register is equal to $[(BAUDR/2)-1]$. Data is driven before $((BAUD/2)-TXD\_DRIVE\_EDGE)$ ssi_clk before the sampling edge. If the programmed value of DDR_DRIVE_EDGE is 0 then data is transmitted edge-aligned with respect to $sclk\_out$ (baud clock). If the programmed value of DDR_DRIVE_EDGE register is set to 1 then the data is transmitted two ssi_clk before the edge of $sclk\_out$. In Figure 5-3-14, transmit edge and driving edge of the data are the same. This is default behavior in DDR mode. Figure 2-81, and Figure 2-82 show how data is transmitted using different values of

the DDR_DRIVE_EDGE register.

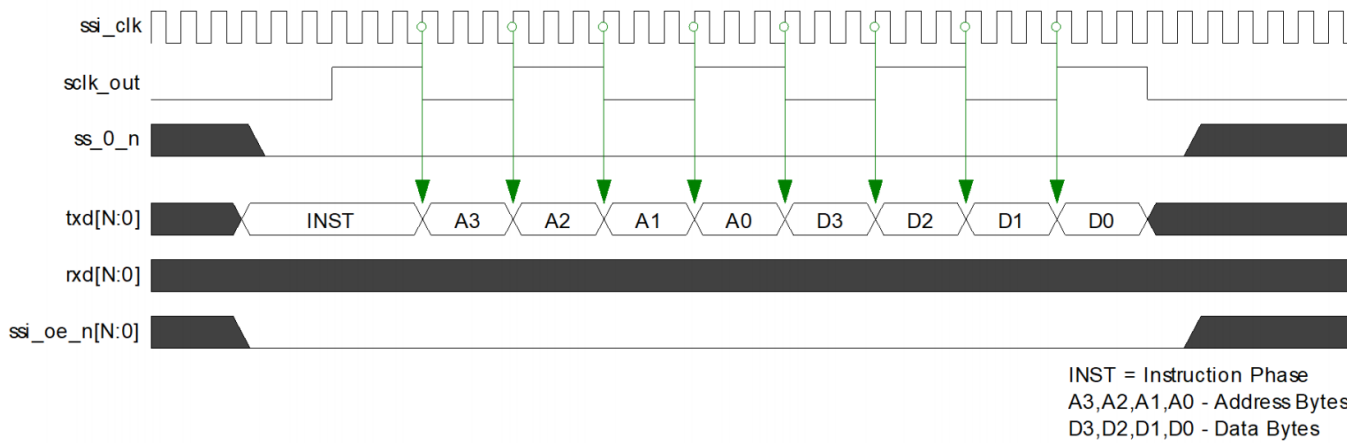


Figure 5-3-14 Transmit Data With DDR_DRIVE_EDGE = 0

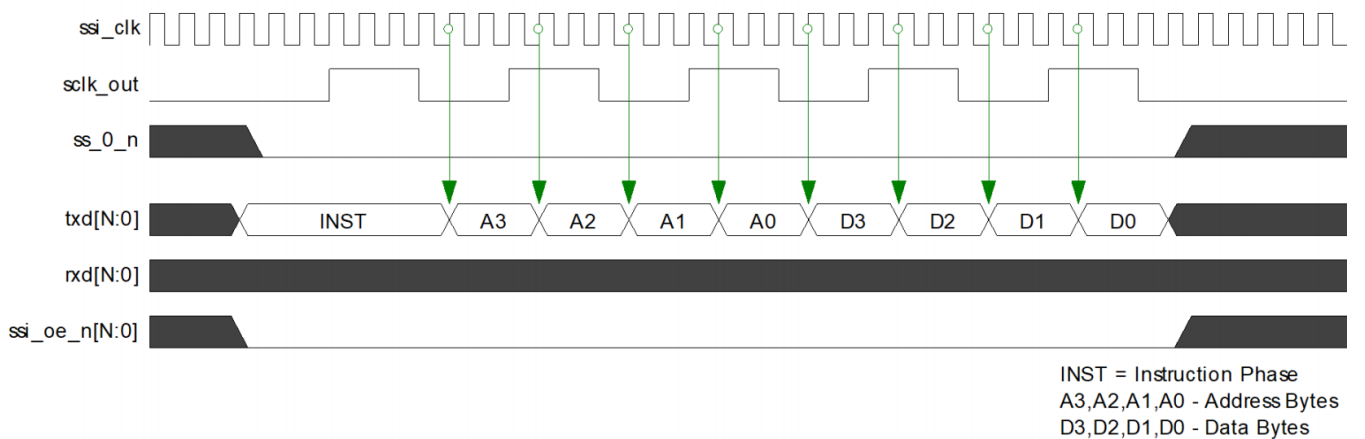


Figure 5-3-15 Transmit Data With DDR_DRIVE_EDGE = 1

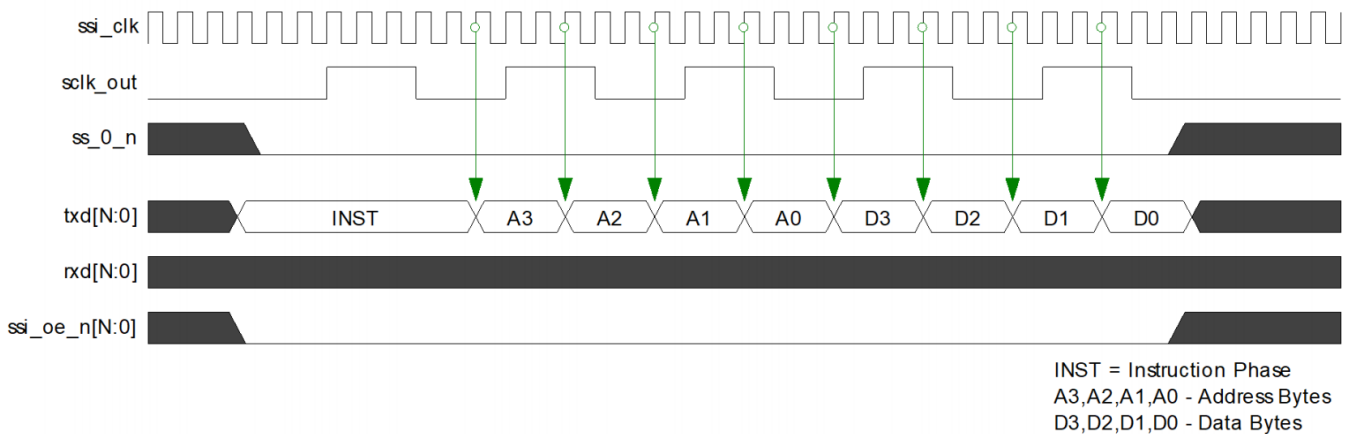


Figure 5-3-16 Transmit Data With DDR_DRIVE_EDGE = 2

5.3.3.4 Process of Reading Data in Internal DMA Mode

To read data in internal DMA mode, perform the following steps:

1. Program CTRLR0.TMOD = 1 (SPI write).
2. Program the CTRLR0.SPI_FRF register field with the frame format for SPI transfer.
3. Set transfer characteristics for SPI operation in the CTRLR0, BAUDR, TXFTLR.TXFTHR, and SPI_CTRLR0 registers.
4. Program CTRLR1.NDF with the amount of data to be transmitted to the device.
5. Enable DMA transfers using the DMACR.IDMAE register field.
6. Set the source transfer width in the DMACR.ATW field.

7. Set AXI transfer characteristics using the DMACR.AID, DMACR.AINC, DMACR.APROT, DMACR.ACACHE, and AXIARLEN.ARLN register fields.
8. Set Source address in the AXIAR0, and AXIAR1 registers.
9. Set device destination address in the SPIAR register.
10. Set instruction code in the SPIDR.SPI_INST register field.
11. Enable the controller using the SSIENR register.

5.3.3.5 Process of Writing Data in Internal DMA Mode

To write data in internal DMA mode, perform the following steps:

1. Program CTRLR0.TMOD = 1 (SPI write).
2. Program the CTRLR0.SPI_FRF register field with the frame format for SPI transfer.
3. Set transfer characteristics for SPI operation in the CTRLR0, BAUDR, TXFTLR.TXFTHR, and SPI_CTRLR0 registers.
4. Program CTRLR1.NDF with the amount of data to be transmitted to the device.
5. Enable DMA transfers using the DMACR.IDMAE register field.
6. Set the source transfer width in the DMACR.ATW field.
7. Set AXI transfer characteristics using the DMACR.AID, DMACR.AINC, DMACR.APROT, DMACR.ACACHE, and AXIARLEN.ARLN register fields.
8. Set Source address in the AXIAR0, and AXIAR1 registers.
9. Set device destination address in the SPIAR register.
10. Set instruction code in the SPIDR.SPI_INST register field.
11. Enable the controller using the SSIENR register.

5.3.3.6 Write Operation in Enhanced SPI Modes

Dual, Quad, or Octal SPI write operations can be divided into three parts:

- Instruction phase
- Address phase
- Data phase The following register fields are used for a write operation:
 - CTRLR0.SPI_FRF - Specifies the format in which the transmission happens for the frame.
 - SPI_CTRLR0 (SPI Control Register 0 register) – Specifies length of instruction, address, and data.
 - SPI_CTRLR0.INST_L – Specifies length of an instruction (possible values for instruction length are 0, 4, 8, or 16 bits.)
 - SPI_CTRLR0.ADDR_L – Specifies address length
 - CTRLR0.DFS – Specifies data length. An instruction takes one FIFO location and address can take more than one FIFO locations. Both the instruction and address must be programmed in the data register (DR). DWC_ssi waits until both have been programmed to start the write operation. The instruction, address and data can be programmed to send in dual mode/quad mode/octal mode, which can be selected from the SPI_CTRLR0.TRANS_TYPE and CTRLR0.SPI_FRF fields.

Note:

- If CTRLR0.SPI_FRF is selected to be Standard SPI Format, everything is sent in Standard SPI mode and SPI_CTRLR0.TRANS_TYPE field is ignored.
- CTRLR0.SPI_FRF is only applicable if CTRLR0.FRF is programmed to 00.

5.3.3.7 Read Operation in Enhanced SPI Modes

A Dual, Quad, or Octal SPI read operation can be divided into four phases:

- Instruction phase

- Address phase
- Wait cycles
- Data phase Wait Cycles can be programmed using SPIC_CTRLR0.WAIT_CYCLES field. The wait cycles are introduced for target slave to change their mode from input to output and the wait cycles can vary for different devices. For a READ operation, DWC_ssi sends instruction and control data once and waits until it receives NDF (CTRLR1 register) number of data frames and then de-asserts slave select signal.

5.3.4 FMC Register

This chapter details all possible registers in the controller. They are arranged hierarchically into maps and blocks (banks).

5.3.4.1 Register Summary

Table 4-1 describes the registers.

Offset Address	Register	Description
0x0	CTRLR0	Control Register 0
0x4	CTRLR1	Control Register 1
0x8	SSIENR	SSI Enable Register
0xc	MWCR	Microwire Control Register
0x10	SER	Slave Enable Register
0x14	BAUDR	Baud Rate Select
0x18	TXFTLR	Transmit FIFO Threshold Level
0x1c	RXFTLR	Receive FIFO Threshold Level
0x20	TXFLR	Transmit FIFO Level Register
0x24	RXFLR	Receive FIFO Level Register
0x28	SR	Status Register
0x2c	IMR	Interrupt Mask Register
0x30	ISR	Interrupt Status Register
0x34	RISR	Raw Interrupt Status Register
0x38	TXEICR	Transmit FIFO Error Interrupt Clear Registers
0x3c	RXOICR	Receive FIFO Overflow Interrupt Clear Register
0x40	RXUICR	Receive FIFO Underflow Interrupt Clear Register
0x44	MSTICR	Multi-Master Interrupt Clear Register
0x48	ICR	Interrupt Clear Register
0x4c	DMACR	DMA Control Register
0x50	DMATDLR	DMA Transmit Data Level
0x50	AXIAWLEN	Destination Burst Length
0x54	DMARDLR	DMA Receive Data Level

Offset Address	Register	Description
0x54	AXIARLEN	Source Burst Length
0x58	IDR	Identification Register
0x5c	SSIC_VERSION_ID	Component version Register
0x60+i*0x4	DRx(for i = 0; i <= 35)	Data Register
0x68	SSI_CTRL	Control Register
0xf0	RX_SAMPLE_DELAY	RX Sample Delay Register
0xf4	SPI_CTRLR0	SPI_CTRLR0 - SPI Control Register
0xf8	DDR_DRIVE_EDGE	DDR_DRIVE_EDGE - Transmit Drive Edge Register
0xfc	XIP_MODE_BITS	eXecute in Place - Mode bits
0x100	XIP_INCR_INST	XIP_INCR_INST - XIP INCR transfer opcode
0x104	XIP_WRAP_INST	XIP_WRAP_INST - XIP WRAP transfer opcode
0x108	XIP_CTRL	XIP_CTRL - XIP Control Register
0x10c	XIP_SER	Slave Enable Register
0x110	XR XOICR	XIP Receive FIFO Overflow Interrupt Clear Register
0x114	XIP_CNT_TIME_OUT	XIP time out register for continuous transfers
0x118	SPI_CTRLR1	SPI Control 1 register
0x11c	SPITECR	SPI Transmit Error Interrupt Clear Register
0x120	SPIDR	SPI Device Register
0x124	SPIAR	SPI Device Address Register
0x128	AXIAR0	AXI Address Register 0
0x12c	AXIAR1	AXI Address Register 1
0x130	AXIECR	AXI Master Error Interrupt Clear Register
0x134	DONECR	Transfer Done Clear Interrupt Clear Register
0x140	XIP_WRITE_INCR_INST	XIP_WRITE_INCR_INST - XIP Write INCR transfer opcode
0x144	XIP_WRITE_WRAP_INST	XIP_WRITE_WRAP_INST - XIP Write WRAP transfer opcode
0x148	XIP_WRITE_CTRL	XIP_WRITE_CTRL - XIP Write Control Register

5.3.4.2 Register Descriptions

CTRLR0

CTRLR0 is a Control Register 0 Offset Address: 0x0 Total Reset Value:0x00004007

Bits	Access	Name	Description	Reset
31	* Varies	SSI_IS_MST	This field selects if SPI is working in Master or Slave mode Values: - 0x1 (MASTER) - 0x0 (SLAVE)	0x0

Bits	Access	Name	Description	Reset
30:26	R	RSVD_CTRLR0_26_31	Reserved bits - read as zero	0x0
25	* Varies	SPI_DWS_EN	Enable Dynamic wait states in SPI mode of operation. Values: - 0x0 (DISABLE): Disable SPI Dynamic Wait State - 0x1 (ENABLE): Enable SPI Dynamic Wait State	0x0
24	* Varies	SPI_HYPERBUS_EN	SPI Hyperbus Frame format enable: Values: 0x0 (DISABLE): Disable Hyperbus Format 0x1 (ENABLE): Enable Hyperbus Format	0x0
23:22	* Varies	SPI_FRF	SPI Frame Format: Values: - 0x0 (SPI_STANDARD): Standard SPI Format - 0x1 (SPI_DUAL): Dual SPI Format - 0x2 (SPI_QUAD): Quad SPI Format - 0x3 (SPI_OCTAL): Octal SPI Format	0x0
21:20	R	RSVD_CTRLR0_20_21	Reserved bits - read as zero	0x0
19:16	R/W	CFS	Control Frame Size.	0x0
15	R	RSVD_CTRLR0_15	Reserved bits - read as zero	0x0
14	R/W	SSTE	Slave Select Toggle Enable. Values: - 0x1 (TOGGLE_EN): ss_n line will toggle between consecutive data frames, with the serial clock (sclk) being held to its default value while ssn is high <i>0x0 (TOGGLE_DISABLE): ss*_n will stay low and sclk will run continuously for the duration of the transfer</i>	0x1
13	* Varies	SRL	Shift Register Loop. Used for testing purposes only. Values: 0x1 (TESTING_MODE): Test Mode Operation - 0x0 (NORMAL_MODE): Normal mode operation	0x0
12	R/W	SLV_OE	Slave Output Enable. Values: - 0x1 (DISABLED): Slave Output is disabled -0x0 (ENABLED): Slave Output is enabled	0x0
11:10	R/W	TMOD	Transfer Mode. Values: - 0x0 (TX_AND_RX): Transmit & Receive; Not Applicable in enhanced SPI operating mode or when SSIC_HAS_TX_RX_EN is set to 0 - 0x1 (TX_ONLY): Transmit only mode; Or Write in enhanced SPI operating mode - 0x2 (RX_ONLY): Receive only mode; Or Read in enhanced SPI operating mode - 0x3 (EEPROM_READ): EEPROM Read mode; Not Applicable in enhanced SPI operating mode	0x0

Bits	Access	Name	Description	Reset
9	* Varies	SCPOL	Serial Clock Polarity. Values: - 0x0 (INACTIVE_HIGH): Inactive state of serial clock is low - 0x1 (INACTIVE_LOW): Inactive state of serial clock is high	0x0
8	* Varies	SCPH	Serial Clock Phase. Values: - 0x1 (START_BIT): Serial clock toggles at start of first bit - 0x0 (MIDDLE_BIT): Serial clock toggles in middle of first bit	0x0
7:6	* Varies	FRF	Frame Format: 0x0 (SPI): Motorola SPI Frame Format	0x0
5	R	RSVD_CTRLR0_5	Reserved bits - read as zero	0x0
4:0	R/W	DFS	Data Frame Size.Note: When SSIC_SPI_MODE is set to "Dual", "Quad" or "Octal" mode and SPI_FRF is not set to 2'b00: - DFS value must be a multiple of 2 if SPI_FRF = 01 - DFS value must be multiple of 4 if SPI_FRF = 10 - DFS value must be multiple of 8 if SPI_FRF = 11	0x7

CTRLR1

CTRLR1 is a Control Register 1 Offset Address: 0x4 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:16	R	RSVD_CTRLR1	Reserved bits - read as zero	0x0
15:0	R/W	NDF	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. When SPI_CTRLR0.CLK_STRETCH_EN=1 and TMOD = 01, this register field sets the number of data frames to be continuously transmitted by SPI. If the Transmit FIFO goes empty in-between, SPI masks the serial clock (sclk_out) and wait for rest of the data until the programmed amount of frames are transferred successfully.When the SPI is configured as a serial slave, the transfer continues for as long as the slave is selected. Therefore, this register serves no purpose and is not present when the SPI is configured as a serial slave.	0x0

SSIENR

SSIENR is a SSI Enable Register Offset Address: 0x8 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:1	R	RSVD_SSIENR	Reserved bits - read as zero	0x0

Bits	Access	Name	Description	Reset
0	R/W	SSIC_EN	SSI Enable.Enables and disables all FMC 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 FMC control registers when enabled. When disabled, the ssi sleep output is set (after delay) to inform the system that it is safe to remove the ssi_clk, thus saving power consumption in the system. Values: - 0x1 (ENABLED): Enables FMC - 0x0 (DISABLE): Disables FMC	0x0

MWCR

MWCR is a Microwire Control Register Offset Address: 0xc Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:3	R	RSVD_MWCR	Reserved bits - read as zero	0x0
2	* Varies	MHS	Microwire Handshaking. Values: - 0x1 (ENABLED): handshaking interface is enabled - 0x0 (DISABLE): handshaking interface is disabled	0x0
1	R/W	MDD	Microwire Control. Values: - 0x1 (TRANSMIT): SSI transmits data - 0x0 (RECEIVE): SSI receives data	0x0
0	R/W	MWMOD	Microwire Transfer Mode. Values: - 0x1 (SEQUENTIAL): Sequential Transfer - 0x0 (NON_SEQUENTIAL): Non-Sequential Transfer	0x0

SER

SER is a Slave Enable Register Offset Address: 0x10 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:y	R	RSVD_SER	Reserved bits - read as zero	0x0
x:0	R/W	SER	Slave Select Enable Flag.Each bit in this register corresponds to a slave select line (ss_x_n) from the controller. When a bit in this register is set (1), the corresponding slave select line from the master is activated when a serial transfer begins. It should be noted that setting or clearing bits in this register have no effect on the corresponding slave select outputs until a Before beginning a transfer, you should enable the bit in this register that corresponds to the slave device with which the master wants to communicate. When not operating in broadcast mode, only one bit in this field should be set.	0x0

BAUDR

BAUDR is a Baud Rate Select Offset Address: 0x14 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:16	R	RSVD_BAUDR_16_31	Reserved bits - read as zero	0x0
15:1	R/W	SCKDV	SSI Clock Divider.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: $F_{sclk\_out} = F_{ssi\_clk} / BAUDR$ where BAUDR is any even value between 2 and 65534 ($BAUDR = \{SCKDV * 2\}$). For example: for $F_{ssi\_clk} = 3.6864\text{MHz}$ and $BAUDR = 2$ $F_{sclk\_out} = 3.6864 / 2 = 1.8432\text{MHz}$	SSIC_DFLT_BAUDR/2
0	R	RSVD_BAUDR_0	Reserved bits - read as zero	0x0

TXFTLR

TXFTLR is a Transmit FIFO Threshold Level Offset Address: 0x18 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:y	R	RSVD_TXFTHR	Reserved bits - read as zero	0x0
x:16	* Varies	TXFTHR	Transfer start FIFO level.Used to control the level of entries in transmit FIFO above which transfer will start on serial line. This register can be used to ensure that sufficient data is present in transmit FIFO before starting a write operation on serial line. In Internal DMA mode, this field sets the minimum amount of data frames present in the FIFO after which the controller starts the transfer.	0x0
15:y	R	RSVD_TXFTLR	Reserved bits - read as zero	0x0
x:0	R/W	TFT	Transmit FIFO Threshold.Controls the level of entries (or below) at which the transmit FIFO controller triggers an interrupt. The FIFO depth is configurable in the range 8-256; this register is sized to the number of address bits needed to access the FIFO. 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

RXFTLR

RXFTLR is a Receive FIFO Threshold Level Offset Address: 0x1c Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:y	R	RSVD_RXFTLR	Reserved bits - read as zero	0x0

Bits	Access	Name	Description	Reset
x:0	R/W	RFT	Receive FIFO Threshold.Controls the level of entries (or above) at which the receive FIFO controller triggers an interrupt. The FIFO depth is configurable in the range 8-256. This register is sized to the number of address bits needed to access the FIFO. 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

TXFLR

TXFLR is a Transmit FIFO Level Register Offset Address: 0x20 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:y	R	RSVD_TXFLR	Reserved bits - read as zero	0x0
x:0	R	TXTFL	Transmit FIFO Level.Contains the number of valid data entries in the transmit FIFO.	0x0

RXFLR

RXFLR is a Receive FIFO Level Register

Offset Address: 0x24 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:y	R	RSVD_RXFLR	Reserved bits - read as zero	0x0
x:0	R	RXTFL	Receive FIFO Level.Contains the number of valid data entries in the receive FIFO.	0x0

SR

SR is a Status Register Offset Address: 0x28 Total Reset Value:0x00000006

Bits	Name	Access	Description	Reset
31:15	R	CMPLTD_DF	Completed Data frames This field indicates total data frames transferred in the previous internal DMA transfer	0x0
14:7	R	RSVD_SR *	Reserved bits - read as zero	0x0
6	R	DCOL	Data Collision Error.This bit will be set if ss_in_n input is asserted by other master, when the controller is in the middle of the transfer. This informs the processor that the last data transfer was halted before completion. This bit is cleared when read. Values: - 0x1 (TX_COLLISION_ERROR): Transmit Data Collision Error - 0x0 (NO_ERROR_CONDITION): No Error	0x0
5	R	TXE	Transmission Error. Values: - 0x1 (TX_ERROR): Transmission Error - 0x0 (NO_ERROR): No Error	0x0

Bits	Name	Access	Description	Reset
4	R	RFF	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. Values: - 0x1 (FULL): Receive FIFO is full - 0x0 (NOT_FULL): Receive FIFO is not full	0x0
3	R	RFNE	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. Values: - 0x1 (NOT_EMPTY): Receive FIFO is not empty - 0x0 (EMPTY): Receive FIFO is empty	0x0
2	R	TFE	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. Values: - 0x1 (EMPTY): Transmit FIFO is empty - 0x0 (NOT_EMPTY): Transmit FIFO is not empty	0x1
1	R	TFNF	Transmit FIFO Not Full. Set when the transmit FIFO contains one or more empty locations, and is cleared when the FIFO is full. Values: - 0x1 (NOT_FULL): Tx FIFO is not Full - 0x0 (FULL): Tx FIFO is full	0x1
0	R	BUSY	SSI Busy Flag. When set, indicates that a serial transfer is in progress; when cleared indicates that the controller is idle or disabled. Values: - 0x1 (ACTIVE): FMC is actively transferring data - 0x0 (INACTIVE): FMC is idle or disabled	0x0

IMR

IMR is an Interrupt Mask Register Offset Address: 0x2c Total Reset Value: 0x0000003f

Bits	Access	Name	Description	Reset
31:12	R	RSVD_12_32_IMR	Reserved bits - read as zero	0x0
11	* Varies	DONEM	SSI Done Interrupt Mask Values: - 0x1 (UNMASKED): ssi_done_intr interrupt is not masked - 0x0 (MASKED): ssi_done_intr interrupt is masked	0x0
10	* Varies	SPITEM	SPI Transmit Error Interrupt Mask Values: - 0x1 (UNMASKED): ssi_spite_intr interrupt is not masked - 0x0 (MASKED): ssi_spite_intr interrupt is masked	0x0
9	R	RSVD_9_IMR	Reserved bits - read as zero	0x0

Bits	Access	Name	Description	Reset
8	* Varies	AXIEM	AXI Error Interrupt Mask Values: - 0x1 (UNMASKED): ssi_axie_intr interrupt is not masked - 0x0 (MASKED): ssi_axie_intr interrupt is masked	0x0
7	* Varies	TXUIM	Transmit FIFO Underflow Interrupt Mask Values: - 0x1 (UNMASKED): ssi_txu_intr interrupt is not masked - 0x0 (MASKED): ssi_txu_intr interrupt is masked	0x0
6	* Varies	XR XOIM	XIP Receive FIFO Overflow Interrupt Mask Values: - 0x1 (UNMASKED): ssi_xrxo_intr interrupt is not masked - 0x0 (MASKED): ssi_xrxo_intr interrupt is masked	0x0
5	* Varies	MSTIM	Multi-Master Contention Interrupt Mask. This bit field is not present if the controller is configured as a serial-master device Values: - 0x1 (UNMASKED): ssi_mst_intr interrupt is not masked - 0x0 (MASKED): ssi_mst_intr interrupt is masked	0x1
4	R/W	RXFIM	Receive FIFO Full Interrupt Mask Values: - 0x1 (UNMASKED): ssi_rxf_intr interrupt is not masked - 0x0 (MASKED): ssi_rxf_intr interrupt is masked	0x1
3	R/W	RXOIM	Receive FIFO Overflow Interrupt Mask Values: - 0x1 (UNMASKED): ssi_rxo_intr interrupt is not masked - 0x0 (MASKED): ssi_rxo_intr interrupt is masked	0x1
2	R/W	RXUIM	Receive FIFO Underflow Interrupt Mask Values: 0x1 (UNMASKED): ssi_rxu_intr interrupt is not masked 0x0 (MASKED): ssi_rxu_intr interrupt is masked	0x1
1	R/W	TXOIM	Transmit FIFO Overflow Interrupt Mask Values: - 0x1 (UNMASKED): ssi_txo_intr interrupt is not masked - 0x0 (MASKED): ssi_txo_intr interrupt is masked	0x1
0	R/W	TXEIM	Transmit FIFO Empty Interrupt Mask Values: - 0x1 (UNMASKED): ssi_txe_intr interrupt is not masked - 0x0 (MASKED): ssi_txe_intr interrupt is masked	0x1

ISR

ISR is an Interrupt Status Register

Offset Address: 0x30 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:12	R	RSVD_12_32_RISR	Reserved bits - read as zero	0x0

Bits	Access	Name	Description	Reset
11	R	DONES	SSI Done Interrupt Status Values: - 0x1 (ACTIVE): ssi_done_intr interrupt is active after masking - 0x0 (INACTIVE): ssi_done_intr interrupt is not active after masking	0x0
10	R	SPITES	SPI Transmit Error Interrupt Values: - 0x1 (ACTIVE): ssi_spite_intr interrupt is active after masking - 0x0 (INACTIVE): ssi_spite_intr interrupt is not active after masking	0x0
9	R	RSVD_9_RISR	Reserved bits - read as zero	0x0
8	R	AXIES	AXI Error Interrupt Status Values: - 0x1 (ACTIVE): ssi_axie_intr interrupt is active after masking - 0x0 (INACTIVE): ssi_axie_intr interrupt is not active after masking	0x0
7	R	TXUIS	Transmit FIFO Underflow Interrupt Status Values: - 0x1 (ACTIVE): ssi_txu_intr interrupt is active after masking - 0x0 (INACTIVE): ssi_txu_intr interrupt is not active after masking	0x0
6	R	XR XOIS	XIP Receive FIFO Overflow Interrupt Status Values: - 0x1 (ACTIVE): ssi_xrxo_intr interrupt is active after masking - 0x0 (INACTIVE): ssi_xrxo_intr interrupt is not active after masking	0x0
5	R	MSTIS	Multi-Master Contention Interrupt Status. This bit field is not present if the controller is configured as a serial-slave device. Values: - 0x1 (ACTIVE): ssi_mst_intr interrupt is active after masking - 0x0 (INACTIVE): ssi_mst_intr interrupt is not active after masking	0x0
4	R	RXFIS	Receive FIFO Full Interrupt Status Values: - 0x1 (ACTIVE): ssi_rxf_intr interrupt is active after masking - 0x0 (INACTIVE): ssi_rxf_intr interrupt is not active after masking	0x0
3	R	RXOIS	Receive FIFO Overflow Interrupt Status Values: - 0x1 (ACTIVE): ssi_rxo_intr interrupt is active after masking - 0x0 (INACTIVE): ssi_rxo_intr interrupt is not active after masking	0x0

Bits	Access	Name	Description	Reset
2	R	RXUIS	Receive FIFO Underflow Interrupt Status Values: - 0x1 (ACTIVE): ssi_rxu_intr interrupt is active after masking - 0x0 (INACTIVE): ssi_rxu_intr interrupt is not active after masking	0x0
1	R	TXOIS	Transmit FIFO Overflow Interrupt Status Values: - 0x1 (ACTIVE): ssi_txo_intr interrupt is active after masking - 0x0 (INACTIVE): ssi_txo_intr interrupt is not active after masking	0x0
0	R	TXEIS	Transmit FIFO Empty Interrupt Status Values: - 0x1 (ACTIVE): ssi_txe_intr interrupt is active after masking - 0x0 (INACTIVE): ssi_txe_intr interrupt is not active after masking	0x0

RISR

RISR is a Raw Interrupt Status Register

Offset Address: 0x34 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:12	R	RSVD_12_32_RISR	Reserved bits - read as zero	0x0
11	R	DONER	SSI Done Interrupt Raw Status Values: - 0x1 (ACTIVE): ssi_done_intr interrupt is active prior to masking - 0x0 (INACTIVE): ssi_done_intr interrupt is not active prior to masking	0x0
10	R	SPITER	SPI Transmit Error Interrupt status.This bit gets set, If SPI Master fails to get a READY status from the slave until the amount of time defined in SPI_CTRLR1.MAX_WS field, then it will stop the SPI transfer and the FIFO is flushed (in case of write operation). Values: - 0x1 (ACTIVE): ssi_spite_intr interrupt is active prior to masking - 0x0 (INACTIVE): ssi_spite_intr interrupt is not active prior to masking 0x0	
9	R	RSVD_9_RISR	Reserved bits - read as zero	0x0
8	R	AXIER	AXI Error Interrupt Raw Status Values: -0x1 (ACTIVE): ssi_axie_intr interrupt is active prior to masking - 0x0 (INACTIVE): ssi_axie_intr interrupt is not active prior to masking	0x0

Bits	Access	Name	Description	Reset
7	R	TXUIR	Transmit FIFO Underflow Interrupt Raw Status Values: - 0x1 (ACTIVE): ssi_txu_intr interrupt is active prior to masking - 0x0 (INACTIVE): ssi_txu_intr interrupt is not active prior masking	0x0
6	R	XR XOIR	XIP Receive FIFO Overflow Raw Interrupt Status Values: - 0x1 (ACTIVE): ssi_xrxo_intr interrupt is active prior to masking - 0x0 (INACTIVE): ssi_xrxo_intr interrupt is not active prior masking	0x0
5	R	MSTIR	Multi-Master Contention Raw Interrupt Status. Values: - 0x1 (ACTIVE): ssi_mst_intr interrupt is active prior to masking - 0x0 (INACTIVE): ssi_mst_intr interrupt is not active prior masking	0x0
4	R	RXFIR	Receive FIFO Full Raw Interrupt Status Values: - 0x1 (ACTIVE): ssi_rxf_intr interrupt is active prior to masking - 0x0 (INACTIVE): ssi_rxf_intr interrupt is not active prior masking	0x0
3	R	RXOIR	Receive FIFO Overflow Raw Interrupt Status Values: - 0x1 (ACTIVE): ssi_rxo_intr interrupt is active prior to masking - 0x0 (INACTIVE): ssi_rxo_intr interrupt is not active prior masking	0x0
2	R	RXUIR	Receive FIFO Underflow Raw Interrupt Status Values: - 0x1 (ACTIVE): ssi_rxu_intr interrupt is active prior to masking - 0x0 (INACTIVE): ssi_rxu_intr interrupt is not active prior masking	0x0
1	R	TXOIR	Transmit FIFO Overflow Raw Interrupt Status Values: - 0x1 (ACTIVE): ssi_txo_intr interrupt is active prior to masking - 0x0 (INACTIVE): ssi_txo_intr interrupt is not active prior masking	0x0
0	R	TXEIR	Transmit FIFO Empty Raw Interrupt Status Values: - 0x1 (ACTIVE): ssi_txe_intr interrupt is active prior to masking - 0x0 (INACTIVE): ssi_txe_intr interrupt is not active prior masking	0x0

TXEICR

TXEICR is a Transmit FIFO Error Interrupt Clear Registers Offset Address: 0x38 Total Reset Value:0x0

Bits	Access	Name	Description	Reset

Bits	Access	Name	Description	Reset
31:1	R	RSVD_TXEICR	Reserved bits - read as zero	0x0
0	RC	TXEICR	Clear Transmit FIFO Overflow/Underflow Interrupt. This register reflects the status of the interrupt. A read from this register clears the ssi_txo_intr/ssi_txu_intr interrupt; writing has no effect.	0x0

RXOICR

RXOICR is a Receive FIFO Overflow Interrupt Clear Register Offset Address: 0x3c Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:1	R	RSVD_RXOICR	Reserved bits - read as zero	0x0
0	RC	RXOICR	Clear Receive FIFO Overflow Interrupt. This register reflects the status of the interrupt. A read from this register clears the ssi_rxo_intr interrupt; writing has no effect.	0x0

RXUICR

RXUICR is a Receive FIFO Underflow Interrupt Clear Register Offset Address: 0x40 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:1	R	RSVD_RXUICR	Reserved bits - read as zero	0x0
0	RC	RXUICR	Clear Receive FIFO Underflow Interrupt. This register reflects the status of the interrupt. A read from this register clears the ssi_rxu_intr interrupt; writing has no effect.	0x0

MSTICR

MSTICR is a Multi-Master Interrupt Clear Register Offset Address: 0x44 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:1	R	RSVD_MSTICR	Reserved bits - read as zero	0x0
0	RC	MSTICR	Clear Multi-Master Contention Interrupt. This register reflects the status of the interrupt. A read from this register clears the ssi_mst_intr interrupt; writing has no effect.	0x0

ICR

ICR is an Interrupt Clear Register Offset Address: 0x48 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:1	R	RSVD_ICR	Reserved bits - read as zero	0x0
0	RC	ICR	Clear Interrupts. This register is set if any of the interrupts below are active. A read clears the ssi_txo_intr, ssi_rxu_intr, ssi_rxo_intr, and the ssi_mst_intr interrupts. Writing to this register has no effect.	0x0

DMACR

DMACR is a DMA Control Register Offset Address: 0x4c Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:y	R	RSVD_DMCCR	Reserved bits - read as zero	0x0
x:15	* Varies	AID	AXI awid/arid signal value.	0x0
14:12	* Varies	APROT	AXI arprot/awprot signal value.	0x0
11:8	* Varies	ACACHE	AXI arcache/awcache signal value.	0x0
7	R	RSVD_DMCCR7	Reserved bits - read as zero	0x0
6	* Varies	AINC	Address Increment. Indicates whether to increment the AXI address on every transfer. 1 = Increment 0 = No Change Note: Increment aligns the address to the next DMCCR.ATW boundary	0x0
5	R	RSVD_DMCCR5	Reserved bits - read as zero	0x0
4:3	* Varies	ATW	AXI transfer width for DMA transfers mapped to arsize/awsize. This value must be less than or equal to SSIC_AXI_DW. Values: 0x0: 1 byte 0x1: 2 bytes 0x2: 4 bytes 0x3: 8 bytesNote: When SSIC_AXI_DW is set to 32 bits, if user programs this field to 0x8(3 bytes). The controller will use 4 bytes as transfer size for the AXI transfers.	0x0
4:3	* Varies	ATW	AXI transfer width for DMA transfers mapped to arsize/awsize. This value must be less than or equal to SSIC_AXI_DW. Values: 0x0: 1 byte 0x1: 2 bytes 0x2: 4 bytes 0x3: 8 bytes Note: When SSIC_AXI_DW is set to 32 bits, if user programs this field to 0x8(3 bytes). The controller will use 4 bytes as transfer size for the AXI transfers.	0x0
2	* Varies	IDMAE	Internal DMA Enable. This bit should be enabled only when CTRLR0.FRF = 0 (Motorola SPI) and CTRLR0.SPI_FRF > 0.	0x0
1	* Varies	TDMAE	Transmit DMA Enable. This bit enables/disables the transmit FIFO DMA channel. Values: - 0x1 (ENABLED): Transmit DMA enabled - 0x0 (DISABLE): Transmit DMA disabled	0x0
0	* Varies	RDMAE	Receive DMA Enable. This bit enables/disables the receive FIFO DMA channel. Values: - 0x1 (ENABLED): Receive DMA enabled - 0x0 (DISABLE): Receive DMA disabled	0x0

DMATDLR is a DMA Transmit Data Level Register Offset Address: 0x50 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:y	R	RSVD_DMATDLR	Reserved bits - read as zero	0x0
x:0	R/W	DMATDL	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

AXIAWLEN

AXIAWLEN is a Destination Burst Length Register Offset Address: 0x50 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:y	R	RSVD_AXIAWLEN2	Reserved bits - read as zero	0x0
x:8	R/W	AWLEN	Destination Burst Length	SSIC_DFLT_AWLEN
7:0	R	RSVD_AXIAWLEN	Reserved bits - read as zero	0x0

DMARDLR

DMARDLR is a DMA Receive Data Level Register Offset Address: 0x54 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:y	R	RSVD_DMARDLR	Reserved bits - read as zero	0x0
x:0	R/W	DMARDL	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

AXIARLEN

AXIARLEN is an Source Burst Length Register Offset Address: 0x54 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:y	R	RSVD_AXIARLEN2	Reserved bits - read as zero	0x0
x:8	R/W	ARLEN	Source Burst Length	SSIC_DFLT_ARLEN
7:0	R	RSVD_AXIARLEN	Reserved bits - read as zero	0x0

IDR

IDR is an Identification Register Offset Address: 0x58 Total Reset Value:0xha1b2c3d5

Bits	Access	Name	Description	Reset
31:0	R	IDCODE	Identification code. The register contains the peripheral's identification code, which is written into the register at configuration time using CoreConsultant.	0xha1b2c3d5

SSIC_VERSION_ID

SSIC_VERSION_ID is a component version register Offset Address: 0x5c Total Reset Value:0x3130332a

Bits	Access	Name	Description	Reset
31:0	R	SSIC_COMP_VERSION	Contains the hex representation of the Synopsys component version. Consists of ASCII value for each number in the version, followed by . <i>For example 31_30_33_2A represents the version 1.03.</i>	0x3130332a

DRx(for i = 0; i <= 35)

DRx is a Data Register Offset Address:0x60 +i*0x4 Total Reset Value:0x0

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

SSI_CTRL

SSI_CTRL is a control Register Offset Address:0x68 Total Reset Value:0x00004000

Bits	Access	Name	Description	Reset
17:13	R/W	rxds_delay_num	rxds delay line number bit[0]:0:negedge 1:posedge bit[3:1]:delay number	0x2
17:13	RO	ssi2_spi_mode		0x0
10	RO	ssi2_ssi_sleep		0x0
9:8	RO	ssi1_spi_mode		0x0
7	RO	ssi1_ssi_sleep		0x0
6:5	RO	ssi0_spi_mode		0x0
4	RO	ssi0_ssi_sleep		0x0
3:2	R		Reserved bits - read as zero	0x0
0	R/W	ssi0_xip_en		0x0

RX_SAMPLE_DELAY

RX_SAMPLE_DELAY is a RX Sample Delay Register Offset Address: 0xf0 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:17	R	RSVD1_RX_SAMPLE_DLY	Reserved bits - read as zero	0x0

Bits	Access	Name	Description	Reset
16	* Varies	SE	Receive Data (rx) Sampling Edge. This register is used to decide the sampling edge for RXD signal with ssi_clk. Then this bit is set to 1 then negative edge of ssi_clk will be used to sample the incoming data, otherwise positive edge will be used for sampling.	0x0
15:8	R	RSVD0_RX_SAMPLE_DLY	Reserved bits - read as zero	0x0
7:0	R/W	RSD	Receive Data (rx) Sample Delay. This register is used to delay the sample of the rx input port. Each value represents a single ssi_clk delay on the sample of rx. Note; If this register is programmed with a value that exceeds the depth of the internal shift registers (SSIC_RX_DLY_SR_DEPTH) zero delay will be applied to the rx sample.	0x0

SPI_CTRLR0

SPI_CTRLR0 is a SPI Control Register Offset Address: 0xf4 Total Reset Value:0x28000200

Bits	Access	Name	Description	Reset
31	R	RSVD_SPI_CTRLR0	Reserved bits - read as zero	0x0
30	* Varies	CLK_STRETCH_EN	Enables clock stretching capability in SPI transfers. In case of write, if the FIFO becomes empty the controller will stretch the clock until FIFO has enough data to continue the transfer. In case of read, if the receive FIFO becomes full the controller will stop the clock until data has been read from the FIFO.	0x0
29	* Varies	XIP_PREFETCH_EN	Enables XIP pre-fetch functionality in the controller. Once enabled the controller will pre-fetch data frames from next contiguous location, to reduce the latency for the upcoming contiguous transfer. If the next XIP request is not contiguous then pre-fetched bits will be discarded.	0x1
28	R	RSVD_SPI_CTRLR0_28	Reserved bits - read as zero	0x0
27:26	* Varies	XIP_MBL	XIP Mode bits length. Sets the length of mode bits in XIP mode of operation. These bits are valid only when SPI_CTRLR0.XIP_MD_BIT_EN is set to 1. Values: - 0x0 (MBL_2): Mode bits length equal to 2- 0x1 (MBL_4): Mode bits length equal to 4- 0x2 (MBL_8): Mode bits length equal to 8- 0x3 (MBL_16): Mode bits length equal to 16	0x2

Bits	Access	Name	Description	Reset
25	* Varies	SPI_RXDS_SIG_EN	Enable rxds signaling during address and command phase of Hypebus transfer.This bit enables rxds signaling by Hyperbus slave devices during Command-Address (CA) phase. If the rxds signal is set to 1 during the CA phase of transfer, the controller transmits (2*SPI_CTRLR0.WAIT_CYCLES-1) wait cycles after the address phase is complete.	0x0
24	* Varies	SPI_DM_EN	SPI data mask enable bit.When this bit is enabled, the txd_dmsignal is used to mask the data on the txd data line. This bit is enabled only when the SSIC_DM_EN parameter is set to 1.	0x0
23:22	R	RSVD_SPI_CTRLR0_22_23	Reserved bits - read as zero	0x0
21	* Varies	SSIC_XIP_CONT_XFER_EN	Enable continuous transfer in XIP mode. If this bit is set to 1 then continuous transfer mode in XIP will be enabled, in this mode the controller will keep slave selected until a non-XIP transfer is detected on the AHB interface.	0x0
20	* Varies	XIP_INST_EN	XIP instruction enable bit. If this bit is set to 1 then XIP transfers will also have instruction phase. The instruction op-codes will be chosen from XIP_INCR_INST or XIP_WRAP_INST registers bases on AHB transfer type.	0x0
19	* Varies	XIP_DFS_HC	Fix DFS for XIP transfers. If this bit is set to 1 then data frame size for XIP transfers will be fixed to the programmed value in CTRLR0.DFS. The number of data frames to fetch will be determined by HSIZE and HBURST signals. If this bit is set to 0 then data frame size and number of data frames to fetch will be determined by HSIZE and HBURST signals	0x0
18	* Varies	SPI_RXDS_EN	Read data strobe enable bit. Once this bit is set to 1 the controller will use Read data strobe (rxds) to capture read data in DDR mode.	0x0
17	* Varies	INST_DDR_EN	Instruction DDR Enable bit. This will enable Dual-data rate transfer for Instruction phase.	0x0
16	* Varies	SPI_DDR_EN	SPI DDR Enable bit. This will enable Dual-data rate transfers in Dual/Quad/Octal frame formats of SPI.	0x0
15:11	R/W	WAIT_CYCLES	Wait cycles in Dual/Quad/Octal mode between control frames transmit and data reception. Specified as number of SPI clock cycles.	0x0
10	R	RSVD_SPI_CTRLR0_10	Reserved bits - read as zero	0x0

Bits	Access	Name	Description	Reset
9:8	R/W	INST_L	Dual/Quad/Octal mode instruction length in bits. Values: - 0x0 (INST_L0): No Instruction - 0x1 (INST_L4): 4 bit Instruction length - 0x2 (INST_L8): 8 bit Instruction length - 0x3 (INST_L16): 16 bit Instruction length	0x2
7	* Varies	XIP_MD_BIT_EN	Mode bits enable in XIP mode. If this bit is set to 1, then in XIP mode of operation the controller will insert mode bits after the address phase. These bits are set in register XIP_MODE_BITS register. The length of mode bits is always set to 8 bits.	0x0
6	R	RSVD_SPI_CTRLR0_6	Reserved bits - read as zero	0x0
5:2	R/W	ADDR_L	This bit defines Length of Address to be transmitted. Only after this much bits are programmed in to the FIFO the transfer can begin. Values: - 0x0 (ADDR_L0): No Address - 0x1 (ADDR_L4): 4 bit Address length - 0x2 (ADDR_L8): 8 bit Address length - 0x3 (ADDR_L12): 12 bit Address length - 0x4 (ADDR_L16): 16 bit Address length - 0x5 (ADDR_L20): 20 bit Address length - 0x6 (ADDR_L24): 24 bit Address length - 0x7 (ADDR_L28): 28 bit Address length - 0x8 (ADDR_L32): 32 bit Address length - 0x9 (ADDR_L36): 36 bit Address length - 0xa (ADDR_L40): 40 bit Address length - 0xb (ADDR_L44): 44 bit Address length - 0xc (ADDR_L48): 48 bit Address length - 0xd (ADDR_L52): 52 bit Address length - 0xe (ADDR_L56): 56 bit Address length - 0xf (ADDR_L60): 60 bit Address length	0x0
1:0	R/W	TRANS_TYPE	Address and instruction transfer format. Selects whether the controller will transmit instruction/address either in Standard SPI mode or the SPI mode selected in CTRLR0.SPI_FRF field. Values: - 0x0 (TT0): Instruction and Address will be sent in Standard SPI Mode. - 0x1 (TT1): Instruction will be sent in Standard SPI Mode and Address will be sent in the mode specified by CTRLR0.SPI_FRF. - 0x2 (TT2): Both Instruction and Address will be sent in the mode specified by SPI_FRF. - 0x3 (TT3): Reserved.	0x0

DDR_DRIVE_EDGE is a Transmit Drive Edge Register Offset Address: 0xf8 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:8	R	RSVD_DDR_DRIVE_EDGE	Reserved bits - read as zero	0x0
7:0	R/W	TDE	TXD Drive edge register which decided the driving edge of transmit data.The maximum value of this register is = (BAUDR/2) -1.	0x0

XIP_MODE_BITS

- Name: eXecute in Place - Mode bits Offset Address: 0xfc Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:16	R	RSVD_XIP_MD_BITS	Reserved bits - read as zero	0x0
15:0	R/W	XIP_MD_BITS	XIP mode bits to be sent after address phase of XIP transfer.	0x0

XIP_INCR_INST

XIP_INCR_INST is a XIP INCR transfer opcode Offset Address: 0x100 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:16	R	RSVD_INCR_INST	Reserved bits - read as zero	0x0
15:0	R/W	INCR_INST	XIP INCR transfer opcode. When SPI_CTRLR0.XIP_INST_EN bit is set to 1, the controller sends instruction for all XIP transfers, this register field stores the instruction op-code to be sent when an INCR type transfer is requested on AHB bus. The number of bits to be send in instruction phase is determined by SPI_CTRL0.INST_L field.	0x0

XIP_WRAP_INST

XIP_WRAP_INST is a XIP WRAP transfer opcode

Offset: 0x0

Bits	Access	Name	Description	Reset
31:16	R	RSVD_WRAP_INST	Reserved bits - read as zero	0x0
15:0	R/W	WRAP_INST	XIP WRAP transfer opcode. When SPI_CTRLR0.XIP_INST_EN bit is set to 1, the controller sends instruction for all XIP transfers, this register field stores the instruction op-code to be sent when an WRAP type transfer is requested on AHB bus. The number of bits to be send in instruction phase is determined by SPI_CTRL0.INST_L field.	0x0

XIP_CTRL

XIP_CTRL is a XIP Control Register

Offset Address: 0x108 Total Reset Value:0x08000401

Bits	Access	Name	Description	Reset
31:30	R	RSVD_XIP_CTRL	Reserved bits - read as zero	0x0
29	R/W	XIP_PREFETCH_EN	Enables XIP pre-fetch functionality in the controller. Once enabled the controller will pre-fetch data frames from next contiguous location, to reduce the latency for the upcoming contiguous transfer. If the next XIP request is not contiguous then pre-fetched bits will be discarded.	0x0
28	R	RSVD_XIP_CTRL_28	Reserved bits - read as zero	0x0
27:26	R/W	XIP_MBL	XIP Mode bits length. Sets the length of mode bits in XIP mode of operation. These bits are valid only when XIP_CTRL.XIP_MD_BIT_EN is set to 1. Values: - 0x0 (MBL_2): Mode bits length equal to 2 - 0x1 (MBL_4): Mode bits length equal to 4 - 0x2 (MBL_8): Mode bits length equal to 8 - 0x3 (MBL_16): Mode bits length equal to 16	0x2
25	* Varies	RXDS_SIG_EN	Enable rxds signaling during address and command phase of Hyperbus transfer.	0x0
24	* Varies	XIP_HYPERBUS_EN	SPI Hyperbus Frame format enable for XIP transfers. Selects if data frame format for XIP transfers is in Hyperbus mode.	0x0
23	* Varies	CONT_XFER_EN	Enable continuous transfer in XIP mode. If this bit is set to 1 then continuous transfer mode in XIP will be enabled, in this mode the controller will keep slave selected until a non-XIP transfer is detected on the AHB interface.	0x0
22	* Varies	INST_EN	XIP instruction enable bit. If this bit is set to 1 then XIP transfers will also have instruction phase. The instruction op-codes will be chosen from XIP_INCR_INST or XIP_WRAP_INST registers bases on AHB transfer type.	0x0
21	* Varies	RXDS_EN	Read data strobe enable bit. Once this bit is set to 1 the controller will use Read data strobe (rxds) to capture read data in DDR mode.	0x0
20	* Varies	INST_DDR_EN	Instruction DDR Enable bit. This will enable Dual-data rate transfer for Instruction phase.	0x0
19	* Varies	DDR_EN	SPI DDR Enable bit. This will enable Dual-data rate transfers in Dual/Quad/Octal frame formats of SPI.	0x0
18	* Varies	DFS_HC	Fix DFS for XIP transfers. If this bit is set to 1 then data frame size for XIP transfers will be fixed to the programmed value in CTRLR0.DFS. The number of data frames to fetch will be determined by HSIZE and HBURST signals. If this bit is set to 0 then data frame size and number of data frames to fetch will be determined by HSIZE and HBURST signals	0x0

Bits	Access	Name	Description	Reset
17:13	R/W	WAIT_CYCLES	Wait cycles in Dual/Quad/Octal mode between control frames transmit and data reception. Specified as number of SPI clock cycles.	0x0
12	R/W	MD_BITS_EN	Mode bits enable in XIP mode. If this bit is set to 1, then in XIP mode of operation the controller will insert mode bits after the address phase. These bits are set in register XIP_MODE_BITS register. The length of mode bits is always set to 8 bits.	0x0
11	R	RSVD_SPI_CTRLR0_11	Reserved bits - read as zero	0x0
10:9	R/W	INST_L	Dual/Quad/Octal mode instruction length in bits. Values: - 0x0 (INST_L0): No Instruction - 0x1 (INST_L4): 4 bit Instruction length - 0x2 (INST_L8): 8 bit Instruction length - 0x3 (INST_L16): 16 bit Instruction length	0x2
8	R	RSVD_XIP_CTRL_8	Reserved bits - read as zero	0x0
7:4	R/W	ADDR_L	This bit defines Length of Address to be transmitted. Only after this much bits are programmed in to the FIFO the transfer can begin. Values: - 0x0 (ADDR_L0): No Address - 0x1 (ADDR_L4): 4 bit Address length - 0x2 (ADDR_L8): 8 bit Address length - 0x3 (ADDR_L12): 12 bit Address length - 0x4 (ADDR_L16): 16 bit Address length - 0x5 (ADDR_L20): 20 bit Address length - 0x6 (ADDR_L24): 24 bit Address length - 0x7 (ADDR_L28): 28 bit Address length - 0x8 (ADDR_L32): 32 bit Address length - 0x9 (ADDR_L36): 36 bit Address length - 0xa (ADDR_L40): 40 bit Address length - 0xb (ADDR_L44): 44 bit Address length - 0xc (ADDR_L48): 48 bit Address length - 0xd (ADDR_L52): 52 bit Address length - 0xe (ADDR_L56): 56 bit Address length - 0xf (ADDR_L60): 60 bit Address length	0x0

Bits	Access	Name	Description	Reset
3:2	R/W	TRANS_TYPE	Address and instruction transfer format. Selects whether the controller will transmit instruction/address either in Standard SPI mode or the SPI mode selected in CTRLR0.SPI_FRF field. Values: - 0x0 (TT0): Instruction and Address will be sent in Standard SPI Mode. - 0x1 (TT1): Instruction will be sent in Standard SPI Mode and Address will be sent in the mode specified by XIP_CTRL.SPI_FRF. - 0x2 (TT2): Both Instruction and Address will be sent in the mode specified by XIP_CTRL.SPI_FRF. - 0x3 (TT3): Reserved.	0x0
1:0	R/W	FRF	SPI Frame Format Selects data frame format for Transmitting/Receiving the data. Values: - 0x0 (RSVD): Reserved - 0x1 (SPI_DUAL): Dual SPI Format - 0x2 (SPI_QUAD): Quad SPI Format - 0x3 (SPI_OCTAL): Octal SPI Format	0x1

XIP_SER

XIP_SER is a Slave Enable Register Offset Address: 0x10c Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:y	R	RSVD_SER	Reserved bits - read as zero	0x0
x:0	R/W		SER Slave Select Enable Flag. Each bit in this register corresponds to a slave select line (ss_x_n) from the controller. When a bit in this register is set (1), the corresponding slave select line from the master is activated when a XIP transfer begins. It should be noted that setting or clearing bits in this register have no effect on the corresponding slave select outputs until a XIP transfer is started. Before beginning a transfer, you should enable the bit in this register that corresponds to the slave device with which the master wants to communicate. When not operating in broadcast mode, only one bit in this field should be set.	0x0

XR XOICR

XR XOICR is a XIP Receive FIFO Overflow Interrupt Clear Register

Offset Address: 0x110 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:1	R	RSVD_XR XOICR	Reserved bits - read as zero	0x0
0	RC	XR XOICR	Clear XIP Receive FIFO Overflow Interrupt. This register reflects the status of the interrupt. A read from this register clears the ssi_xrxo_intr(_n) interrupt; writing has no effect.	0x0

XIP_CNT_TIME_OUT

XIP_CNT_TIME_OUT is an XIP time out register for continuous transfers Offset Address: 0x114 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:8	R	RSVD_XTOC	Reserved bits - read as zero	0x0
7:0	R/W	XTOC	XIP time out value in terms of hclk. Once slave is selected in continuous XIP mode this counter will be used to de-select the slave if there is no request for the time specified in the counter.	0x0

SPI_CTRLR1

SPI_CTRLR1 is a SPI Control 1 register Offset Address: 0x118 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:12	R	RSVD_SPI_CTRLR1_12_32	Reserved bits - Read Only	0x0
11:8	R/W	MAX_WS	Maximum wait cycles allowed per transaction.This field indicate, up to how many times SPI slave could insert the wait states. The internal counter is incremented every time when the controller checks for the status from the slave and receives wait response.	0x0
7:3	R	RSVD_SPI_CTRLR1_3_7	Reserved bits - Read Only	0x0
2:0	R/W	DYN_WS	SPI Dynamic Wait states field.This field is used to set the value for wait states which will be introduced when SPI slave sends BUSY status to the Master.The programmed value of wait States will be introduced before checking status again.Number of wait states = DYN_WS+1	0x0

SPITECR

SPITECR is a SPI Transmit Error Interrupt Clear Register Offset Address: 0x11c Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:1	R	RSVD_SPITECR	Reserved bits - Read Only	0x0
0	R	SPITECR	Clear SPI Transmit Error interrupt.This register will reflect the status of the interrupt.A read from this register clears the ssi_spite_intr interrupt. Writing to this register has no effect.	0x0

SPIDR

SPIDR is a SPI Device Register Offset Address: 0x120 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:16	R	RSVD_SPIDR	Reserved bits - read as zero	0x0
15:0	R/W	SPI_INST	SPI Instruction code. This instruction code will be used for SPI operation	0x0

SPIAR

SPIAR is a SPI Device Address Register
Offset Address: 0x124 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:0	R/W	SDAR	SPI Device Address Register.This address will be used during read/write on SPI interface for DMA transfer	0x0

AXIAR0

AXIAR0 is an AXI Address Register 0 Offset Address: 0x128 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:0	R/W	AXIAR_0_31	LSB for AXI address for DMA operation.	0x0

AXIAR1

AXIAR1 is an AXI Address Register 1 Offset Address: 0x12c Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:y	R	RSVD_AXIAR1	Reserved bits - read as zero	0x0
x:0	R/W	AXIAR_32_63	MSB for source address for DMA operation.	0x0

AXIECR

AXIECR is an AXI Master Error Interrupt Clear Register Offset Address: 0x130 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:1	R	RSVD_AXIECR	Reserved bit	0x0
0	RC	AXIECR	Clear AXI Error Interrupt. This register will reflect the status of the interrupt. A read from this register clears the ssi_axie_intr interrupt. Writing to this register has no effect.	0x0

DONECR

DONECR is an Transfer Done Clear Interrupt Clear Register Offset Address: 0x134 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:1	R	RSVD_DONECR	Reserved bit	0x0
0	RC	DONECR	Clear Transfer Done Interrupt. This register will reflect the status of the interrupt. A read from this register clears the ssi_done_intr interrupt.Writing to this register has no effect.	0x0

XIP_WRITE_INCR_INST

XIP_WRITE_INCR_INST is a XIP Write INCR transfer opcode Offset Address: 0x140 Total Reset Value:0x0

Bits	Access	Name	Description	Reset

Bits	Access	Name	Description	Reset
31:16	R	RSVD_INCR_INST_16to31	Reserved bits - Read Only	0x0
15:0	R/W	INCR_WRITE_INST	XIP Write INCR transfer opcode. When XIP_WRITE_CTRL.INST_L is not equal to 0, the controller sends instruction for all XIP write transfers, this register field stores the instruction op-code to be sent when an INCR type XIP Write transfer is requested on AHB bus. The number of bits to be send in instruction phase is determined by XIP_WRITE_CTRL.INST_L field.	0x0

XIP_WRITE_WRAP_INST

XIP_WRITE_WRAP_INST is an XIP Write WRAP transfer opcode Offset Address: 0x144 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:16	R	RSVD_WRAP_INST_16to31	Reserved bits - Read Only	0x0
15:0	R/W	WRAP_WRITE_INST	XIP Write WRAP transfer opcode. When XIP_WRITE_CTRL.INST_L is not equal to 0, the controller sends instruction for all XIP write transfers, this register field stores the instruction op-code to be sent when an WRAP type XIP Write transfer is requested on AHB bus. The number of bits to be send in instruction phase is determined by XIP_WRITE_CTRL.INST_L field.	0x0

XIP_WRITE_CTRL

XIP_WRITE_CTRL is a XIP Write Control Register Offset Address: 0x148 Total Reset Value:0x00000072

Bits	Access	Name	Description	Reset
31:21	R	RSVD_XIP_WRITECTRL_21to31	Reserved bits - Read Only	0x0
20:16	* Varies	XIPWR_WAIT_CYCLES	Wait cycles in Dual/Quad/Octal mode between control frames transmit and data reception. Specified as number of SPI clock cycles.	0x0
15:14	R	RSVD_XIP_WRITECTRL_14to15	Reserved bits - Read Only	0x0
13	* Varies	XIPWR_RXDS_SIG_EN	Enable rxds signaling during address and command phase of Hyperbus transfer. This bit enables rxds signaling by Hyperbus slave devices during Command-Address (CA) phase. If the rxds signal is set to 1 during the CA phase of transfer, the controller transmits (2*XIP_WRITE_CTRL.WAIT_CYCLES-1) wait cycles after the address phase is complete.	0x0

Bits	Access	Name	Description	Reset
12	* Varies	XIPWR_HYPERBUS_EN	SPI Hyperbus Frame format enable for XIP Write transfers. Selects if data frame format for XIP Write transfers is in Hyperbus mode. This field is effective only when CTRLRO.FRF is set to SPI frame format.	0x0
11	* Varies	WR_INST_DDR_EN	Instruction DDR Enable bit. This will enable Dual-data rate transfer for Instruction phase.	0x0
10	* Varies	WR_SPI_DDR_EN	SPI DDR Enable bit. This will enable Dual-data rate transfers in Dual/Quad/Octal frame formats of SPI.	0x0
9:8	R/W	WR_INST_L	Dual/Quad/Octal mode instruction length in bits. Values: - 0x0 (INST_L0): No Instruction - 0x1 (INST_L4): 4 bit Instruction length - 0x2 (INST_L8): 8 bit Instruction length - 0x3 (INST_L16): 16 bit Instruction length	0x0
7:4	R/W	WR_ADDR_L	This bit defines Length of Address to be transmitted. Only after this much bits are programmed in to the FIFO the transfer can begin. Values: - 0x0 (ADDR_L0): Reserved - 0x1 (ADDR_L4): 4 bit Address length - 0x2 (ADDR_L8): 8 bit Address length - 0x3 (ADDR_L12): 12 bit Address length - 0x4 (ADDR_L16): 16 bit Address length - 0x5 (ADDR_L20): 20 bit Address length - 0x6 (ADDR_L24): 24 bit Address length - 0x7 (ADDR_L28): 28 bit Address length - 0x8 (ADDR_L32): 32 bit Address length	0x7
3:2	R/W	WR_TRANS_TYPE	Address and instruction transfer format. Selects whether the controller will transmit instruction/address either in Standard SPI mode or the SPI mode selected in XIP_WRITE_CTRL.FRF field. Values: - 0x0 (TT0): Instruction and Address will be sent in Standard SPI Mode. - 0x1 (TT1): Instruction will be sent in Standard SPI Mode and Address will be sent in the mode specified by XIP_WRITE_CTRL.FRF. - 0x2 (TT2): Both Instruction and Address will be sent in the mode specified by XIP_WRITE_CTRL.FRF. - 0x3 (TT3): Reserved.	0x0

Bits	Access	Name	Description	Reset
1:0	R/W	WR_FRF	SPI Frame Format.Selects data frame format for Transmitting the data. Values: - 0x0 (RSVD): Reserved - 0x1 (SPI_DUAL): Dual SPI Format - 0x2 (SPI_QUAD): Quad SPI Format - 0x3 (SPI_OCTAL): Octal SPI Format	0x2

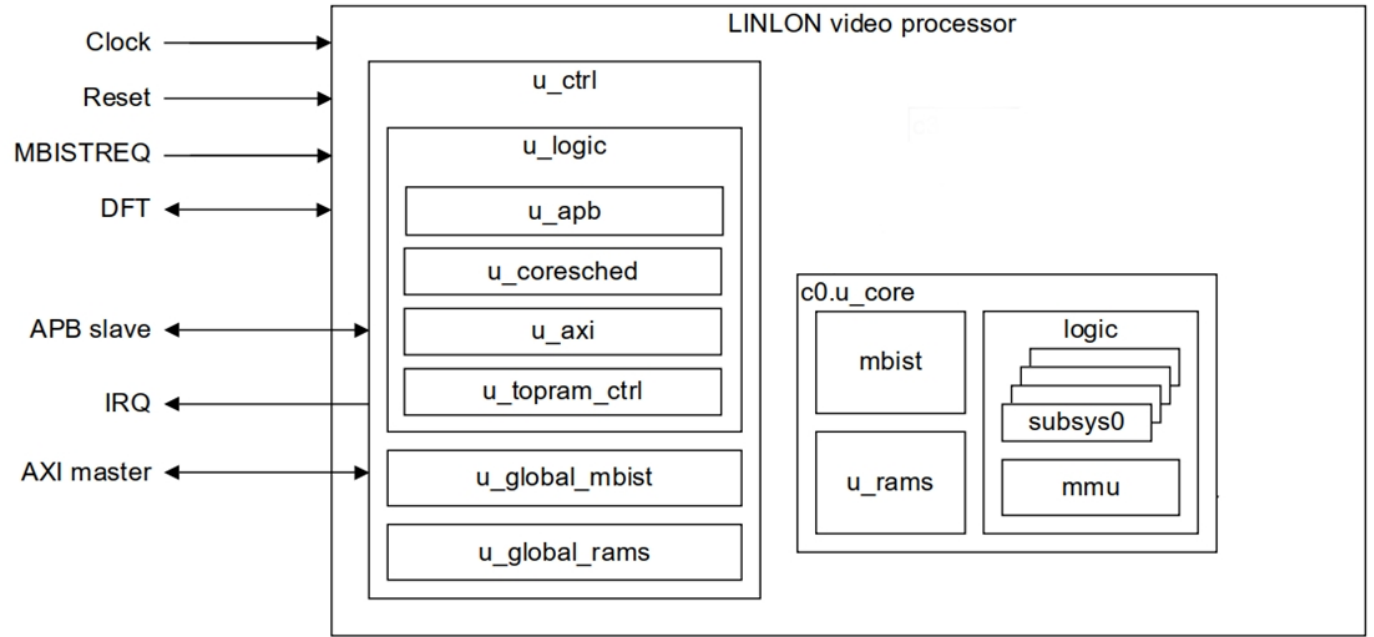
6 Video Codec

6.1 Overview

The video is codec that supports H.264/H265/JPEG/MJPEG protocol and performs encode and decode by hardware. The design is half-duplex. At a low level, encoding and decoding never run simultaneously, instead they share hardware resources. When the processor is encoding, decoding, or both encoding and decoding multiple streams concurrently, the video processor is time-multiplexed by switching sessions between frames.

6.2 Block diagram

Figure 6-2-1 video core block diagram



6.3 Function description

6.3.1 Features

Video encoding The video processor supports several video encoding formats.

General encoding features

- Stride support.
- Horizontal and vertical mirroring.

- Support for optional rotation of the source frame in 90 degree steps before encoding. Note If YUV422 is rotated by 90 degrees or 270 degrees and not converted to YUV420, the result is encoded as YUV440.

The video processor supports encoding the following source-frame input formats:

- 1-plane YUV422, scan-line format, interleaved in YUYV or UYVY order. Note YUV422 inputs can be converted to YUV420 format.
- 1-plane RGB, 8-bit, byte address order RGBA, BGRA, ARGB, or ABGR.
- 2-plane YUV420, scan-line format, chroma interleaved in UV or VU order.
- 3-plane YUV420, scan-line format.
- 3-plane YUV420 10-in-16 bit scan line format. Note Support for 3-plane formats is included to help you with testing. Do not use these for maximum performance.
- YUV420 10-bit, P010, little endian, MSB aligned.
- YUV420 10-bit, 2 × 2 tile format.

Supported codecs

The following codec standards are supported for encoding:

- HEVC (H.265) Main.
- HEVC (H.265) Main 10 Profile.
- H.264 Baseline Profile (BP).
- H.264 Main Profile (MP).
- H.264 High Profile (HP).
- H.264 High 10 Profile.
- VP8.
- VP9 Profile 0.
- VP9 Profile 2 at 10-bit depth.
- JPEG, baseline sequential, YUV420 and YUV422. Note Only the JPEG codec can encode YUV422.

HEVC (H.265) encoding features

- The encoded bitstream complies with the HEVC (H.265) Main Profile.
- Encoding speed of 1080p30 for a single core at 300MHz.
- Bitrates up to 50MBit/s for a single core at 300MHz.
- Maximum frame width of 4,096 pixels.
- Maximum frame height of 4,096 pixels.
- 8-bit or 10-bit sample depth with I, P, and Bframes.
- Progressive encoding with 64 × 64 CTU size.
- Tiled mode, up to four tiles, horizontal splits only.
- Wavefront parallel encoding.
- The Motion Estimation (ME) search window is ±128 pixels horizontally, ±64 pixels vertically. - ME search down to Quarter Picture Element (QPEL) resolution.
- 8 × 8, 16 × 16, and 32 × 32 luma intra-modes.
- 4 × 4, 8 × 8, and 16 × 16 chroma intra-modes.
- 8 × 8, 16 × 16, and 32 × 32 inter-modes.
- 8 × 8, 16 × 16, and 32 × 32 transform size for luma.
- 4 × 4, 8 × 8, and 16 × 16 transform size for chromas.
- Skipped CUs, Merge modes.
- Deblocking.
- Sample Adaptive Offset (SAO).
- Constrained intra-prediction selectable.
- Fixed Quantization Parameters (QP) operation or rate-controlled operation.

- Rate controlled based on bitrate and buffer size settings. This is also known as leaky bucket.
- Long term reference frame.
- Selectable intra-frame refresh interval.
- Slice insertion on a CTU row granularity.
- Possible to limit the search window and the split options.
- Encoders do not prevent the output from exceeding the maximum number of bits per CTU.

H.264 encoding features

- The encoded bitstream complies with the Baseline, Main, High, and High 10 Profiles.
- Encoding speed of 1080p30, single core at 300MHz.
- Bitrates up to 50MBit/s, single core at 300MHz.
- Maximum frame width of 4,096 pixels.
- Maximum frame height of 4,096 pixels.
- I, P, and B frames.
- Progressive encoding.
- Context Adaptive Binary Arithmetic Coding (CABAC) or Context Adaptive Variable Length Coding (CAVLC) entropy coding. Note B frames are not supported with CAVLC entropy coding.
- The Motion Estimation (ME) search window is ± 128 pixels horizontally, ± 64 pixels vertically.
- ME search down to Quarter Picture Element (QPEL) resolution.
- All 4×4 , 8×8 , 16×16 luma, and 8×8 chroma intra-modes evaluated.
- 8×8 , and 16×16 inter-modes.
- 4×4 and 8×8 transform.
- Skipped macroblocks.
- Deblocking.
- Constrained intra-prediction selectable.
- Fixed QP operation or rate-controlled operation.
- Rate controlled based on bitrate and buffer size settings. This is also known as leaky bucket.
- Long term reference frame.
- Selectable intra-frame refresh interval.
- Slice insertion on 32-pixel high row granularity.
- Possible to limit the search window and the macroblock split options.
- Escaping to prevent Network Abstraction Layer (NAL) unit start code emulation. This is always enabled and is independent of the NAL packet format setting. For more information, see ITU-T H.264 Annex B. Note Encoders do not prevent the output from exceeding the maximum number of bits per macroblock.

VP8 encoding features

- Encoding speed of 1080p30, single core at 400MHz.
- Bitrates up to 50MBit/s, single core at 400MHz.
- Maximum frame width of 2,048 pixels.
- Maximum frame height of 2,048 pixels.
- I and P frames.
- Progressive encoding.
- The ME search window is ± 128 pixels horizontally, ± 64 pixels vertically.
- ME search down to QPEL resolution.
- All 4×4 and 16×16 luma, and 8×8 chroma intra-modes evaluated.
- 8×8 , and 16×16 inter-modes.
- Skipped macroblocks.
- Deblocking.
- Fixed QP operation or rate-controlled operation.
- Rate controlled based on bitrate and buffer size settings. This is also known as leaky bucket.

- Selectable intra-frame refresh interval.
- Possible to limit the search window and the macroblock split options.

VP9 encoding features

- The encoded bitstream complies with the VP9 Profile 0 at 8-bit depth, or Profile 2 at 10-bit depth.
- Encoding speed of 1080p30, single core at 300MHz.
- Bitrates up to 50MBit/s, single core at 300MHz.
- Maximum frame width of 4,096 pixels.
- Maximum frame height of 4,096 pixels.
- 8-bit or 10-bit sample depth. I and P frames.
- Progressive encoding.
- Tiled rows and columns.
- The Motion Estimation (ME) search window is ± 128 pixels horizontally and ± 64 pixels vertically.
- ME search down to Quarter Picture Element (QPel) resolution.
- 8×8 , 16×16 , and 32×32 luma intra-modes.
- 4×4 , 8×8 , and 16×16 chroma intra-modes.
- 8×8 , 16×16 , and 32×32 inter-modes.
- 8×8 , 16×16 , and 32×32 transform size for luma.
- 4×4 , 8×8 , and 16×16 transform size for chroma.
- Skipped superblocks.
- Deblocking.
- Fixed QP operation or rate-controlled operation.
- Rate controlled based on bitrate and buffer size settings, this is also known as leaky bucket.
- Selectable intra-frame refresh interval.
- Implicit or explicit probability update, using delayed contexts.
- Encoding speed of 1080p40 for 4:2:2, single core at 300MHz.
- Bitrates up to 90MBit/s for 4:2:2, single core at 300MHz.
- 4:2:0 and 4:2:2 baseline sequential.
- Programmable full set of Huffman symbols, separate for luma and chroma.
- Programmable full set of quantization matrices, separate for all color components.
- Programmable DC restart interval.
- Maximum frame width of 8,192 pixels.
- Maximum frame height of 8,192 pixels.

Video decoding

The video processor supports several video codecs for decoding.

General decoding features

- Supported source frame output formats are:
 - 2 plane YUV420 scan line format, chroma interleaved in UV or VU order.
 - 3 plane YUV420 scan line format. Note Support for 3-plane formats is included to help you with testing. Do not use these for maximum performance.
 - YUV420 10 bit, P010, little endian, MSB aligned.
 - YUV420 10 bit, 2×2 tile format. Note The YUV buffer and stride must have correct alignment for maximum performance.
- Stride support for scan-line formats only.
- Support for rotation of the decoded frame in 90 degree steps before output.
- Support for downscaling the decoded frame horizontally and vertically by a factor of 2 or 4.
- Support for downscaling the decoded frame horizontally and vertically by any factor individually. The smallest resolution can be 2×2 pixels.

- Support for output of average frame luminance and chrominance for each 32 x 32 pixel block, for every output display frame.

Supported codecs

The following codec standards are supported for decoding:

- HEVC (H.265) Main/Main10.
- H.264 Baseline, Main, High, and High10 Progressive Profile.
- AVS Part 2 (Jizhun).
- AVS Part 16 (AVS+, Guangdong).
- AVS2 Part 2 (GY/T 299.1) Main Picture/Main/Main10 bit profile.
- VP8.
- VP9 Profile 0 and VP9 Profile 2 at 10-bit depth, not 12-bit.
- JPEG, baseline sequential, YUV420 and YUV422. Note Only the JPEG codec can decode YUV422.
- VC 1 SP/MP/AP.
- MPEG4 SP/ASP.
- MPEG2 MP.
- H.263 Profile 0.
- RealVideo 8, 9, and 10.

HEVC (H.265) decoding features

- The decoder is fully compliant to the Main and Main10 Profiles.
- Decoding speed equals 2160p30, single core at 600MHz.
- Average bitrates up to 100MBit/s, single core at 600MHz.
- Maximum frame width is 4,096 pixels.
- Maximum frame height is 4,096 pixels.
- Error concealment is performed in case of bit errors.
- Stream parameter information is output. H.264 decoding features
- Baseline, Main, High, and High 10 progressive Profiles.
- If Flexible Macroblock Ordering (FMO) or Arbitrary Slice Ordering (ASO) are used for baseline streams, the decoding speed is WVGA at 30fps for a single core at 400MHz.
- If FMA and ASO are not used, the decoding speeds are:
 - 2160p30 for single core at 600MHz.
 - 1080i60 for single core at 400MHz.
- For progressive streams
 - Average bitrates up to 100MBit/s for a single core at 600MHz.
 - Maximum frame width is 4,096 pixels.
 - Maximum frame height is 4,096 pixels.
- For interlaced streams
 - Average bitrates up to 50MBit/s for a single core at 400MHz.
 - A maximum frame width of 2,048 pixels.
 - A maximum frame height of 4,096 pixels.
- Error concealment is performed in case of bit errors.
- Stream parameter information is output. Escaping to prevent NAL unit start code emulation is always expected. This is independent of the NAL packet format setting. For more information, see ITU T H.264 Annex B. AVS decoding features
- The decoder is fully compliant to AVS Part 2 (Jizhun) and Part 15 (AVS+, Guangdong) for YUV420 only.
- Decoding speed equals 1080p30, single core at 400MHz.
- Average bitrates up to 20MBit/s, single core at 400MHz.
- Maximum frame width is 1,920 pixels.
- Maximum frame height is 1,080 pixels.

- Error concealment is performed in case of bit errors. AVS2 decoding features
- The decoder is fully compliant to AVS2 Part 2 (GY/T 299.1) for YUV420 only.
- Decoding speed equals 2160p30, single core at 600MHz.
- Average bitrates up to 100MBit/s, single core at 600MHz.
- Maximum frame width is 4,096 pixels.
- Maximum frame height is 4,096 pixels.
- Error concealment is performed in case of bit errors.

VP8 decoding features

- The decoder is fully compliant to the VP8 Specification.
- Decoding speed equals 1080p30, single core at 400MHz.
- Average bitrates up to 50MBit/s, single core at 400MHz.
- Maximum frame width is 2,048 pixels.
- Maximum frame height is 2,048 pixels.
- Error concealment is performed in case of bit errors.

VP9 decoding features

- The decoder is fully compliant to Profile 0 and Profile 2, 10-bit. Profile 2 at 12-bit is not supported.
- Decoding speed equals 2160p30, single core at 600MHz assuming no non-visible, no AltReframes.
- Decoding speed equals 2160p30, single core at 800MHz assuming an AltRef distance of 4.
- Average bitrates up to 60MBit/s, single core at 600MHz.
- Maximum frame width is 4,096 pixels.
- Maximum frame height is 4,096 pixels.
- Error concealment is performed in case of bit errors.
- Stream parameter information is output. VC-1 decoding features
- The decoder is fully compliant to VC-1 Simple, Main, and Advanced Profiles.
- Decoding speed equals 1080p30, or 1080i60, single core at 400MHz.
- Average bitrates up to 40MBit/s, single core at 400MHz.
- Maximum frame width is 2,048 pixels.
- Maximum frame height is 4,096 pixels.
- Error concealment is performed in case of bit errors. Advanced Profile bitstream data is always expected to include the Encapsulation Mechanism. This is independent of the NAL packet format setting. For more information see SMPTE-421M-2006 Annex E. Note The range mapping feature of the VC-1 Advanced Profile does not apply to AFBC output.

JPEG decoding features

- Decoding speed equals 1080p30, single core at 300MHz.
- Average bitrates up to 20MBit/s, single core at 300MHz.
- Support for 4:2:0 output by the video encoder, baseline sequential YYYYCbCr interleaved MCU format only.
- Support for 4:2:2 output by the video encoder, baseline sequential YYCbCr interleaved MCU format only.
- Maximum frame width is 8,192 pixels.
- Maximum frame height is 8,192 pixels.
- Error concealment is performed in case of bit errors.
-

MPEG4 decoding features

- The decoder is compliant to MPEG4 Simple Profile and Advanced Simple Profile.
- Global Motion Compensation (GMC) is limited to a single warp point.
- Decoding speed equals 1080p30, or 1080i60, single core at 400MHz.

- Average bitrates up to 20MBit/s, single core at 400MHz.
- Maximum frame width is 2,048 pixels.
- Maximum frame height is 2,048 pixels.
- Error concealment is performed in case of bit errors.

MPEG2 decoding features

- The decoder is compliant to MPEG2 Main Profile.
- Decoding speed equals 1080p30, or 1080i60, single core at 400MHz.
- Average bitrates up to 20MBit/s, single core at 400MHz.
- Maximum frame width is 4,096 pixels. A maximum frame width of 2048 pixels for interlaced streams.
- Maximum frame height is 4,096 pixels.
- Error concealment is performed in case of bit errors.

H.263 decoding features

- The decoder is compliant to H.263 Profile 0.
- Decoding speed equals 1080p30, single core at 400MHz.
- Average bitrates up to 20MBit/s, single core at 400MHz.
- Maximum frame width is 2,048 pixels.
- Maximum frame height is 2,048 pixels.
- Error concealment is performed in case of bit errors.

RealVideo decoding features

- The decoder is compliant to RealVideo 8, 9, and 10 streams.
- Decoding speed equals 1080p30, single core at 400MHz.
- Average bitrates up to 20MBit/s, single core at 400MHz.
- Maximum frame width is 2,048 pixels.
- Maximum frame height is 2,048 pixels.
- Error concealment is performed in case of bit errors.
- Video in the RealMedia format can be decoded on both single core and multicore configurations, but video in the start code format can only be decoded on a single core configuration.

7 Video and Graphics Processg

7.1 2.5D GPU

7.1.1 Overview

2.5D GPU defines a next generation high-performance/area UI graphics core designed for hardware acceleration of Graphics API for vector graphics display on a variety of consumer devices.

7.1.2 Function description

Feature	Description
Area	Feature set optimized for small footprint
Blending	8 Porter Duff blending modes
Rotation	To any angle
Fill rules	Odd/even and non-zero

Feature	Description
Paint paths	Linear and curved paint paths
Compression Options	ETC2/DEC600Nano
Coordinate systems and transformations	Tessellation uses 3x3 affine transformation. Image drawing uses a 3x3 perspective transformation matrix
Paints	Solid and linear paint
Pixel formats for image interpolation	14 pixel formats for image interpolation, including A4/A8
Maximum performance	Maximum support 640*480@30fps

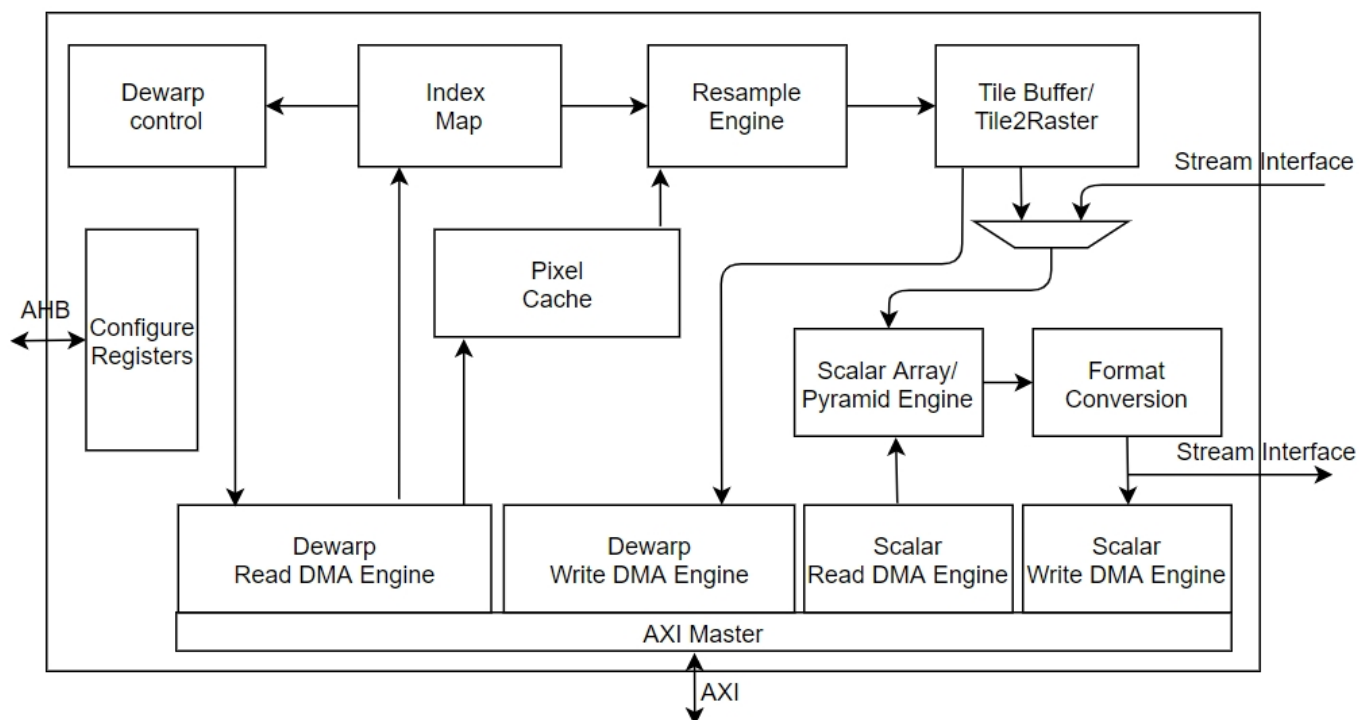
7.2 Dewarp

7.2.1 Overview

Dewarp provides both dewarp processing and scalar functionality in a single module. The high performance dewarp processing module allows for the correction of the distortion that is introduced in images produced by fisheye and wide angle lenses. It is implemented with a line/tile-cache based architecture. With configurable address mapping look up tables and per tile processing, it successfully generates a corrected output image. The scalar processing module provides high-performance scalar processing for YUV images. Three scalar engines and dual AXI master write interfaces provide efficient throughput with a maximum output size of 4096x2160.

7.2.2 Block diagram

Figure 7-2-1 dewarp block diagram



7.2.3 Function description

7.2.3.1 dewarp

- Line based architecture and configurable map to support various lens distortions

- YUV422 and YUV420 inputs and outputs
- Configurable bilinear interpolation engine
- Grid map with 16x16 pixel macroblocks indexed using X, Y vertex coordinates
- Fisheye correction
- Wide Field of View (FOV) correction
- Keystone correction
- Scaling and 4PTZ view

7.2.3.2 Scaler

- support three scaler engine, with maximum resolutions of: 4K
- Supports YCbCr 4:2:2 and YCbCr 4:2:0 semi-planar inputs
 - For data from DDR: supports YUV422sp, YUV422 interleave and YUV420sp
 - For data from the ISP stream interface: supports YUV420sp and YUV422 sp
 - For data from Dewarp: supports YUV422sp
- Supports YUV and RGB888 output
- Supportw different output formats by choosing different scaling factors for luminance and chrominance components: YUV444, YUV422, YUV420, RGB888
- Support for co-sited and non-co-sited output formats by programmable phase offsets.
- Supports 8 bit or 10 bit component data
- Bypass support
- Separate scaling in horizontal and vertical directions, and for chrominance and luminance components
- Lookup table for programmable scaling characteristics
- Shadow registers for on-the-fly reprogramming of control registers
- Low power design techniques

7.3 AI2D Accelerator

7.3.1 overview

The AI2D accelerator is mainly used to realize the calculation of image resize/affine, crop, and pad by hardware, and is used to reduce the workload of the CPU. The CPU or GNNE sends a command to the AI2D accelerator and starts the calculation. After the calculation is completed, the AI2D accelerator will send an interrupt to the CPU or send a done signal to the GNNE.

7.3.2 Functional description

- resize
- affine
- The calculation of resize and affine coordinates supports opencv's round, as well as ceil and floor
- pad Support four edge parts to complement constant pads, the pad value can be specified, and the maximum number of pads on a single side is 1023

- csc

Both input and output images support the following formats YUV420_NV12

YUV420_NV12

YUV420_I420

YUV400

RGB_packed

RGB_planar

NCHW

- shift
RAW16 input -> shift to RAW8 output
- Support CPU to read internal registers
- The accelerator can access the system SRAM(2MB).
- Support software reset through configuration register
- Support clk gating

NOTE

- Note1: The highest source image resolution supports 4K
- Note2: All configured ddr addresses follow 16B align, which is friendly to performance
- Note3: A configuration register cannot exceed 1KB
- Note4: When configuring cpu, configure 32bit each time, and configure 4 times for each 128bit, and must configure 4 times, the order is from low to high; when configuring GNNE, configure 128bit at a time

7.3.3 Register description

Table 7-1 AI2D register list

Offset Address: 0x0 Total Reset Value: 0x0

Bits	Name	Mode	Description
[127:96]	2d1_src_ch3_addr	RW	Start address of source data channel 3 (larger image)
[95:64]	2d1_src_ch2_addr	RW	Start address of source data channel 2 (larger image)
[63:32]	2d1_src_ch1_addr	RW	Start address of source data channel 1 (larger image)
[31:0]	2d1_src_ch0_addr	RW	Start address of source data channel 0 (larger image)

Offset Address: 0x10 Total Reset Value: 0x0

Bits	Name	Mode	Description
[127:96]	2d1_dst_ch3_addr	RW	Start address of destination data channel 3 (larger image)
[95:64]	2d1_dst_ch2_addr	RW	Start address of destination data channel 2 (larger image)
[63:32]	2d1_dst_ch1_addr	RW	Start address of destination data channel 1 (larger image)
[31:0]	2d1_dst_ch0_addr	RW	The start address of destination data channel 0 (larger picture)

Offset Address: 0x20 Total Reset Value: 0x0

Bits	Name	Mode	Description
[127:112]	2d1_dst_ch3_line_stride	RW	line byte stride of destination data channel 3
[111:96]	2d1_dst_ch2_line_stride	RW	line byte stride of destination data channel 2
[95:80]	2d1_dst_ch1_line_stride	RW	line byte stride of destination data channel 1
[79:64]	2d1_dst_ch0_line_stride	RW	Line byte stride of destination data channel 0
[63:48]	2d1_src_ch3_line_stride	RW	Line byte stride of source data channel 3
[47:32]	2d1_src_ch2_line_stride	RW	Line byte stride of source data channel 2

Bits	Name	Mode	Description
[31:16]	2d1_src_ch1_line_stride	RW	Line byte stride of source data channel 1
[15:0]	2d1_src_ch0_line_stride	RW	Line byte stride of source data channel 0

Offset Address: 0x30 Total Reset Value: 0x0

Bits	Name	Mode	Description
[127:96]	2d1_coef_m4	RW	Affine Variation Matrix Coefficient M4
[95:64]	2d1_coef_m3	RW	Affine Variation Matrix Coefficient M3
[63:32]	2d1_coef_m1	RW	Affine Variation Matrix Coefficient M1
[31:0]	2d1_coef_m0	RW	Affine variation matrix coefficient M0

Offset Address: 0x40 Total Reset Value: 0x0

Bits	Name	Mode	Description
[127:113]	reserved		
[112]	2d1_bound_smooth	RW	0: no interpolation smoothing between boundary points and bound, 1: interpolation smoothing between boundary points and bound
[111:96]	2d1_bound_val	RW	When bound_ind==0, the value of the boundary padding
[95:92]	2d1_dst_format	RW	0-yuv_420_nv12 1-yuv_420_nv21 2-yuv420_i420 3-nchw 4-rgb888 5-raw16
[91:88]	2d1_src_format	RW	0-yuv_420_nv12 1-yuv_420_nv21 2-yuv420_i420 3-nchw 4-rgb888 5-raw16
[87:84]	2d1_bound_ind	RW	When bilinear interpolation. 0: Complement constant beyond the boundary, 1: Copy boundary beyond the boundary
[80:76]	2d1_shift	RW	Negative 8 to 8. If shift is less than 0, shift left; if shift is greater than 0, shift right with round, for raw16 to raw8
[75:74]	2d1_pad_mode	RW	0: const mode, 1: copy mode, 2: mirror mode
[73:72]	2d1_inter_mode	RW	0: nearest neighbor; 1: bilinear
[71:70]	2d1_cord_round	RW	Rounding method for coordinate calculations. 0: round 1: ceil, 2: floor. Configure according to tf/opencv
[69:67]	2d1_dst_channel_num	RW	Number of channels, used in nchw and raw16 formats
[66:64]	2d1_src_channel_num	RW	Channel number, used in nchw and raw16 formats
[63:32]	reserved		
[31:0]	reserved		

Offset Address: 0x50 Total Reset Value: 0x0

Bits	Name	Mode	Description

Bits	Name	Mode	Description
[127:120]	reserved		
[119:108]	2d1_csc_coef7	RW	Color gamut conversion coefficient
[107:96]	2d1_csc_coef6	RW	Color gamut conversion coefficient
[87:76]	2d1_csc_coef5	RW	Color gamut conversion coefficient
[75:64]	2d1_csc_coef4	RW	Color gamut conversion coefficient
[55:44]	2d1_csc_coef3	RW	Color gamut conversion coefficient
[43:32]	2d1_csc_coef2	RW	Color gamut conversion coefficient
[23:12]	2d1_csc_coef1	RW	Color gamut conversion coefficient
[11:0]	2d1_csc_coef0	RW	Color gamut conversion coefficient

Offset Address: 0x60 Total Reset Value: 0x0

Bits	Name	Mode	Description
[127:88]	reserved		
[87:76]	2d1_csc_coef9	RW	Color gamut conversion coefficient
[75:64]	2d1_csc_coef8	RW	Color gamut conversion coefficient
[59]	input_data_signed	RW	input data sign, 0: unsigned input 1: signed input
[58]	2d1_cmd_id	RW	0: cpu ,1: kpu
[57]	2d1_dst_id	RW	0: GLB, 1: DDR
[56]	2d1_src_id	RW	0: GLB, 1: DDR
[55:48]	2d1_pad_value_ch3	RW	ch3 constant pad value when pad_mod=0
[47:40]	2d1_pad_value_ch2	RW	ch2 constant pad value when pad_mod=0
[39:32]	2d1_pad_value_ch1	RW	ch1 constant pad value when pad_mod=0
[31:24]	2d1_pad_value_ch0	RW	ch0 constant pad value when pad_mod=0
[23:12]	2d1_csc_coef11	RW	Color gamut conversion coefficient
[11:0]	2d1_csc_coef10	RW	Color gamut conversion coefficient

Offset Address: 0x70 Total Reset Value: 0x0

Bits	Name	Mode	Description
[127]	2d1_csc_en	RW	Whether to enable the conversion from yuv to rgb domain
[126]	2d1_intr_mask	RW	Whether the mask is interrupted or the done signal returned to the kpu, with 1 means mask off interrupt or done signal, with 0 means no mask
[124:112]	2d1_dst_slice_height	RW	The width of the dst slice
[108:96]	2d1_dst_slice_width	RW	The width of the dst slice

Bits	Name	Mode	Description
[92:80]	2d1_src_slice_height	RW	height of src slice
[76:64]	2d1_src_slice_width	RW	Width of the src slice
[57:48]	2d1_pad_right	RW	pad right points
[41:32]	2d1_pad_left	RW	pad left points
[25:16]	2d1_pad_down	RW	pad down points
[9:0]	2d1_pad_top	RW	pad top points

Offset Address: 0x80 Total Reset Value: 0x0

Bits	Name	Mode	Description
[127]	2d1_calc_en WO 2d1 calculation enable		
[126:125]	reserved		
[124:112]	2d1_dst_slice_offset_y	RW	The starting offset of the current slice in the y direction of the target image (pixel offset)
[108:96]	2d1_dst_slice_offset_x	RW	The starting offset of the current slice in the target x direction (pixel offset)
[92:80]	2d1_src_slice_offset_y	RW	The starting offset of the current slice in the y direction of the original image (pixel offset)
[76:64]	2d1_src_slice_offset_x	RW	The starting offset of the current slice in the x direction of the original image (pixel offset)
[63:32]	2d1_coef_m5	RW	Affine Variation Matrix Coefficient M5
[31:0]	2d1_coef_m2	RW	Affine variation matrix coefficient M2

Offset Address: 0x90 Total Reset Value: 0x0

Bits	Name	Mode	Description
[127:32]	reserved		
[31:16]	wt_os_cnt	RO	Write bus outstanding counter value
[15:0]	2d1_intr_num	RO	Interrupt number register, 0 means normal interrupt, 1 means timeout interrupt, 2 means write exception interrupt

Offset Address: 0xa0 Total Reset Value: 0x0

Bits	Name	Mode	Description
[127:1]	reserved		
[0]	2d1_intr_clear	WO	Write 1 to clear interrupt

Offset Address: 0xb0 Total Reset Value: 0x0

Bits	Name	Mode	Description
[127:1]	reserved		
[0]	stop_busy	RO	Hardware internal self-clearing

Offset Address: 0xc0 Total Reset Value: 0x0

Bits	Name	Mode	Description
[127:33]	reserved		
[32]	disable_auto_cg	RW	Not configured by default, with 1 means disable clock gating inside the hardware
[31:0]	wcd_num	RW	Hardware watchdog timeout upper limit

Offset Address: 0xd0 Total Reset Value: 0x0

Bits	Name	Mode	Description
[127:0]	debug_reg	RO	debug register

7.4 NONAI-2D

7.4.1 Overview

The 2D accelerator is mainly used for hardware realization of image OSD, CSC calculation and draw border,the external interface is the standard axi4 master interface (data interface), apb3 interface (configuration path) and interrupt output.

7.4.2 Features

- Support OSD mode
- Support CSC mode
- Support draw border mode
- Support corp operation

7.4.3 Block Diagram

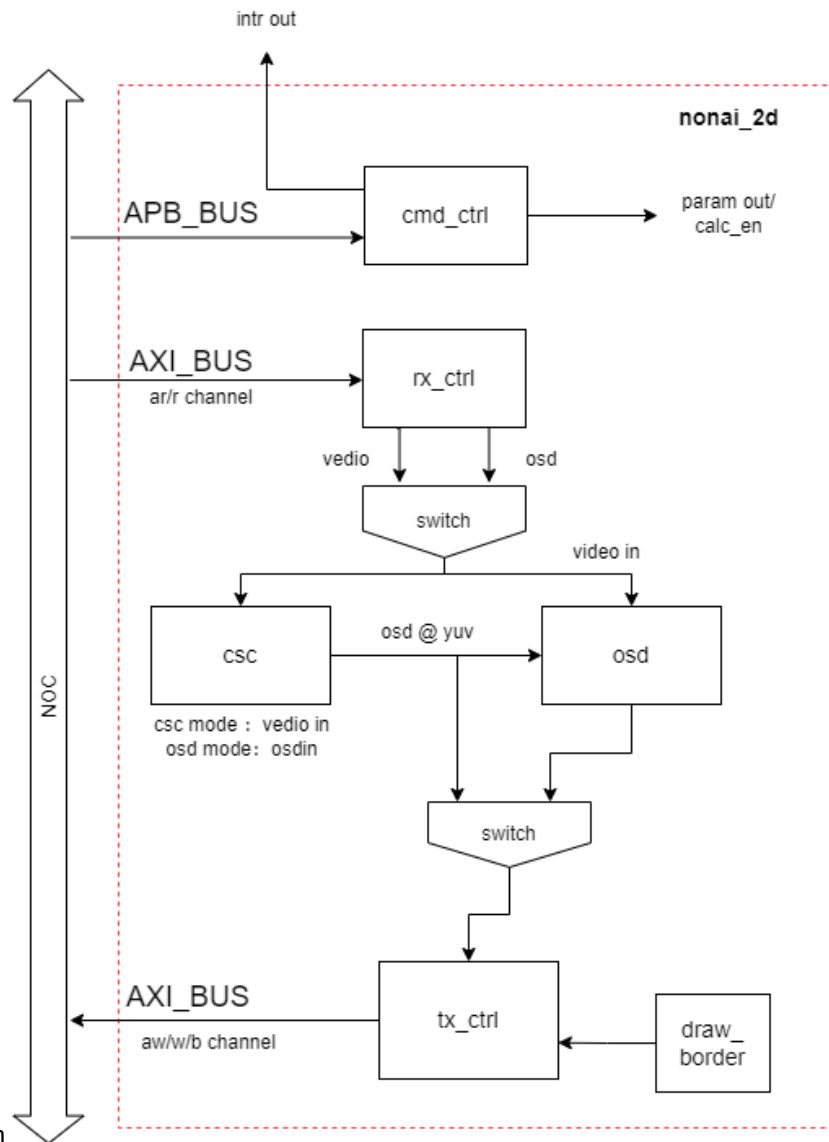


Figure 7-4-1 noai-2d block diagram

Submodule description

1.cmd_ctrl module, the instruction analysis module, receives the configuration information from the bus, analyzes the relevant parameters, and completes the interrupt interaction.

2.rx_ctrl module, the main job of this module is to generate addresses according to the input image and various parameters of osd; and cache the loaded data and output it to the computing module.

3.CSC module, the main function of the csc module is to convert the input data into a variety of different color space formats.

4.OSD module, weighted and superimposed computing unit, bypass in non-osd computing mode.

5.tx_ctrl module, the output result buffer unit; at the same time, the calculation result is written back to the AXI bus.

6.draw_border module, the video format currently only supports NV12, NV21, and I420. It does not need to load data through the bus, and directly writes the pixel value of the border to the corresponding position of the image through the axi write port.

7.4.4 Function Description

- The osd module supports up to three channels of data overlay, and the overlay order is configurable, and each map supports up to 8 areas.

- CSC conversion coefficient can be configured.
- Supports up to 32 borders, and borders can overlap.
- Internally integrates two-dimensional TX DMA and RX DMA, supports crop operation, supports standard axi-4 protocol, data bus bit width 64bit, address bus bit width 32bit.
- The data bit width of the axi bus is 64 bits, the maximum outstanding is 8, and the maximum burst len is limited to 16.
- The configuration interface supports the apb3.0 protocol, with a data bit width of 32 bits and an address bit width of 32 bits.
- Support nonai2d internal reset.
- Support dynamic CLK Gating.

7.4.5 Register Summary

The following table is the list of nonai2d registers, where 0x3a0-0x3c0 are public registers, 0x000-0x380 are stream registers, this design supports 3 streams, and the base addresses of the three streams are 0x0000_0000, 0x0000_0400, 0x0000_0800 respectively. For the convenience of expression, the register and offset address of stream0 are listed, and the base address can be added to the offset addresses of the other two streams.

Offset	Name	Description
0x000	STREAM0_MAIN_CTRL_CFG0	
0x004	STREAM0_MAIN_CTRL_CFG1	
0x008	STREAM0_MAIN_CTRL_CFG2	
0x00c	STREAM0_MAIN_CTRL_CFG3	
0x010	STREAM0_MAIN_CTRL_CFG4	
0x014	STREAM0_MAIN_CTRL_CFG5	
0x018	STREAM0_MAIN_CTRL_CFG6	
0x01c	STREAM0_MAIN_CTRL_CFG7	
0x03c	STREAM0_MAIN_CTRL_CFG8	
0x05c	STREAM0_MAIN_CTRL_CFG9	
0x07c	STREAM0_MAIN_CTRL_CFG10	
0x09c	STREAM0_MAIN_CTRL_CFG11	
0x0bc	STREAM0_MAIN_CTRL_CFG12	
0x0dc	STREAM0_MAIN_CTRL_CFG13	
0x0fc	STREAM0_MAIN_CTRL_CFG14	
0x11c	STREAM0_MAIN_CTRL_CFG15	
0x13c	STREAM0_MAIN_CTRL_CFG16	
0x140	STREAM0_MAIN_CTRL_CFG17	
0x144	STREAM0_MAIN_CTRL_CFG18	
0x148	STREAM0_MAIN_CTRL_CFG19	
0x14c	STREAM0_MAIN_CTRL_CFG20	
0x150	STREAM0_MAIN_CTRL_CFG21	

Offset	Name	Description
0x1d0	STREAM0_MAIN_CTRL_CFG22	
0x250	STREAM0_MAIN_CTRL_CFG23	
0x2d0	STREAM0_MAIN_CTRL_CFG24	
0x350	STREAM0_MAIN_CTRL_CFG25	
0x354	STREAM0_MAIN_CTRL_CFG26	
0x358	STREAM0_MAIN_CTRL_CFG27	
0x35c	STREAM0_MAIN_CTRL_CFG28	
0x360	STREAM0_MAIN_CTRL_CFG29	
0x364	STREAM0_MAIN_CTRL_CFG30	
0x368	STREAM0_MAIN_CTRL_CFG31	
0x36c	STREAM0_MAIN_CTRL_CFG32	
0x370	STREAM0_MAIN_CTRL_CFG33	
0x374	STREAM0_MAIN_CTRL_CFG34	
0x378	STREAM0_MAIN_CTRL_CFG35	
0x37c	STREAM0_MAIN_CTRL_CFG36	
0x380	STREAM0_MAIN_CTRL_CFG37	
0x3a0	MAIN_CTRL_CFG31	
0x3a4	MAIN_CTRL_CFG32	
0x3a8	MAIN_CTRL_CFG33	
0x3ac	MAIN_CTRL_CFG34	
0x3b0	MAIN_CTRL_CFG35	
0x3b4	MAIN_CTRL_CFG36	
0x3b8	MAIN_CTRL_CFG37	
0x3bc	MAIN_CTRL_CFG38	
0x3c0	MAIN_CTRL_CFG39	

7.4.6 Register Description

STREAM0_MAIN_CTRL_CFG0 (description) STREAM0_MAIN_CTRL_CFG0 is a register that configures input image pixels. Offset Address: 0x000 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[31:16]	RW	stream0_gb_src_height	input image height	0x0
[15:0]	RW	stream0_gb_src_width	input image width	0x0

STREAM0_MAIN_CTRL_CFG1 (description) STREAM0_MAIN_CTRL_CFG1 the register to configure the start address of the input image. Offset Address: 0x004 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[31:0]	RW	stream0_gb_src_ch0_addr	the start address of image channel 0	0x0

STREAM0_MAIN_CTRL_CFG2 (description) STREAM0_MAIN_CTRL_CFG2 the register to configure the start address of the input image. Offset Address: 0x008 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[31:0]	RW	stream0_gb_src_ch1_addr	the start address of image channel 1	0x0

STREAM0_MAIN_CTRL_CFG3 (description) STREAM0_MAIN_CTRL_CFG3 the register to configure the start address of the input image. Offset Address: 0x00c Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[31:0]	RW	stream0_gb_src_ch2_addr	the start address of image channel 2	0x0

STREAM0_MAIN_CTRL_CFG4 (description) STREAM0_MAIN_CTRL_CFG4 line byte stride register. Offset Address: 0x010 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[31:16]	RW	stream0_gb_src_ch1_line_byte_stride	Line byte stride of input image channel 1(The size of each row in memory)	0x0
[15:0]	RW	stream0_gb_src_ch0_line_byte_stride	Line byte stride of input image channel 0	0x0

STREAM0_MAIN_CTRL_CFG5 (description) STREAM0_MAIN_CTRL_CFG5 line byte stride register. Offset Address: 0x014 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	Reserved	0x0
[15:0]	RW	stream0_gb_src_ch2_line_byte_stride	Line byte stride of input image channel 2	0x0

STREAM0_MAIN_CTRL_CFG6 (description) STREAM0_MAIN_CTRL_CFG6 image format register. Offset Address: 0x018 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	Reserved	0x0

Bits	Access	Name	Description	Reset
[23:16]	RW	stream0_gb_dst_format	the format of the output image,0-yuv420 nv12; 1-yuv420 nv21; 2-yuv420 i420; 3-yuv422p; 4-argb8888; 5-argb4444; 6-argb1555; 7-xrgb8888; 8-xrgb4444; 9-xrgb1555;10-bgra8888; 11-bgra4444; 12-bgra5551; 13-bgrx8888; 14-bgrx4444; 15-bgrx5551 16-rgb888; 17-bgr888 ;18-rgb565; 19-bgr565; 20-separate rgb	0x0
[15:8]	-	reserved	Reserved	0x0
[7:0]	RW	stream0_gb_src_format	the format of the input image,0-yuv420 nv12; 1-yuv420 nv21; 2-yuv420 i420; 3-yuv422p; 4-argb8888; 5-argb4444; 6-argb1555; 7-xrgb8888; 8-xrgb4444; 9-xrgb1555;10-bgra8888; 11-bgra4444; 12-bgra5551; 13-bgrx8888; 14-bgrx4444; 15-bgrx5551 16-rgb888; 17-bgr888 ;18-rgb565; 19-bgr565; 20-separate rgb	0x0

STREAM0_MAIN_CTRL_CFG7 (description) STREAM0_MAIN_CTRL_CFG7 stream0_region0~stream0_region7
osd overlay area size register. Offset Address: 0x01c~0x038 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[31:16]	RW	gb_height	The height of the osd overlay area	0x0
[15:0]	RW	gb_width	The width of the osd overlay	0x0

STREAM0_MAIN_CTRL_CFG8 (description) STREAM0_MAIN_CTRL_CFG8 stream0_region0~stream0_region7
The location of the osd overlay area. Offset Address: 0x03c~0x058 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[31:16]	RW	gb_start_addr	The pixel offset of the upper left point of the osd overlay area in the y direction of the original image	0x0
[15:0]	RW	gb_start_addrx	The pixel offset of the upper left point of the osd overlay area in the x direction of the original image	0x0

STREAM0_MAIN_CTRL_CFG9 (description) STREAM0_MAIN_CTRL_CFG9 stream0_region0~stream0_region7 the
starting address of the osd overlay area. Offset Address: 0x05c~0x078 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[31:0]	RW	gb_osd_ch0_addr	The starting address of osd channel 0	0x0

STREAM0_MAIN_CTRL_CFG10 (description) STREAM0_MAIN_CTRL_CFG10 stream0_region0~stream0_region7
the starting address of the osd overlay area. Offset Address: 0x07c~0x098 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[31:0]	RW	gb_osd_ch1_addr	The starting address of osd channel 1	0x0

STREAM0_MAIN_CTRL_CFG11 (description) STREAM0_MAIN_CTRL_CFG11 stream0_region0~stream0_region7 the starting address of the osd overlay area. Offset Address: 0x09c~0x0b8 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[31:0]	RW	gb_osd_ch2_addr	The starting address of osd channel 2	0x0

STREAM0_MAIN_CTRL_CFG12 (description) STREAM0_MAIN_CTRL_CFG12 stream0_region0~stream0_region7 osd line byte stride register. Offset Address: 0x0bc~0x0d8 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[31:16]	RW	gb_osd_ch1_line_byte_stride	the line byte stride of osd channel 1	0x0
[15:0]	RW	gb_osd_ch0_line_byte_stride	the line byte stride of osd channel 0	0x0

STREAM0_MAIN_CTRL_CFG13 (description) STREAM0_MAIN_CTRL_CFG13 stream0_region0~stream0_region7 osd line byte stride register. Offset Address: 0x0dc~0x0f8 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	Reserved	0x0
[15:0]	RW	gb_osd_ch2_line_byte_stride	the line byte stride of osd channel 2	0x0

STREAM0_MAIN_CTRL_CFG14 (description) STREAM0_MAIN_CTRL_CFG14 stream0_region0~stream0_region7 osd configuration register. Offset Address: 0x0fc~0x118 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[31:24]	RW	gb_bg_global_alpha	The global alpha of the background layer	0x0
[23:16]	RW	gb_osd_global_alpha	The global alpha of the osd layer	0x0
[15:8]	RW	gb_video_global_alpha	The global alpha of the video layer	0x0
[3:0]	RW	gb_add_order	osd add order: video at the bottom->OSD 0000 OSD at the bottom->video 0001 video at the bottom->background 0010 background at the bottom->video 0011 video at the bottom->background->OSD 0100 video at the bottom->OSD->background 0101 background at the bottom->video->OSD 0110 background at the bottom->OSD->video 0111 OSD at the bottom->video->background 1000 OSD at the bottom->background->video 1001	0x0

STREAM0_MAIN_CTRL_CFG15 (description) STREAM0_MAIN_CTRL_CFG15 stream0_region0~stream0_region7 background color of osd. Offset Address: 0x11c~0x138 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[31:0]	RW	gb_bg_color	background color of osd	0x0

STREAM0_MAIN_CTRL_CFG16 (description) STREAM0_MAIN_CTRL_CFG16 the starting address of the output image. Offset Address: 0x13c Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[31:0]	RW	stream0_gb_dst_ch0_addr	The starting address of channel 0 of the output image	0x0

STREAM0_MAIN_CTRL_CFG17 (description) STREAM0_MAIN_CTRL_CFG17 the starting address of the output image. Offset Address: 0x140 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[31:0]	RW	stream0_gb_dst_ch1_addr	The starting address of channel 1 of the output image	0x0

STREAM0_MAIN_CTRL_CFG18 (description) STREAM0_MAIN_CTRL_CFG18 the starting address of the output image. Offset Address: 0x144 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[31:0]	RW	stream0_gb_dst_ch2_addr	The starting address of channel 2 of the output image	0x0

STREAM0_MAIN_CTRL_CFG19 (description) STREAM0_MAIN_CTRL_CFG19 line byte stride register. Offset Address: 0x148 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[31:16]	RW	stream0_gb_dst_ch1_line_byte_stride	the line byte stride of output image channel 1	0x0
[15:0]	RW	stream0_gb_dst_ch0_line_byte_stride	the line byte stride of output image channel 0	0x0

STREAM0_MAIN_CTRL_CFG20 (description) STREAM0_MAIN_CTRL_CFG20 line byte stride register. Offset Address: 0x14c Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	Reserved	0x0
[15:0]	RW	stream0_gb_dst_ch2_line_byte_stride	the line byte stride of output image channel 2	0x0

STREAM0_MAIN_CTRL_CFG21 (description) STREAM0_MAIN_CTRL_CFG21 stream0_border0~stream0_border31 draw border size register. Offset Address: 0x150~0x1cc Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[31:16]	RW	gb_height	the hight of the border	0x0
[15:0]	RW	gb_width	the width of the border	0x0

STREAM0_MAIN_CTRL_CFG22 (description) STREAM0_MAIN_CTRL_CFG22

stream0_border0~stream0_border31 the line wide register. Offset Address: 0x1d0~0x24c Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	Reserved	0x0
[15:0]	RW	gb_line_wide	the line wide of the border	0x0

STREAM0_MAIN_CTRL_CFG23 (description) STREAM0_MAIN_CTRL_CFG23

stream0_border0~stream0_border31 border color register. Offset Address: 0x250~0x2cc Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[31:0]	RW	gb_color	the color of the border	0x0

STREAM0_MAIN_CTRL_CFG24 (description) STREAM0_MAIN_CTRL_CFG24

stream0_border0~stream0_border31 border position register. Offset Address: 0x2d0~0x34c Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[31:16]	RW	gb_start_addry	The pixel offset of the upper left corner of the border in the y direction of the original image	0x0
[15:0]	RW	gb_start_addrx	The pixel offset of the upper left corner of the border in the x direction of the original image	0x0

STREAM0_MAIN_CTRL_CFG25 (description) STREAM0_MAIN_CTRL_CFG25 the register of image csc conversion coefficients. Offset Address: 0x350 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[31:28]	-	reserved	Reserved	0x0
[27:16]	RW	stream0_gb_csc_coef1	csc coefficient 1	0x0
[15:12]	-	reserved	Reserved	0x0
[11:0]	RW	stream0_gb_csc_coef0	csc coefficient 0	0x0

STREAM0_MAIN_CTRL_CFG26 (description) STREAM0_MAIN_CTRL_CFG26 the register of image csc conversion coefficients. Offset Address: 0x354 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[31:28]	-	reserved	Reserved	0x0
[27:16]	RW	stream0_gb_csc_coef3	csc coefficient 3	0x0
[15:12]	-	reserved	Reserved	0x0
[11:0]	RW	stream0_gb_csc_coef2	csc coefficient 2	0x0

STREAM0_MAIN_CTRL_CFG27 (description) STREAM0_MAIN_CTRL_CFG27 the register of image csc conversion coefficients. Offset Address: 0x358 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[31:28]	-	reserved	Reserved	0x0
[27:16]	RW	stream0_gb_csc_coef5	csc coefficient 5	0x0
[15:12]	-	reserved	Reserved	0x0
[11:0]	RW	stream0_gb_csc_coef4	csc coefficient 4	0x0

STREAM0_MAIN_CTRL_CFG28 (description) STREAM0_MAIN_CTRL_CFG28 the register of image csc conversion coefficients. Offset Address: 0x35c Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[31:28]	-	reserved	Reserved	0x0
[27:16]	RW	stream0_gb_csc_coef7	csc coefficient 7	0x0
[15:12]	-	reserved	Reserved	0x0
[11:0]	RW	stream0_gb_csc_coef6	csc coefficient 6	0x0

STREAM0_MAIN_CTRL_CFG29 (description) STREAM0_MAIN_CTRL_CFG29 the register of image csc conversion coefficients. Offset Address: 0x360 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[31:28]	-	reserved	Reserved	0x0
[27:16]	RW	stream0_gb_csc_coef9	csc coefficient 9	0x0
[15:12]	-	reserved	Reserved	0x0
[11:0]	RW	stream0_gb_csc_coef8	csc coefficient 8	0x0

STREAM0_MAIN_CTRL_CFG30 (description) STREAM0_MAIN_CTRL_CFG30 the register of image csc conversion coefficients. Offset Address: 0x364 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[31:28]	-	reserved	Reserved	0x0
[27:16]	RW	stream0_gb_csc_coef11	csc coefficient 11	0x0
[15:12]	-	reserved	Reserved	0x0
[11:0]	RW	stream0_gb_csc_coef10	csc coefficient 10	0x0

STREAM0_MAIN_CTRL_CFG31 (description) STREAM0_MAIN_CTRL_CFG31 the register of image osd conversion coefficients. Offset Address: 0x368 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset

Bits	Access	Name	Description	Reset
[31:28]	-	reserved	Reserved	0x0
[27:16]	RW	stream0_gb_osd_coef1	osd coefficient 1	0x0
[15:12]	-	reserved	Reserved	0x0
[11:0]	RW	stream0_gb_osd_coef0	osd coefficient 0	0x0

STREAM0_MAIN_CTRL_CFG32 (description) STREAM0_MAIN_CTRL_CFG32 the register of image osd conversion coefficients. Offset Address: 0x36c Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[31:28]	-	reserved	Reserved	0x0
[27:16]	RW	stream0_gb_osd_coef3	osd coefficient 3	0x0
[15:12]	-	reserved	Reserved	0x0
[11:0]	RW	stream0_gb_osd_coef2	osd coefficient 2	0x0

STREAM0_MAIN_CTRL_CFG33 (description) STREAM0_MAIN_CTRL_CFG33 the register of image osd conversion coefficients. Offset Address: 0x370 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[31:28]	-	reserved	Reserved	0x0
[27:16]	RW	stream0_gb_osd_coef5	osd coefficient 5	0x0
[15:12]	-	reserved	Reserved	0x0
[11:0]	RW	stream0_gb_osd_coef4	osd coefficient 4	0x0

STREAM0_MAIN_CTRL_CFG34 (description) STREAM0_MAIN_CTRL_CFG34 the register of image osd conversion coefficients. Offset Address: 0x374 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[31:28]	-	reserved	Reserved	0x0
[27:16]	RW	stream0_gb_osd_coef7	osd coefficient 7	0x0
[15:12]	-	reserved	Reserved	0x0
[11:0]	RW	stream0_gb_osd_coef6	osd coefficient 6	0x0

STREAM0_MAIN_CTRL_CFG35 (description) STREAM0_MAIN_CTRL_CFG35 the register of image osd conversion coefficients. Offset Address: 0x378 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[31:28]	-	reserved	Reserved	0x0
[27:16]	RW	stream0_gb_osd_coef9	osd coefficient 9	0x0

Bits	Access	Name	Description	Reset
[15:12]	-	reserved	Reserved	0x0
[11:0]	RW	stream0_gb_osd_coef8	osd coefficient 8	0x0

STREAM0_MAIN_CTRL_CFG36 (description) STREAM0_MAIN_CTRL_CFG36 the register of image osd conversion coefficients. Offset Address: 0x37c Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[31:28]	-	reserved	Reserved	0x0
[27:16]	RW	stream0_gb_osd_coef11	osd coefficient 11	0x0
[15:12]	-	reserved	Reserved	0x0
[11:0]	RW	stream0_gb_osd_coef10	osd coefficient 10	0x0

STREAM0_MAIN_CTRL_CFG37 (description) STREAM0_MAIN_CTRL_CFG37 The osd layer format of stream0_region0~stream0_region7. Offset Address: 0x380~0x39c Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[31:0]	RW	gb_osd_format	the format of the osd layer,0-argb8888; 1-argb4444; 2-argb1555; 3-xrgb8888; 4-xrgb4444; 5-xrgb1555; 6-bgra8888; 7-bgra4444; 8-bgra5551; 9-bgrx8888; 10-bgrx4444;11-bgrx5551 12-rgb888; 13-bgr888 ;14-rgb565; 15-bgr565; 16-seperate rgb	0x0

MAIN_CTRL_CFG31 (description) MAIN_CTRL_CFG31 configuration register. Offset Address: 0x3a0 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[31:17]	-	reserved	Reserved	0x0
[16]	RW	nonai_2d_intr_mask	mask the interrupt signal enable	0x0
[15:10]	RW	gb_draw_border_num	draw border number	0x0
[9:6]	RW	gb_osd_region_num	osd region number	0x0
[5:4]	RW	gb_nonai_2d_calc_mode	calculation mode 0-csc 1-osd 2-draw border 3-do osd first, then draw border	0x0
[3:2]	RW	gb_nonai_2d_calc_stream_id	the currently calculated stream id	0x0
[1]	RW	gb_csc_matrix_en	the CSC matrix enable	0x0
[0]	RWSC	gb_nonai_2d_calc_en	nonai 2d computing enable	0x0

MAIN_CTRL_CFG32 (description) MAIN_CTRL_CFG32 the number of watchdog timeout detection cycles. Offset Address: 0x3a4 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset

Bits	Access	Name	Description	Reset
[31:0]	RW	nonai_2d_wcd_num	The number of watchdog timeout detection cycles	0x0

MAIN_CTRL_CFG33 (description) MAIN_CTRL_CFG33 interrupt type. Offset Address: 0x3a8 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[31:16]	RO	nonai_2d_os_cnt	write bus outstanding counter	0x0
[15:0]	RO	nonai_2d_intr_type	interrupt type 0-normal interrupt 1-timeout interrupt 2-bus exception interrupt	0x0

MAIN_CTRL_CFG34 (description) MAIN_CTRL_CFG34 clear interrupt register. Offset Address: 0x3ac Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	Reserved	0x0
[0]	WO	nonai_2d_intr_clear	clear interrupt register	0x0

MAIN_CTRL_CFG35 (description) MAIN_CTRL_CFG35 nonai 2d internal self-cleaning busy. Offset Address: 0x3b0 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	Reserved	0x0
[0]	RO	nonai_2d_stop_busy	nonai 2d internal self-cleaning busy	0x0

MAIN_CTRL_CFG36 (description) MAIN_CTRL_CFG36 debug register. Offset Address: 0x3b4 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[31:0]	RO	nonai_2d_debug0	debug register	0x0

MAIN_CTRL_CFG37 (description) MAIN_CTRL_CFG37 debug register. Offset Address: 0x3b8 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[31:0]	RO	nonai_2d_debug1	debug register	0x0

MAIN_CTRL_CFG38 (description) MAIN_CTRL_CFG38 debug register. Offset Address: 0x3bc Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[31:0]	RO	nonai_2d_debug2	debug register	0x0

MAIN_CTRL_CFG39 (description) MAIN_CTRL_CFG39 clock gating register. Offset Address: 0x3bc Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	Reserved	0x0
[0]	RW	nonai_2d_disable_cg	disbale the inside clock gating	0x0

7.4.7 Interface Description

- The data interface is AXI master, which can read and write DDR at the same time
- The configuration interface is APB slave, which can issue configuration commands from the CPU
- The interrupt interface is a single-level signal. After the CPU receives the interrupt, it can query the current interrupt status of nonai2d by accessing the internal status register.

8 Video Interfaces

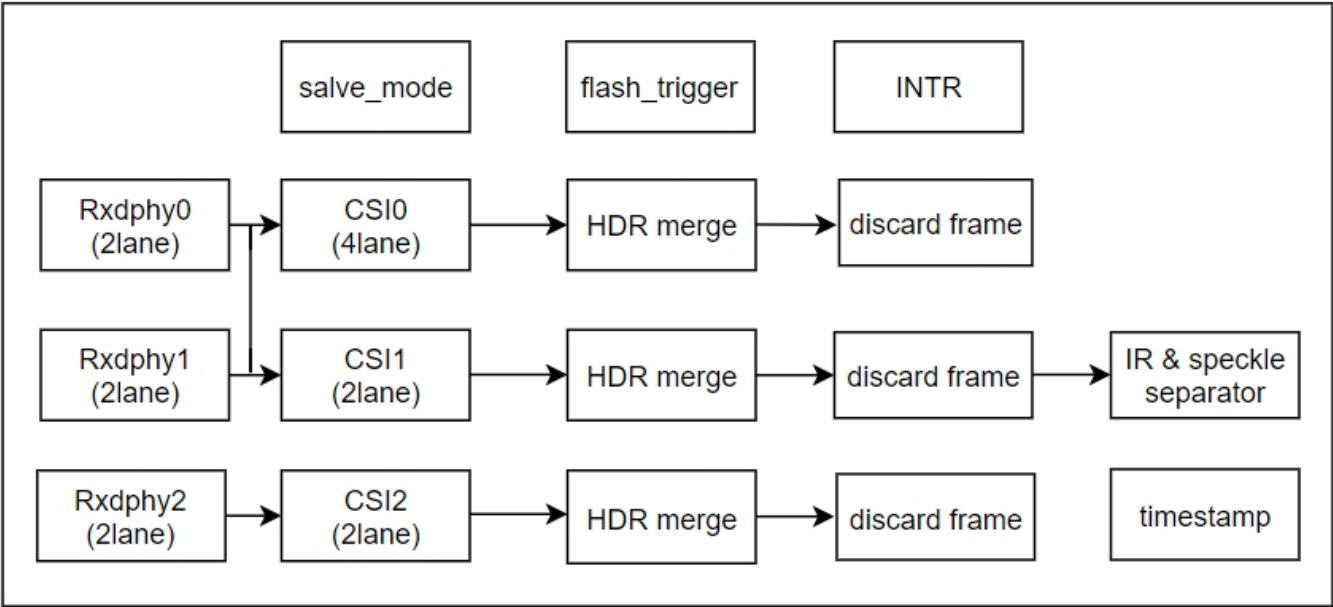
8.1 VI

8.1.1 Overview

VI(video input) receives the sensor data over the MIPI interface, and transfers the sensor data to ISP. It also supports the light source control and some data processing.

8.1.2 Block diagram

Figure 8-1-1 VI block diagram



8.1.3 Function description

8.1.3.1 Features

- Compatible with Synopsys MIPI RX DPHY and CSI-2 host controller, max up to three 2lane sensors or one 4lane sensor and one 2lane sensor.
- Support sensor slave mode
- Support flash trigger mode which support both follow-strobe mode and pwm mode.
- Support 3D sensor and can separate the IR data and speckle data to different DVP interface.
- Support timestamp.
- Support Sony DOL HDR sensor identification code.
- Support hardware discard frame.
- 8/10/12/16 Bit Bayer RAW.

8.1.3.2 MIPI transmission

VI supports up to three sensors. It may be three 2lane sensors or one 4lane sensor and one 2lane sensor. For the three 2lane sensors scenario, the diagram is shown as Figure 8-2.

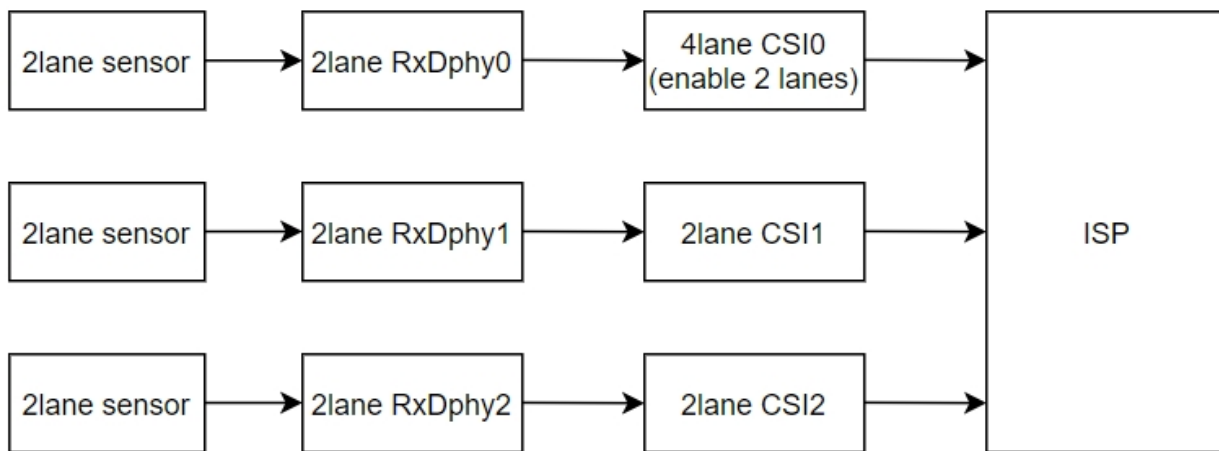


Figure 8-2 three 2lane sensors input

For the one 4lane sensor and one 2lane sensor, the diagram is shown as Figure 8-3.

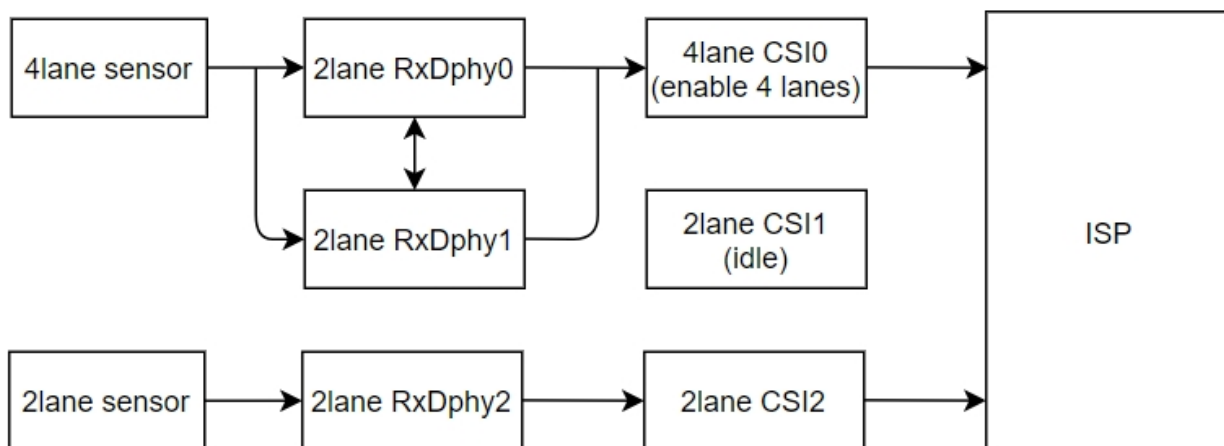


Figure 8-3 one 4lane sensor + one 2lane sensor input

If the sensor is 3D sensor, it can only be connected to MIPI RxDPHY1.

8.1.3.3 Slave mode

VI supports sensor dual-slave mode. Both sensors work in slave mode and VI provide the clock, vsync and hsync signals to sensors to control the synchronization of the two sensors. The diagram is shown as Figure 8-4.

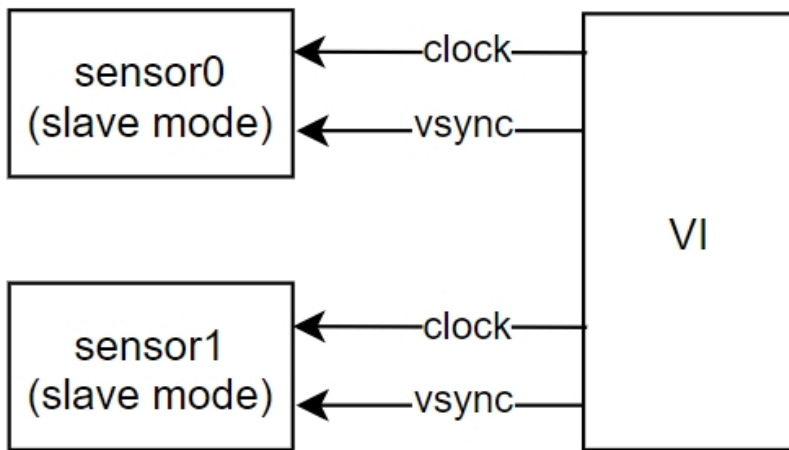


Figure 8-4 slave sensor mode

8.1.3.4 Flash trigger mode

VI supports the illuminant control of the 3D sensor. It supports the follow-strobe mode and strobe-based pwm mode.

- follow strobe mode for 3D sensor VI detect the strobe signals of 3D sensor and provide the flash_trigger_0 and flash_trigger_1 signals. The delay between the strobe signal and flash_trigger_0/1 signal doesn't exceed 1us. Also, the strobe detection supports deglitch. The timing is shown as Figure 8-5.

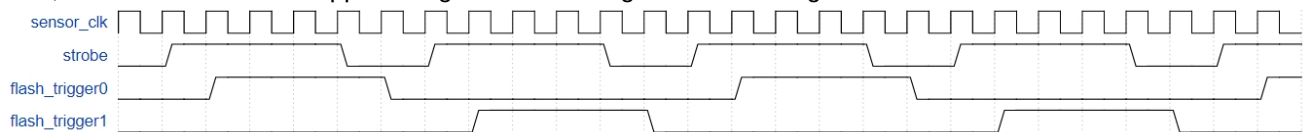


Figure 8-5 follow strobe mode for 3D sensor

- strobe-based pwm mode for 3D sensor In order to increase the flexibility of strobe, VI support strobe-based pwm mode. When the strobe signal is detected, VI generate the flash_trigger_0 signal via counter configured by software. And then generate the flash_trigger_1 signal according to the delay counter configured by software. In this mode, there is no need to follow the strobe signal except for the first one. The timing is shown as Figure 8-6.

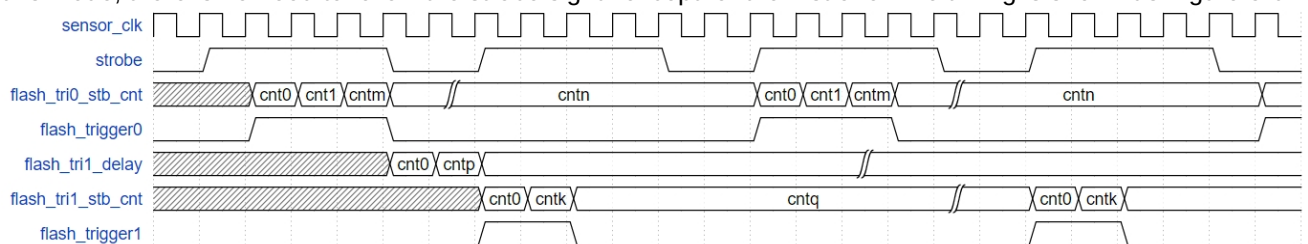


Figure 8-6 strobe-based pwm mode for 3D sensor

VI also support the flash trigger for non 3D sensor. It supports the follow-strobe mode and strobe-based pwm mode and normal pwm mode.

- follow strobe mode for non 3D sensor It is the same as follow strobe mode for 3D sensor except that only flash_trigger_0 is active.
- strobe-based pwm mode for non 3D sensor It is the same as strobe-based pwm mode for 3D sensor except that only flash_trigger_0 is active.
- normal pwm mode

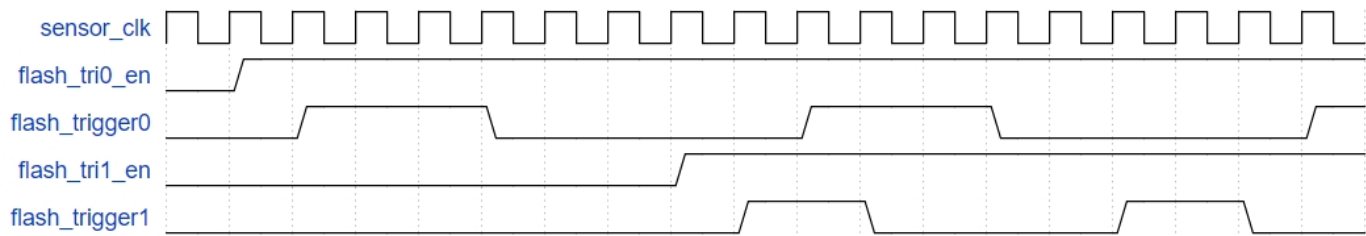


Figure 8-7 pwm mode

8.1.3.5 IR data and speckle data separator

VI support to separate the IR data and speckle data to different dvp interface based on the frame interrupt.

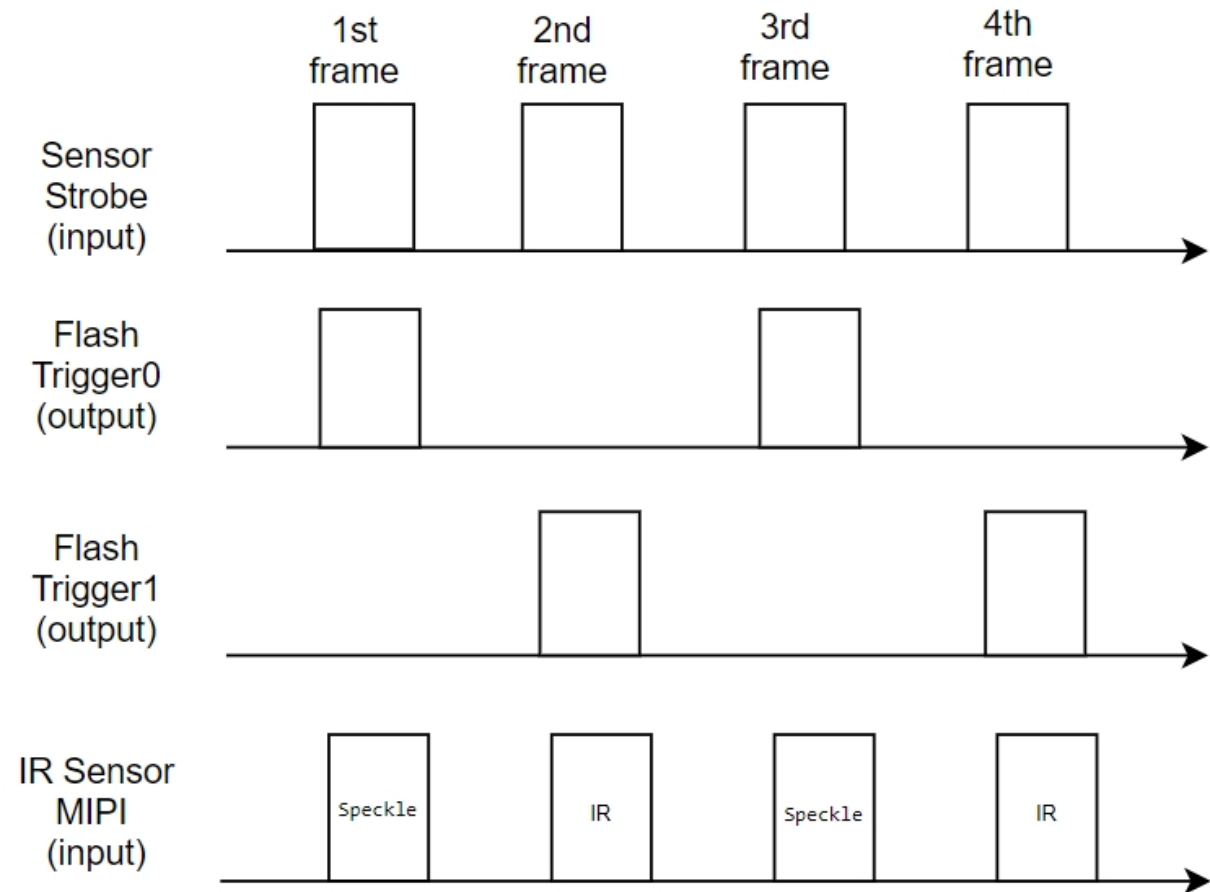


Figure 8-8 IR and speckle data separator

8.1.3.6 Timestamp

VI support the hardware timestamp of each frame which can be used to determine and synchronize the image from different cameras, as well as the subsequent video encoding.

When VI receive a frame of the image, the timestamp mechanism is triggered by the frame interrupt, and SYSCTRL capture the current STC value and write it to the captured STC value register. When the VI timestamp function is enabled, it automatically works after each frame of the image without the need for software to participate.

8.1.3.7 Sony HDR process

Some Sony DOL HDR sensors are different from the normal HDR sensors whose virtual channel for different exposure times are different. Since CSI only supports virtual channel mechanism, frame/line data with different virtual channel can be output through different pixel streams. But the Sony DOL HDR sensor can only be distinguished by the

identification code. In order to support these special Sony DOL HDR sensors, VI supports to parse the identification code to distinguish which kind of exposed frame data the current packet is.

8.1.3.8 Hardware discard frame

VI supports flexible frame extraction schemes. Several frames can be dropped every few frames, so as to achieve the purpose of flexible frame rate conversion. There is no additional bandwidth increase compared to ISP discarding frames or software discarding frames.

8.1.4 Interface description

Pins	I/O	Description
rext_0	IO	200 ohm External termination resistor
rext_1	IO	200 ohm External termination resistor.
atb_0	IO	Analog test bus pin
atb_0	IO	Analog test bus pin
Clkn_0	IO	Negative D-PHY Differential Clock Line for phy0
Clkp_0	IO	Positive D-PHY Differential Clock Line for phy0
Datan0_0	IO	Negative D-PHY Differential Data Line 0 for phy0
Datap0_0	IO	Positive D-PHY Differential Data Line 0 for phy0
Datan1_0	IO	Negative D-PHY Differential Data Line 1 for phy0
Datap1_0	IO	Positive D-PHY Differential Data Line 1 for phy0
Clkn_1	IO	Negative D-PHY Differential Clock Line for phy1
Clkp_1	IO	Positive D-PHY Differential Clock Line for phy1
Datan0_1	IO	Negative D-PHY Differential Data Line 0 for phy1
Datap0_1	IO	Positive D-PHY Differential Data Line 0 for phy1
Datan1_1	IO	Negative D-PHY Differential Data Line 1 for phy1
Datap1_1	IO	Positive D-PHY Differential Data Line 1 for phy2
Clkn_2	IO	Negative D-PHY Differential Clock Line for phy2
Clkp_2	IO	Positive D-PHY Differential Clock Line for phy2
Datan0_2	IO	Negative D-PHY Differential Data Line 0 for phy2
Datap0_2	IO	Positive D-PHY Differential Data Line 0 for phy2
Datan1_2	IO	Negative D-PHY Differential Data Line 1 for phy2
Datap1_2	IO	Positive D-PHY Differential Data Line 1 for phy2
sensor_strobe	I	sensor strobe signal
flash_trigger_0	O	flash trigger 0 signal
flash_trigger_1	O	flash trigger 1 signal
sensor0_vsync	O	vsync for sensor 0
sensor0_hsync	O	hsync for sensor 0

Pins	I/O	Description
sensor1_vsync	O	vsync for sensor 1
sensor1_hsync	O	hsync for sensor 1

8.1.5 Register summary

offset address	register	description
0x0000	RXDPHY0_CFG0	configuration register 0 for rxdphy0
0x0004	RXDPHY0_CFG1	configuration register 1 for rxdphy0
0x0008	RXDPHY1_CFG0	configuration register 0 for rxdphy1
0x000c	RXDPHY1_CFG1	configuration register 1 for rxdphy1
0x0010	RXDPHY2_CFG0	configuration register 0 for rxdphy2
0x0014	RXDPHY2_CFG1	configuration register 1 for rxdphy2
0x0018	CSI_CFG	configuration register for csi
0x001C	CSI_STA	status register for csi
0x0020	ISP_CFG	configuration register for isp
0x0024	SLA_MODE	slave mode configuration register
0x0028	SEN0_VS_CFG	vsync cycles for sensor0
0x002C	SEN0_HS_CFG	hsync cycles for sensor0
0x0030	SEN1_VS_CFG	vsync cycles for sensor1
0x0034	SEN1_HS_CFG	hsync cycles for sensor1
0x0038	SEN0_VS_HIGH	vsync high cycles for sensor0
0x003C	SEN0_HS_HIGH	hsync high cycles for sensor0
0x0040	SEN1_VS_HIGH	vsync high cycles for sensor1
0x0044	SEN1_HS_HIGH	vsync high cycles for sensor1
0x0048	LIGHT_CTL	light control register
0x004C	FLASH_TRIGGER_0_CFG_0	flash trigger0 inactivate level
0x0050	FLASH_TRIGGER_0_CFG_1	flash trigger0 activate level
0x0054	FLASH_TRIGGER_1_CFG_0	flash trigger1 inactivate level
0x0058	FLASH_TRIGGER_1_CFG_1	flash trigger1 activate level
0x005C	FLASH_TRIGGER_CFG	flash trigger configuration register
0x0060	IMG_CTL	dvp select register
0x0064	CAP_STC_VI_CFG	timestamp register
0x0068	VI_INTR_SET	vi interrupt set register
0x006C	VI_INTR_CLR	vi interrupt clear register
0x0070	VI_INTR_MASK	vi interrupt mask register

offset address	register	description
0x0074	VI_INTR_STATUS	vi interrupt status register
0x0078	VI_IPI_POL_CFG	ipi polarity register
0x007C	VI_CSI0_DIS_FRAME_M	discard frame register 0 for csi0
0x0080	VI_CSI0_DIS_FRAME_N	discard frame register 1 for csi0
0x0084	VI_CSI1_DIS_FRAME_M	discard frame register 0 for csi1
0x0088	VI_CSI1_DIS_FRAME_N	discard frame register 1 for csi1
0x008C	VI_CSI2_DIS_FRAME_M	discard frame register 0 for csi2
0x0090	VI_CSI2_DIS_FRAME_N	discard frame register 1 for csi2
0x0094	VI_DIS_FRAME_EN	discard frame enable register
0x0098	VI_SONY_CONFIG	configuration regitser for sony dol sensor
0x009C	VI_PHY_STA_CLR	phy status clear register
0x00A0	FLASH_TRIGGER_CFG_1	flash trigger configuration register 1
0x00A4	VI_SONY_CONFIG_1	configuration regitser 1 for sony dol sensor
0x00A0	VI_SONY_CONFIG_2	configuration regitser 2 for sony dol sensor
0x00AC	VI_HDR	hdr mode
0x00B0	VI_PHY_ENABLECLK	enableclk register for phy

8.1.6 Register description

Name=RXDPHY0_CFG0,offset=0x0

bits	Access Mode	Name	Description	Reset Value
31:25	RW	hsfreqrange_0	"High-speed Frequency Range Selection See Operating Frequency Range selection and default values. This signal should be set with the D-PHY in power-down and must not change during operation."	0x0
24	RW	forcetxstopmode_0_0	"Force Lane Module into Transmit Mode and Generate Stop State. This signal allows the protocol layer to force a bi-directional lane module into transmit mode and Stop state following an error indication (for example, Expired Timeout). When this signal is high and datalane0 direction is low, the lane module immediately transitions into transmit mode and the module state machine is forced into the Stop state."	0x0
23	RW	forcerxmode_1_0	"Force Lane Module into Receive Mode/Wait for Stop State for lane 1 This signal allows the protocol layer to initialize a lane module or force a bi-directional lane module into receive mode. This signal is used to solve a contention situation. When this signal is high, the lane module immediately transitions into receive control mode and waits for a Stop state to appear on the Lane interconnect"	0x0

bits	Access Mode	Name	Description	Reset Value
22	RW	forcerxmode_0_0	"Force Lane Module into Receive Mode/Wait for Stop State for lane 0 This signal allows the protocol layer to initialize a lane module or force a bi-directional lane module into receive mode. This signal is used to solve a contention situation. When this signal is high, the lane module immediately transitions into receive control mode and waits for a Stop state to appear on the Lane interconnect"	0x0
21	RO	errsyncesc_1_0	"Receiver Synchronization Observed for lane 1. This signal indicates that the lane module has seen an appropriate synchronization event. In a typical high-speed transmission rxsynchs_n is high for one cycle of rxbyteclkhs at the beginning of a high-speed transmission when rxactivehs_n is first asserted"	0x0
20	RO	errsyncesc_0_0	"Receiver Synchronization Observed for lane 0 This signal indicates that the lane module has seen an appropriate synchronization event. In a typical high-speed transmission rxsynchs_n is high for one cycle of rxbyteclkhs at the beginning of a high-speed transmission when rxactivehs_n is first asserted"	0x0
19	RO	errcontrol_1_0	"Control Error for lane 1 This signal is asserted when an incorrect line state sequence is detected. For example, if a turnaround request or Escape mode request is immediately followed by a Stop state instead of the required Bridge state, this signal is asserted and remains high until the line returns to Stop state"	0x0
18	RO	errcontrol_0_0	"Control Error for lane 0 This signal is asserted when an incorrect line state sequence is detected. For example, if a turnaround request or Escape mode request is immediately followed by a Stop state instead of the required Bridge state, this signal is asserted and remains high until the line returns to Stop state"	0x0
17	RO	errcontentionlp1_0_0	"LP1 Contention Error This signal is asserted when a lane module functioning as TX, while its base direction is RX detects a contention situation on a line while trying to drive the line high"	0x0
16	RO	errcontentionlp0_0_0	"LP0 Contention Error This signal is asserted when a lane module functioning as TX, while its base direction is RX detects a contention situation on a line while trying to drive the line low"	0x0
15	RO	direction_0_0	"Transmit/Receive Direction This signal is used to indicate the current direction of the lane interconnect. When direction is low, the lane interconnect is in transmit mode. When direction is high, the lane interconnect is in receive mode."	0x0

bits	Access Mode	Name	Description	Reset Value
14:8	RO	cont_data_0	"Each pin of this output bus is used to verify the continuity from the chips pin all the way to the PHYs analog ports (package, bonding, die, among others). cont_data output signals should be ignored when cont_en is deasserted (PHY not set for IO Continuity Test mode).cont_data[0]: rext cont_data[1]: clkn cont_data[2]: clkp cont_data[3]: datan0 cont_data[4]: datap0 cont_data[5]: datan1 cont_data[6]: datap1 This signal can be overridden through D-PHY test control registers."	0x0
7	RW	cont_en_0	"This signal places the macro in IO continuity test mode. All other interface signals should be placed in their default values."	0x0
6:1	RW	cfgclkfreqrage_0	"System clock frequency configuration preset. Should be set according to the following equation: $\text{cfgclkfreqrage}[5:0] = \text{round}[(\text{Fcfg\_clk}(\text{MHz})-17)*4]$ "	0x0
0	RW	basedir_0_0	"Configures the Base Direction for Lane basedir_0 = 1: Configures lane as RX upon startup of the PHY basedir_0 = 0: Configures lane as TX"	0x1

Name=RXDPHY0_CFG1,offset=0x4

bits	Access Mode	Name	Description	Reset Value
14	RO	ulpsactivenot_0_0_inv		0x0
13	RO	ulpsactivenot_1_0_inv		0x0
12	RO	ulpsactivenotclk_0_inv		0x0
11:8	RO	rxtriggeresc_1_0	"Escape Mode Receive Trigger 0-3 This signal indicates that a trigger event has been received in the lane interconnect. The asserted rxtriggeresc_n signal remains active until a Stop state is detected on the lane interconnect. rxtriggeresc_n[0] corresponds to Reset-Trigger. rxtriggeresc_n[1] corresponds to Unknown-3 Trigger. rxtriggeresc_n[2] corresponds to Unknown-4 Trigger. rxtriggeresc_n[3] corresponds to Unknown-5 Trigger."	0x0
7:4	RO	rxtriggeresc_0_0	"Escape Mode Receive Trigger 0-3 This signal indicates that a trigger event has been received in the lane interconnect. The asserted rxtriggeresc_n signal remains active until a Stop state is detected on the lane interconnect. rxtriggeresc_n[0] corresponds to Reset-Trigger. rxtriggeresc_n[1] corresponds to Unknown-3 Trigger. rxtriggeresc_n[2] corresponds to Unknown-4 Trigger. rxtriggeresc_n[3] corresponds to Unknown-5 Trigger."	0x0

bits	Access Mode	Name	Description	Reset Value
3	RO	rxsynchs_1_0	"Receiver Synchronization Observed for lane 1 This signal indicates that the lane module has seen an appropriate synchronization event. In a typical high-speed transmission rxsynchs_n is high for one cycle of rxbyteclkhs at the beginning of a high-speed transmission when rxactivehs_n is first asserted"	0x0
2	RO	rxsynchs_0_0	"Receiver Synchronization Observed for lane 0 This signal indicates that the lane module has seen an appropriate synchronization event. In a typical high-speed transmission rxsynchs_n is high for one cycle of rxbyteclkhs at the beginning of a high-speed transmission when rxactivehs_n is first asserted"	0x0
1	RO	rxactivehs_1_0	"High-Speed Reception Active for lane 1 This signal indicates that the lane module is actively receiving a high speed data transmission from the lane interconnect."	0x0
0	RO	rxactivehs_0_0	"High-Speed Reception Active for lane 0 This signal indicates that the lane module is actively receiving a high speed data transmission from the lane interconnect."	0x0

Name=RXDPHY1_CFG0,offset=0x8

bits	Access Mode	Name	Description	Reset Value
31:25	RW	hsfreqrange_1	"High-speed Frequency Range Selection See Operating Frequency Range selection and default values. This signal should be set with the D-PHY in power-down and must not change during operation."	0x0
24	RW	forcetxstopmode_0_1	"Force Lane Module into Transmit Mode and Generate Stop State This signal allows the protocol layer to force a bi-directional lane module into transmit mode and Stop state following an error indication (for example, Expired Timeout). When this signal is high and datalane0 direction is low, the lane module immediately transitions into transmit mode and the module state machine is forced into the Stop state."	0x0
23	RW	forcerxmode_1_1	"Force Lane Module into Receive Mode/Wait for Stop State for lane 1 This signal allows the protocol layer to initialize a lane module or force a bi-directional lane module into receive mode. This signal is used to solve a contention situation. When this signal is high, the lane module immediately transitions into receive control mode and waits for a Stop state to appear on the Lane interconnect"	0x0

bits	Access Mode	Name	Description	Reset Value
22	RW	forcerxmode_0_1	"Force Lane Module into Receive Mode/Wait for Stop State for lane 0 This signal allows the protocol layer to initialize a lane module or force a bi-directional lane module into receive mode. This signal is used to solve a contention situation. When this signal is high, the lane module immediately transitions into receive control mode and waits for a Stop state to appear on the Lane interconnect"	0x0
21	RO	errsyncesc_1_1	"Receiver Synchronization Observed for lane 1 This signal indicates that the lane module has seen an appropriate synchronization event. In a typical high-speed transmission rxsynchs_n is high for one cycle of rxbyteclkhs at the beginning of a high-speed transmission when rxactivehs_n is first asserted"	0x0
20	RO	errsyncesc_0_1	"Receiver Synchronization Observed for lane 0 This signal indicates that the lane module has seen an appropriate synchronization event. In a typical high-speed transmission rxsynchs_n is high for one cycle of rxbyteclkhs at the beginning of a high-speed transmission when rxactivehs_n is first asserted"	0x0
19	RO	errcontrol_1_1	"Control Error for lane 1 This signal is asserted when an incorrect line state sequence is detected. For example, if a turnaround request or Escape mode request is immediately followed by a Stop state instead of the required Bridge state, this signal is asserted and remains high until the line returns to Stop state"	0x0
18	RO	errcontrol_0_1	"Control Error for lane 0 This signal is asserted when an incorrect line state sequence is detected. For example, if a turnaround request or Escape mode request is immediately followed by a Stop state instead of the required Bridge state, this signal is asserted and remains high until the line returns to Stop state"	0x0
17	RO	errcontentionlp1_0_1	"LP1 Contention Error This signal is asserted when a lane module functioning as TX, while its base direction is RX detects a contention situation on a line while trying to drive the line high"	0x0
16	RO	errcontentionlp0_0_1	"LP0 Contention Error This signal is asserted when a lane module functioning as TX, while its base direction is RX detects a contention situation on a line while trying to drive the line low"	0x0
15	RO	direction_0_1	"Transmit/Receive Direction This signal is used to indicate the current direction of the lane interconnect. When direction is low, the lane interconnect is in transmit mode. When direction is high, the lane interconnect is in receive mode."	0x0

bits	Access Mode	Name	Description	Reset Value
14:8	RO	cont_data_1	"Each pin of this output bus is used to verify the continuity from the chips pin all the way to the PHYs analog ports (package, bonding, die, among others). cont_data output signals should be ignored when cont_en is deasserted (PHY not set for IO Continuity Test mode). cont_data[0]: rext cont_data[1]: clkn cont_data[2]: clkp cont_data[3]: datan0 cont_data[4]: datap0 cont_data[5]: datan1 cont_data[6]: datap1 This signal can be overridden through D-PHY test control registers."	0x0
7	RW	cont_en_1	"This signal places the macro in IO continuity test mode. All other interface signals should be placed in their default values."	0x0
6:1	RW	cfgclkfreqrage_1	"System clock frequency configuration preset. Should be set according to the following equation: $\text{cfgclkfreqrage}[5:0] = \text{round}[(\text{Fcfg\_clk}(\text{MHz}) - 17) * 4]$ "	0x0
0	RW	basedir_0_1	"Configures the Base Direction for Lane basedir_0 = 1: Configures lane as RX upon startup of the PHY basedir_0 = 0: Configures lane as TX"	0x1

Name=RXDPHY1_CFG1,offset=0xc

bits	Access Mode	Name	Description	Reset Value
14	RO	ulpsactivenot_0_1_inv		0x0
13	RO	ulpsactivenot_1_1_inv		0x0
12	RO	ulpsactivenotclk_1_inv		0x0
11:8	RO	rxtriggeresc_1_1	"Escape Mode Receive Trigger 0-3 This signal indicates that a trigger event has been received in the lane interconnect. The asserted rxtriggeresc_n signal remains active until a Stop state is detected on the lane interconnect. rxtriggeresc_n[0] corresponds to Reset-Trigger. rxtriggeresc_n[1] corresponds to Unknown-3 Trigger. rxtriggeresc_n[2] corresponds to Unknown-4 Trigger. rxtriggeresc_n[3] corresponds to Unknown-5 Trigger."	0x0
7:4	RO	rxtriggeresc_0_1	"Escape Mode Receive Trigger 0-3 This signal indicates that a trigger event has been received in the lane interconnect. The asserted rxtriggeresc_n signal remains active until a Stop state is detected on the lane interconnect. rxtriggeresc_n[0] corresponds to Reset-Trigger. rxtriggeresc_n[1] corresponds to Unknown-3 Trigger. rxtriggeresc_n[2] corresponds to Unknown-4 Trigger. rxtriggeresc_n[3] corresponds to Unknown-5 Trigger."	0x0

bits	Access Mode	Name	Description	Reset Value
3	RO	rxsynchs_1_1	"Receiver Synchronization Observed for lane 1 This signal indicates that the lane module has seen an appropriate synchronization event. In a typical high-speed transmission rxsynchs_n is high for one cycle of rxbyteclkhs at the beginning of a high-speed transmission when rxactivehs_n is first asserted"	0x0
2	RO	rxsynchs_0_1	"Receiver Synchronization Observed for lane 0 This signal indicates that the lane module has seen an appropriate synchronization event. In a typical high-speed transmission rxsynchs_n is high for one cycle of rxbyteclkhs at the beginning of a high-speed transmission when rxactivehs_n is first asserted"	0x0
1	RO	rxactivehs_1_1	"High-Speed Reception Active for lane 1 This signal indicates that the lane module is actively receiving a high speed data transmission from the lane interconnect."	0x0
0	RO	rxactivehs_0_1	"High-Speed Reception Active for lane 0 This signal indicates that the lane module is actively receiving a high speed data transmission from the lane interconnect."	0x0

Name=RXDPHY2_CFG0,offset=0x10

bits	Access Mode	Name	Description	Reset Value
31:25	RW	hsfreqrange_2	"High-speed Frequency Range Selection See Operating Frequency Range selection and default values. This signal should be set with the D-PHY in power-down and must not change during operation."	0x0
24	RW	forcetxstopmode_0_2	"Force Lane Module into Transmit Mode and Generate Stop State This signal allows the protocol layer to force a bi-directional lane module into transmit mode and Stop state following an error indication (for example, Expired Timeout). When this signal is high and datalane0 direction is low, the lane module immediately transitions into transmit mode and the module state machine is forced into the Stop state."	0x0
23	RW	forcerxmode_1_2	"Force Lane Module into Receive Mode/Wait for Stop State for lane 1 This signal allows the protocol layer to initialize a lane module or force a bi-directional lane module into receive mode. This signal is used to solve a contention situation. When this signal is high, the lane module immediately transitions into receive control mode and waits for a Stop state to appear on the Lane interconnect"	0x0

bits	Access Mode	Name	Description	Reset Value
22	RW	forcerxmode_0_2	"Force Lane Module into Receive Mode/Wait for Stop State for lane 0 This signal allows the protocol layer to initialize a lane module or force a bi-directional lane module into receive mode. This signal is used to solve a contention situation. When this signal is high, the lane module immediately transitions into receive control mode and waits for a Stop state to appear on the Lane interconnect"	0x0
21	RO	errsyncesc_1_2	"Receiver Synchronization Observed for lane 1 This signal indicates that the lane module has seen an appropriate synchronization event. In a typical high-speed transmission rxsynchs_n is high for one cycle of rxbyteclkhs at the beginning of a high-speed transmission when rxactivehs_n is first asserted"	0x0
20	RO	errsyncesc_0_2	"Receiver Synchronization Observed for lane 0 This signal indicates that the lane module has seen an appropriate synchronization event. In a typical high-speed transmission rxsynchs_n is high for one cycle of rxbyteclkhs at the beginning of a high-speed transmission when rxactivehs_n is first asserted"	0x0
19	RO	errcontrol_1_2	"Control Error for lane 1 This signal is asserted when an incorrect line state sequence is detected. For example, if a turnaround request or Escape mode request is immediately followed by a Stop state instead of the required Bridge state, this signal is asserted and remains high until the line returns to Stop state"	0x0
18	RO	errcontrol_0_2	"Control Error for lane 0 This signal is asserted when an incorrect line state sequence is detected. For example, if a turnaround request or Escape mode request is immediately followed by a Stop state instead of the required Bridge state, this signal is asserted and remains high until the line returns to Stop state"	0x0
17	RO	errcontentionlp1_0_2	"LP1 Contention Error This signal is asserted when a lane module functioning as TX, while its base direction is RX detects a contention situation on a line while trying to drive the line high"	0x0
16	RO	errcontentionlp0_0_2	"LP0 Contention Error This signal is asserted when a lane module functioning as TX, while its base direction is RX detects a contention situation on a line while trying to drive the line low"	0x0
15	RO	direction_0_2	"Transmit/Receive Direction This signal is used to indicate the current direction of the lane interconnect. When direction is low, the lane interconnect is in transmit mode. When direction is high, the lane interconnect is in receive mode."	0x0

bits	Access Mode	Name	Description	Reset Value
14:8	RO	cont_data_2	"Each pin of this output bus is used to verify the continuity from the chips pin all the way to the PHYs analog ports (package, bonding, die, among others). cont_data output signals should be ignored when cont_en is deasserted (PHY not set for IO Continuity Test mode). cont_data[0]: rext cont_data[1]: clkn cont_data[2]: clkp cont_data[3]: datan0 cont_data[4]: datap0 cont_data[5]: datan1 cont_data[6]: datap1 This signal can be overridden through D-PHY test control registers."	0x0
7	RW	cont_en_2	"This signal places the macro in IO continuity test mode. All other interface signals should be placed in their default values."	0x0
6:1	RW	cfgclkfreqrage_2	"System clock frequency configuration preset. Should be set according to the following equation: $\text{cfgclkfreqrage}[5:0] = \text{round}[(\text{Fcfg\_clk}(\text{MHz}) - 17) * 4]$ "	0x0
0	RW	basedir_0_2	"Configures the Base Direction for Lane basedir_0 = 1: Configures lane as RX upon startup of the PHY basedir_0 = 0: Configures lane as TX"	0x1

Name=RXDPHY2_CFG1,offset=0x14

bits	Access Mode	Name	Description	Reset Value
14	RO	ulpsactivenot_0_2_inv		0x0
13	RO	ulpsactivenot_1_2_inv		0x0
12	RO	ulpsactivenotclk_2_inv		0x0
11:8	RO	rxtriggeresc_1_2	"Escape Mode Receive Trigger 0-3 This signal indicates that a trigger event has been received in the lane interconnect. The asserted rxtriggeresc_n signal remains active until a Stop state is detected on the lane interconnect. rxtriggeresc_n[0] corresponds to Reset-Trigger. rxtriggeresc_n[1] corresponds to Unknown-3 Trigger. rxtriggeresc_n[2] corresponds to Unknown-4 Trigger. rxtriggeresc_n[3] corresponds to Unknown-5 Trigger."	0x0
7:4	RO	rxtriggeresc_0_2	"Escape Mode Receive Trigger 0-3 This signal indicates that a trigger event has been received in the lane interconnect. The asserted rxtriggeresc_n signal remains active until a Stop state is detected on the lane interconnect. rxtriggeresc_n[0] corresponds to Reset-Trigger. rxtriggeresc_n[1] corresponds to Unknown-3 Trigger. rxtriggeresc_n[2] corresponds to Unknown-4 Trigger. rxtriggeresc_n[3] corresponds to Unknown-5 Trigger."	0x0

bits	Access Mode	Name	Description	Reset Value
3	RO	rxsynchs_1_2	"Receiver Synchronization Observed for lane 1 This signal indicates that the lane module has seen an appropriate synchronization event. In a typical high-speed transmission rxsynchs_n is high for one cycle of rxbyteclkhs at the beginning of a high-speed transmission when rxactivehs_n is first asserted"	0x0
2	RO	rxsynchs_0_2	"Receiver Synchronization Observed for lane 0 This signal indicates that the lane module has seen an appropriate synchronization event. In a typical high-speed transmission rxsynchs_n is high for one cycle of rxbyteclkhs at the beginning of a high-speed transmission when rxactivehs_n is first asserted"	0x0
1	RO	rxactivehs_1_2	"High-Speed Reception Active for lane 1 This signal indicates that the lane module is actively receiving a high speed data transmission from the lane interconnect."	0x0
0	RO	rxactivehs_0_2	"High-Speed Reception Active for lane 0 This signal indicates that the lane module is actively receiving a high speed data transmission from the lane interconnect."	0x0

Name=CSI_CFG,offset=0x18

bits	Access Mode	Name	Description	Reset Value
11	RW	phy_shutdown_sel	reserved	0x0
10	RW	phy_test_sel	0: test for phy0 1:test for phy1	0x0
9	RW	lane_num	1:2lane 0:4lane	0x1
8	RW	ipi_halt_3if_2	IPI2 ipi halt for phy2	0x0
7	RW	ipi_halt_2if_2	IPI1 ipi halt for phy2	0x0
6	RW	ipi_halt_2	IPI0 ipi halt for phy2	0x0
5	RW	ipi_halt_3if_1	IPI2 ipi halt for phy1	0x0
4	RW	ipi_halt_2if_1	IPI1 ipi halt for phy1	0x0
3	RW	ipi_halt_1	IPI0 ipi halt for phy1	0x0
2	RW	ipi_halt_3if_0	IPI2 ipi halt for phy0	0x0
1	RW	ipi_halt_2if_0	IPI1 ipi halt for phy0	0x0
0	RW	ipi_halt_0	IPI0 ipi halt for phy0	0x0

Name=CSI_STA,offset=0x1c

bits	Access Mode	Name	Description	Reset Value
8	RO	ipi_embedded_3if_2	IPI2 is outputting embedded non-image data for phy2	0x0

bits	Access Mode	Name	Description	Reset Value
7	RO	ipi_embedded_2if_2	IPI1 is outputting embedded non-image data for phy2	0x0
6	RO	ipi_embedded_2	IPI0 is outputting embedded non-image data for phy2	0x0
5	RO	ipi_embedded_3if_1	IPI2 is outputting embedded non-image data for phy1	0x0
4	RO	ipi_embedded_2if_1	IPI1 is outputting embedded non-image data for phy1	0x0
3	RO	ipi_embedded_1	IPI0 is outputting embedded non-image data for phy1	0x0
2	RO	ipi_embedded_3if_0	IPI2 is outputting embedded non-image data for phy0	0x0
1	RO	ipi_embedded_2if_0	IPI1 is outputting embedded non-image data for phy0	0x0
0	RO	ipi_embedded_0	IPI0 is outputting embedded non-image data for phy0	0x0

Name=ISP_CFG,offset=0x20

bits	Access Mode	Name	Description	Reset Value
0	RW	disable_isp	disable isp output	0x0

Name=SLA_MODE_CFG,offset=0x24

bits	Access Mode	Name	Description	Reset Value
31:20	RW	Sensor1_HS_DLY	Sensor 1 HS delay	0x0
19	RW	Sensor1_HS_INV	"Sensor 1 HS polarity 0:high 1:low"	0x0
18	RW	Sensor1_VS_INV	"Sensor 1 VS polarity 0:high 1:low"	0x0
17	RW	Sensor1_HS_EN	"Sensor 1 HS enable 0:disable 1:enable"	0x0
16	RW	Sensor1_VS_EN	"Sensor 1 VS enable 0:disable 1:enable"	0x0
15:4	RW	Sensor0_HS_DLY	Sensor 0 HS delay	0x0
3	RW	Sensor0_HS_INV	"Sensor 0 HS polarity 0:high 1:low"	0x0
2	RW	Sensor0_VS_INV	"Sensor 0 VS plarity 0:high 1:low"	0x0
1	RW	Sensor0_HS_EN	"Sensor 0 HS enable 0:disable 1:enable"	0x0
0	RW	Sensor0_VS_EN	"Sensor 0 VS enable 0:disable 1:enable"	0x0

Name=SEN0_VS_CFG,offset=0x28

bits	Access Mode	Name	Description	Reset Value
31:0	RW	sensor_0_vs_cycle	VS cycles	0x0

Name=SEN0_HS_CFG,offset=0x2c

bits	Access Mode	Name	Description	Reset Value
31:0	RW	sensor_0_hs_cycle	HS cycles	0x0

Name=SEN1_VS_CFG,offset=0x30

bits	Access Mode	Name	Description	Reset Value
31:0	RW	sensor_1_vs_cycle	VS cycles	0x0

Name=SEN1_HS_CFG,offset=0x34

bits	Access Mode	Name	Description	Reset Value
31:0	RW	sensor_1_hs_cycle	HS cycles	0x0

Name=SEN0_VS_HIGH,offset=0x38

bits	Access Mode	Name	Description	Reset Value
31:0	RW	sensor_0_vs_high	VS high	0x0

Name=SEN0_HS_HIGH,offset=0x3c

bits	Access Mode	Name	Description	Reset Value
31:0	RW	sensor_0_hs_high	HS high	0x0

Name=SEN1_VS_HIGH,offset=0x40

bits	Access Mode	Name	Description	Reset Value
31:0	RW	sensor_1_vs_high	VS high	0x0

Name=SEN1_HS_HIGH,offset=0x44

bits	Access Mode	Name	Description	Reset Value
31:0	RW	sensor_1_hs_high	HS high	0x0

Name=LIGHT_CTL,offset=0x48

bits	Access Mode	Name	Description	Reset Value
31	RW	ctl_en	"0:disable 3D struct 1:enable"	0x0
30	RW	strobe_inv	"0: strobe high active 1: strobe low active"	0x0
29	RW	flash_trigger_0_inv	"0: strobe high active 1:strobe low active"	0x0
28	RW	flash_trigger_1_inv	"0:strobe high active 1:strobe low active"	0x0
27:16	RW	clk_div	clock div	0x0
15:0	RW	glitch_filter	0 default disable glitch filter	0x0

Name=FLASH_TRIGGER_0_CFG_0,offset=0x4C

bits	Access Mode	Name	Description	Reset Value
31:0	RW	flash_trigger_0_cfg_0	flash trigger 0 active level time default: low level	0x0

Name=FLASH_TRIGGER_0_CFG_1,offset=0x50

bits	Access Mode	Name	Description	Reset Value
31:0	RW	flash_trigger_0_cfg_1	flash trigger 0 inactive level time default: high level	0x0

Name=FLASH_TRIGGER_1_CFG_0,offset=0x54

bits	Access Mode	Name	Description	Reset Value
31:0	RW	flash_trigger_1_cfg_0	flash trigger 1 active level time default: low level	0x0

Name=FLASH_TRIGGER_1_CFG_1,offset=0x58

bits	Access Mode	Name	Description	Reset Value
31:0	RW	flash_trigger_1_cfg_1	flash trigger 1 inactive level time default: high level	0x0

Name=FLASH_TRIGGER_CFG,offset=0x5c

bits	Access Mode	Name	Description	Reset Value
31:4	RW	flash_trigger1_delay	flash trigger1 delay	0x0
3	RW	flash_trigger_1_en	Flash trigger_1 enable	0x0
2	RW	flash_trigger_0_en	Flash trigger_0 enable	0x0
1:0	RW	flash_trigger_mode	00: follow strobe mode; 01: strobe-based pwm mode 10: pwm mode	0x0

Name=IMG_CTL,offset=0x60

bits	Access Mode	Name	Description	Reset Value
9	RW	first_frame	"0: first frame is Flash_Trigger_0 1: first frame is Flash_Trigger_1"	0x0
8:6	RW	dvp3_sel	"000:CS00 001:CSI1 010:CSI1 flash trigger0 011:CSI1 flash trigger1 100: CSI2"	0x4
5:3	RW	dvp2_sel	"000:CS00 001:CSI1 010:CSI1 flash trigger0 011:CSI1 flash trigger1 100: CSI2"	0x1
2:0	RW	dvp1_sel	"000:CS00 001:CSI1 010:CSI1 flash trigger0 011:CSI1 flash trigger1 100: CSI2"	0x0

Name=CAP_STC_VI_CFG,offset=0x64

bits	Access Mode	Name	Description	Reset Value
8	RW	cap_stc_csi2_clr	cap_stc_csi2_clr	0x0
7	RW	cap_stc_csi2_set	cap_stc_csi2_set	0x0

bits	Access Mode	Name	Description	Reset Value
6	RW	cap_stc_csi2_en	cap_stc_csi2_en	0x0
5	RW	cap_stc_csi1_clr	cap_stc_csi1_clr	0x0
4	RW	cap_stc_csi1_set	cap_stc_csi1_set	0x0
3	RW	cap_stc_csi1_en	cap_stc_csi1_en	0x0
2	RW	cap_stc_csi0_clr	cap_stc_csi0_clr	0x0
1	RW	cap_stc_csi0_set	cap_stc_csi0_set	0x0
0	RW	cap_stc_csi0_en	cap_stc_csi0_en	0x0

Name=VI_INTR_SET,offset=0x68

bits	Access Mode	Name	Description	Reset Value
6	RW	flash_tri1_strobe_miss_intr_set	flash_tri1_strobe_miss_intr_set	0x0
5	RW	flash_tri0_strobe_miss_intr_set	flash_tri0_strobe_miss_intr_set	0x0
4	RW	flash_tri1_frame_miss_intr_set	flash_tri1_frame_miss_intr_set	0x0
3	RW	flash_tri0_frame_miss_intr_set	flash_tri0_frame_miss_intr_set	0x0
2	RW	flash_trigger1_intr_set	flash_trigger1_intr_set	0x0
1	RW	flash_trigger0_intr_set	flash_trigger0_intr_set	0x0
0	RW	strobe_intr_set	strobe_intr_set	0x0

Name=VI_INTR_CLR,offset=0x6C

bits	Access Mode	Name	Description	Reset Value
8	RW	flash_tri1_vsync_cnt_clr	flash_tri1_vsync_cnt_clr	0x0
7	RW	flash_tri0_vsync_cnt_clr	flash_tri0_vsync_cnt_clr	0x0
6	RW	flash_tri1_strobe_miss_intr_clr	flash_tri1_strobe_miss_intr_clr	0x0
5	RW	flash_tri0_strobe_miss_intr_clr	flash_tri0_strobe_miss_intr_clr	0x0
4	RW	flash_tri1_frame_miss_intr_clr	flash_tri1_frame_miss_intr_clr	0x0
3	RW	flash_tri0_frame_miss_intr_clr	flash_tri0_frame_miss_intr_clr	0x0
2	RW	flash_trigger1_intr_clr	flash_trigger1_intr_clr	0x0
1	RW	flash_trigger0_intr_clr	flash_trigger0_intr_clr	0x0
0	RW	strobe_intr_clr	strobe_intr_clr	0x0

Name=VI_INTR_MASK,offset=0x70

bits	Access Mode	Name	Description	Reset Value
6	RW	flash_tri1_strobe_miss_intr_mask	flash_tri1_strobe_miss_intr_mask	0x0
5	RW	flash_tri0_strobe_miss_intr_mask	flash_tri0_strobe_miss_intr_mask	0x0
4	RW	flash_tri1_frame_miss_intr_mask	flash_tri1_frame_miss_intr_mask	0x0

bits	Access Mode	Name	Description	Reset Value
3	RW	flash_tri0_frame_miss_intr_mask	flash_tri0_frame_miss_intr_mask	0x0
2	RW	flash_trigger1_intr_mask	flash_trigger1_intr_mask	0x0
1	RW	flash_trigger0_intr_mask	flash_trigger0_intr_mask	0x0
0	RW	strobe_intr_mask	strobe_intr_mask	0x0

Name=VI_INTR_STATUS,offset=0x74

bits	Access Mode	Name	Description	Reset Value
6	RO	flash_tri1_strobe_miss_intr_sta	flash_tri1_strobe_miss_intr_sta	0x0
5	RO	flash_tri0_strobe_miss_intr_sta	flash_tri0_strobe_miss_intr_sta	0x0
4	RO	flash_tri1_frame_miss_intr_sta	flash_tri1_frame_miss_intr_sta	0x0
3	RO	flash_tri0_frame_miss_intr_sta	flash_tri0_frame_miss_intr_sta	0x0
2	RO	flash_trigger1_intr_sta	flash_trigger1_intr_sta	0x0
1	RO	flash_trigger0_intr_sta	flash_trigger0_intr_sta	0x0
0	RO	strobe_intr_sta	strobe_intr_sta	0x0

Name=VI_IPI_POL_CFG,offset=0x78

bits	Access Mode	Name	Description	Reset Value
5	RW	ipi_hsync_csi2_pol	0: high 1: low	0x0
4	RW	ipi_vsync_csi2_pol	0: pos edge 1: neg edge	0x0
3	RW	ipi_hsync_csi1_pol	0: high 1: low	0x0
2	RW	ipi_vsync_csi1_pol	0: pos edge 1: neg edge	0x0
1	RW	ipi_hsync_csi0_pol	0: high 1: low	0x0
0	RW	ipi_vsync_csi0_pol	0: pos edge 1: neg edge	0x0

Name=VI_CSI0_DIS_FRAME_M,offset=0x7c

bits	Access Mode	Name	Description	Reset Value
31 : 0	RW	discard_frame_csi0_m		0x0

Name=VI_CSI0_DIS_FRAME_N,offset=0x80

bits	Access Mode	Name	Description	Reset Value
31 : 0	RW	discard_frame_csi0_n		0x0

Name=VI_CSI1_DIS_FRAME_M,offset=0x84

bits	Access Mode	Name	Description	Reset Value
31 : 0	RW	discard_frame_csi1_m		0x0

Name=VI_CSI1_DIS_FRAME_N,offset=0x88

bits	Access Mode	Name	Description	Reset Value
31 : 0	RW	discard_frame_csi1_n		0x0

Name=VI_CSI2_DIS_FRAME_M,offset=0x8c

bits	Access Mode	Name	Description	Reset Value
31 : 0	RW	discard_frame_csi2_m		0x0

Name=VI_CSI2_DIS_FRAME_N,offset=0x90

bits	Access Mode	Name	Description	Reset Value
31 : 0	RW	discard_frame_csi2_n		0x0

Name=VI_DIS_FRAME_EN,offset=0x94

bits	Access Mode	Name	Description	Reset Value
2	RW	discard_frame_csi2_en		0x0
1	RW	discard_frame_csi1_en		0x0
0	RW	discard_frame_csi0_en		0x0

Name=VI_SONY_CONFIG,offset=0x98

bits	Access Mode	Name	Description	Reset Value
18	RW	sony_wdr_334_mode_en	Sony 334-series hdr sensor enable	0x0
17	RW	sony_format_2	reserved	0x0
16	RW	sony_format_1	reserved	0x0
15	RW	sony_format_0	1: RAW10 0:RAW12	0x0
14:11	RW	sony_wdr_mode_2		0x0
10:7	RW	sony_wdr_mode_1		0x0
6:3	RW	sony_wdr_mode_0		0x0
2	RW	sony_wdr_mode_en_2		0x0
1	RW	sony_wdr_mode_en_1		0x0
0	RW	sony_wdr_mode_en_0		0x0

Name=VI_PHY_STA_CLR,offset=0x9C

bits	Access Mode	Name	Description	Reset Value
2	RW	sta_phy2_clr	clr phy2 err sta	0x0
1	RW	sta_phy1_clr	clr phy1 err sta	0x0
0	RW	sta_phy0_clr	clr phy0 err sta	0x0

Name=FLASH_TRIGGER_CFG_1,offset=0xa0

bits	Access Mode	Name	Description	Reset Value
1	RW	flash_tri_sel	flash trigger sel for non-3D sensor; 0: sensor0 1:sensor1	0x0
0	RW	normal_stb_en	flash trigger enable for non-3D sensor	0x0

Name=VI_SONY_CONFIG_1,offset=0xa4

bits	Access Mode	Name	Description	Reset Value
31:16	RW	sony_vsync_last_0		0x1
15:0	RW	sony_hsync_last_0		0x1

Name=VI_SONY_CONFIG_2,offset=0xa8

bits	Access Mode	Name	Description	Reset Value
31:16	RW	sony_hsync_delay_0		0x1
15:0	RW	sony_valid_delay_0		0x1

Name=VI_HDR,offset=0xac

bits	Access Mode	Name	Description	Reset Value
5:4	RW	hdr_mode_2	00: no hdr 01: dol2 10: dol3	0x0
3:2	RW	hdr_mode_1	00: no hdr 01: dol2 10: dol3	0x0
1:0	RW	hdr_mode_0	00: no hdr 01: dol2 10: dol3	0x0

Name=VI_PHY_ENABLECLK,offset=0xb0

bits	Access Mode	Name	Description	Reset Value
2	RW	enableclk_2	enable clk for phy2	0x0
1	RW	enableclk_1	enable clk for phy1	0x0
0	RW	enableclk_0	enable clk for phy0	0x0

8.2 VO

8.2.1 Overview

VO (Video Output) generates the picture that will be displayed, which are from the video sources and the OSD sources. The unit supports up to 13 display layers, including one back-ground layer, four video layers, and eight OSD layers. The output size and position of each layer could be configurable.

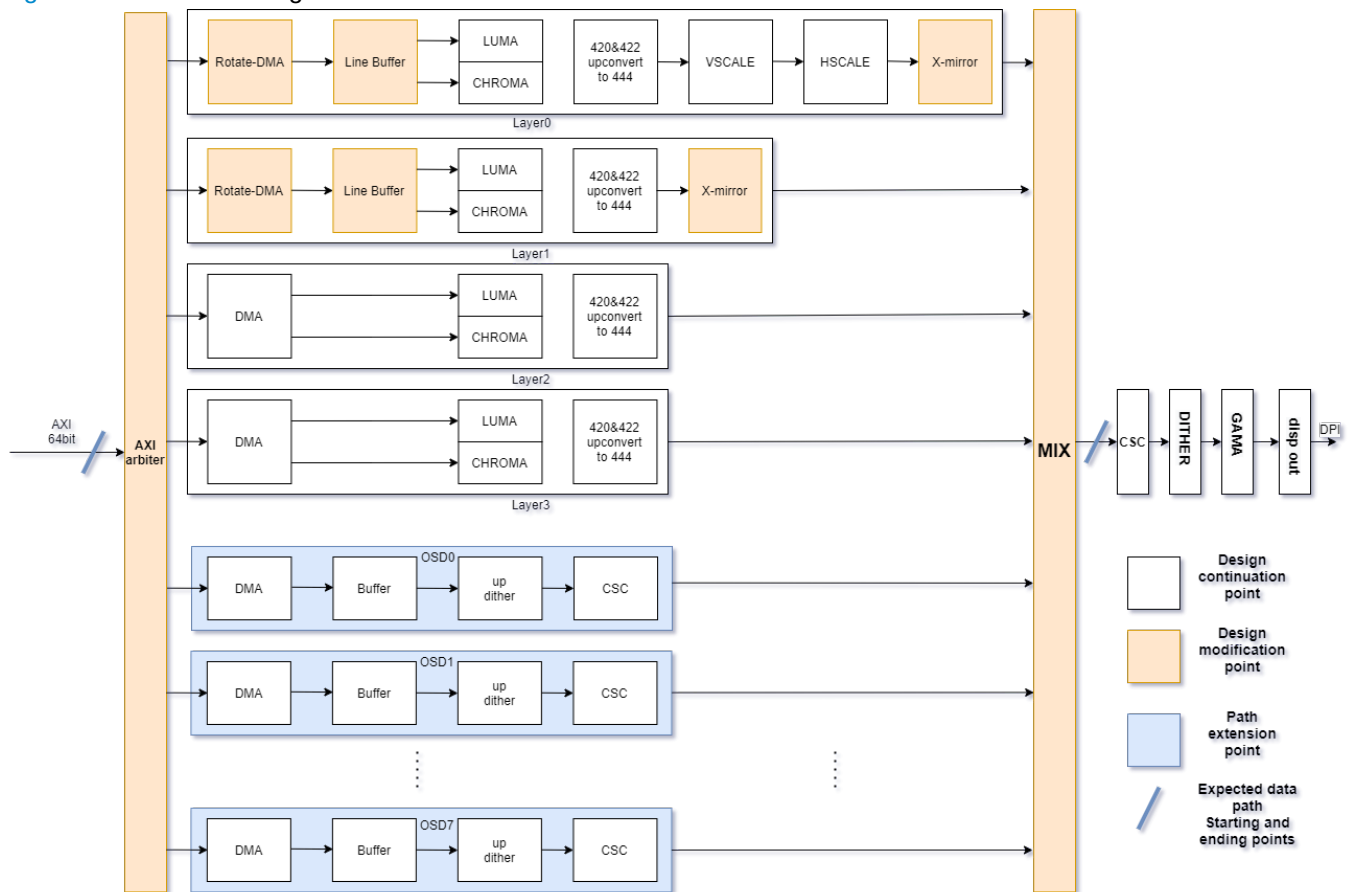
VO reads image data from DDR by AXI interface. After all layers mixed with each other, the final picture is fed out through DPI. Key features:

- Support up to 13 layers
- Support all common output picture formats: UXVGA (1600x1200), SXVGA (1280x1024), 1080p, 720p
- Video Layer 0:
 - support scaling up and scaling down function

- support 2-plane, YUV420 data format input
- support configurable start pixel & start line input
- Support 90 °, 180 °, 270 ° rotation
- Support x-mirror/y-mirror function.
- Video Layer 1:
 - support 2-plane, YUV420 data format input
 - support configurable start pixel & start line input
 - support 90 °, 180 °, 270 ° rotation
 - Support x-mirror/y-mirror function.
- Video Layer 2 & Layer 3:
 - Support 2-plane, YUV420 & YUV422 data format input
 - Support configurable start pixel & start line input
 - Do not support scaling function
- 8 OSD layers support 1 or 2 plane ARGB formats:
 - RGB888 with separate alpha channel
 - RGB565 with separate alpha channel
 - ARGB8888
 - ARGB4444
 - ARGB1555
 - Monochrome(8-bit)
- Display mix:
 - OSD layer support configurable global alpha blending or pixel by pixel alpha mode
 - Video layer only support configurable global alpha
 - Configurable output size and position for each layer
- Support vsync interrupt, DPI interface underrun interrupt and configurable output line number interrupt
- Support configurable color space conversion, yuv2rgb or rgb2yuv
- Support gamma correction on RGB data before output
- Support dither function

8.2.2 Block Diagram

Figure 8-2-1 vo block diagram



Up to 13 layers are supported. Each layer could be configured with output position and size. And layers should be configured within background layer. Background layer corresponds to display region indicated by “DEN” of DPI interface. OSD layers support pixel-by-pixel alpha blending and global-alpha-blending, other layers only support global-alpha-blending.

8.2.3 Function Block

8.2.3.1 Video Layer 0

- Supports all common picture formats (i.e. 1080p, 1080i, 720p, 576i, 480i)
- Free horizontal and vertical up/down scaling
- Only support progressive mode
- Free horizontal and vertical up/down scaling
- 4 taps in vertical direction, 8 taps in horizontal direction
- 8-bit YUV 4:4:4 output
- Support scale ratio from 1/4 to 64
- Configurable horizontal and vertical scaling coefficients
- Programmable picture display size
- Programmable picture display position within display area
- Global alpha blending
- Support 90 °, 180 °, 270 ° rotation
- Support x-mirror/y-mirror function.

Layer0 load image from DDR through 4-line memory DMA interface, after YUV component up-sampling image is scaled up or down in YUV4:4:4 color space. Layer0 first do vertical scale, then do horizontal scale. Vertical scaler does poly-phase filtering by 4-tap vertical pixel, and horizontal scaler does poly-phase filtering by 8-tap horizontal pixel. Max down ratio of 1/4 is suggested for both directions. Coefficients of vertical and horizontal scale are in two programmable SRAMs. According to phase of output pixel, sets of coefficients are indexed and read out for pixel filtering calculation.

Figure 1 vertical scale coefficient table organization Figure 2 horizontal scale coefficient table organization Figure 3 vertical scale coefficient table organization

In horizontal direction, poly-phase filter uses 8 pixels (8 tap) to calculate 1 output pixel, 64 steps are used to indicated the position, where output pixel is located between pixel3 and pixel4. Corresponding to these 64 positions, the weight of pixel is variable. In vertical direction, poly-phase filter is almost same, except that it uses 4 pixels (4 tap) to calculate output pixel. Scale coefficient tables are programmable. Before enabling of layer 0, coefficient tables of vertical and horizontal scale need to be loaded by external CPU through APB configure interface. Horizontal coefficient table range : BASE + 8000 ~ BASE+ 87FF Vertical coefficient table range : BASE + 8C00 ~ BASE+ 8FFF Address maps of horizontal and vertical coefficient tables organize as Figure 2 and Figure 3 shown. Each coefficient is 12-bit, and occupies 4-byte space in system address space.

8.2.3.2 Video Layer 1

- Supports all common picture formats (i.e. 1080p, 1080i, 720p, 576i, 480i)
- Only support progressive mode
- 8-bit YUV 4:4:4 output
- Programmable picture display size
- Programmable picture display position within display area
- Global alpha blending
- Support 90 °, 180 °, 270 ° rotation
- Support x-mirror/y-mirror function.

8.2.3.3 Video Layer 2 & 3

- Supports all common picture formats (i.e. 1080p, 1080i, 720p, 576i, 480i)
- Only support progressive mode
- 8-bit YUV 4:4:4 output
- Programmable picture display size
- Programmable picture display position within display area
- Global alpha blending

8.2.3.4 OSD Layer 0&1&2&3&4&5&6&7

OSD layers will get OSD data from external memory, and send to mix block to merge with other layers. OSD data include the value of alpha channel and RGB channel; you can refer to register description of OSD to know more about OSD data type and configuration.

- Supports all common OSD data formats:
 - RGB888 with separate alpha channel
 - RGB565 with separate alpha channel
 - ARGB8888
 - ARGB4444
 - ARGB1555
 - Monochrome (8 bit)
- Programmable display size
- Programmable display position
- Support OSD frame crop by program OSD start-address (should be 8-byte aligned and must be start address of new pixel), OSD stride (in unit of pixel) and display size (OSD size in unit of pixel).
- 8-bit output for each ARGB to mix with other layer
- Global or pixel-by-pixel alpha blending

8.2.3.5 Display Mixer

The display mixer mixes all layers to a final picture. It allows also a layer reordering which gives a maximum flexibility to the display generation. Each mixing input has also a programmable alpha blending value. The display controller is signaling the window of each plane. Once the plane becomes active, the pixel is mixed with the input value of the mixer and the alpha value that's coming together with the pixel information. Display mixer has its internal layer, 0~6. Internal layer 0 is the first to do blending, and then internal layer 1, and so on. Each internal mixer layer source could be selected from 7 external layer 0~6, such that blending sequence could be configurable. Each mixing input has also a programmable alpha blending value.

8.2.3.6 Display Controller

The display controller is the timing generator for all signals required within the video output unit. It has a programmable horizontal and vertical counter to support all required standards. Its main function generates Hsync, Vsync and DE signals for the DPI interface and also generates horizontal and vertical positions of all layers. Feature:

- Timing generator for relevant features of all video output
- Synchronization signal output
- IRQ generation
- Support interlaced output For interlaced output, the last line length of either field or only the second field could be shortened. This mechanism can be used to adapt the frame rate to the encoder frame rate, when horizontal and vertical size do not match exactly with the input rate. A possible application would be that an LCD panel is connected directly to the chip. It cannot be used for the HDMI IF, because all lines need to have the exact same length, here.

8.2.4 register summary

Offset Adress	Register	Description
0x0	VO_SOFT_RST_CTL	Sub-system soft reset
0x4	VO_REG_LOAD_CTL	Register load control
0x8	VO_DMA_SW_CTL	AXI reset request
0xc	VO_DMA_RD_CTL_OUT	AXI reset status
0x10	VO_DMA_ARB_MODE	AXI master arbiter mode
0x14	VO_DMA_WEIGHT_WR0	Reserved
0x18	VO_DMA_WEIGHT_WR1	Reserved
0x1c	VO_DMA_WEIGHT_WR2	Reserved
0x20	VO_DMA_WEIGHT_WR3	Reserved
0x24	VO_DMA_WEIGHT_RD0	0~3 channel read priority weight
0x28	VO_DMA_WEIGHT_RD1	4~7 channel read priority weight
0x2c	VO_DMA_WEIGHT_RD2	8~11 channel read priority weight
0x30	VO_DMA_WEIGHT_RD3	12~15 channel read priority weight
0x34	VO_DMA_PRIORITY_WR_0	Reserved
0x38	VO_DMA_PRIORITY_WR_1	Reserved
0x3c	VO_DMA_PRIORITY_RD_0	0~7 channel read priority
0x40	VO_DMA_PRIORITY_RD_1	8~15 channel read priority

Offset Adress	Register	Description
0x44	VO_DMA_ID_WR_0	Reserved
0x48	VO_DMA_ID_WR_1	Reserved
0x4c	VO_DMA_ID_RD_0	0~7 channel read ID
0x50	VO_DMA_ID_RD_1	8~15 channel read ID
0x94	VO_LAYER2_BD_CTL	Layer 2 bandwidth control
0x98	VO_LAYER3_BD_CTL	Layer 3 bandwidth control
0x9c	VO_OSD0_BD_CTL	OSD 0 bandwidth control
0xa0	VO_OSD1_BD_CTL	OSD 1 bandwidth control
0xa4	VO_OSD2_BD_CTL	OSD 2 bandwidth control
0x800	VO_OSD3_BD_CTL	OSD 3 bandwidth control
0x804	VO_OSD4_BD_CTL	OSD 4 bandwidth control
0x808	VO_OSD5_BD_CTL	OSD 5 bandwidth control
0x80c	VO_OSD6_BD_CTL	OSD 6 bandwidth control
0x810	VO_OSD7_BD_CTL	OSD 7 bandwidth control
0xa00	VO_LAYER0_CFG	Video layer0 config
0xa04	VO_LAYER0_SIZE	Video layer0 vsize & hsize
0xa08	VO_LAYER0_STRIDE	Video layer0 ystride&uv stride
0xa0c	VO_LAYER0_BADDR_Y0	Video layer0 y base address0
0xa10	VO_LAYER0_BADDR_UV0	Video layer0 uv base address0
0xa14	VO_LAYER0_BADDR_Y1	Video layer0 y base address1
0xa18	VO_LAYER0_BADDR_UV1	Video layer0 uv base address1
0xa1c	VO_LAYER0_ADDR_SEL	Video buffer address select
0xa20	VO_LAYER1_CFG	Video layer1 config
0xa24	VO_LAYER1_SIZE	Video layer1 vsize & hsize
0xa28	VO_LAYER1_STRIDE	Video layer1 ystride&uv stride
0xa2c	VO_LAYER1_BADDR_Y0	Video layer1 y base address0
0xa30	VO_LAYER1_BADDR_UV0	Video layer1 uv base address0
0xa34	VO_LAYER1_BADDR_Y1	Video layer1 y base address1
0xa38	VO_LAYER1_BADDR_UV1	Video layer1 uv base address1
0xa3c	VO_LAYER1_ADDR_SEL	Video buffer address select
0xc0	VO_DISP_XZONE_CTL	Background layer horizontal start & stop
0xc4	VO_DISP_YZONE_CTL	Background layer vertical start & stop
0xc8	VO_DISP_LAYER0_XCTL	Layer0 horizontal start & stop

Offset Adress	Register	Description
0xcc	VO_DISP_LAYER0_YCTL	Layer0 vertical start & stop
0xd0	VO_DISP_LAYER1_XCTL	Layer1 horizontal start & stop
0xd4	VO_DISP_LAYER1_YCTL	Layer1 vertical start & stop
0xd8	VO_DISP_LAYER2_XCTL	Layer2 horizontal start & stop
0xdc	VO_DISP_LAYER2_YCTL	Layer2 vertical start & stop
0xe0	VO_DISP_LAYER3_XCTL	Layer3 horizontal start & stop
0xe4	VO_DISP_LAYER3_YCTL	Layer3 vertical start & stop
0xe8	VO_DISP_OSD0_XCTL	OSD0 horizontal start & stop
0xec	VO_DISP_OSD0_YCTL	OSD0 vertical start & stop
0xf0	VO_DISP_OSD1_XCTL	OSD1 horizontal start & stop
0xf4	VO_DISP_OSD1_YCTL	OSD1 vertical start & stop
0xf8	VO_DISP_OSD2_XCTL	OSD2 horizontal start & stop
0xfc	VO_DISP_OSD2_YCTL	OSD2 vertical start & stop
0x820	VO_DISP_OSD3_XCTL	OSD3 horizontal start & stop
0x824	VO_DISP_OSD3_YCTL	OSD3 vertical start & stop
0x828	VO_DISP_OSD4_XCTL	OSD4 horizontal start & stop
0x82c	VO_DISP_OSD4_YCTL	OSD4 vertical start & stop
0x830	VO_DISP_OSD5_XCTL	OSD5 horizontal start & stop
0x834	VO_DISP_OSD5_YCTL	OSD5 vertical start & stop
0x838	VO_DISP_OSD6_XCTL	OSD6 horizontal start & stop
0x83c	VO_DISP_OSD6_YCTL	OSD6 vertical start & stop
0x840	VO_DISP_OSD7_XCTL	OSD7 horizontal start & stop
0x844	VO_DISP_OSD7_YCTL	OSD7 vertical start & stop
0x10c	VO_DISP_HSYNC2_CTL	Display horizontal position of VSYNC2
0x110	VO_DISP_VSYNC2_CTL	Display vertical position of VSYNC2
0x114	VO_DISP_CTL	Display control
0x118	VO_DISP_ENABLE	Display output enable
0x11c	VO_DISP_SIZE	Display frame horizontal & vertical
0x128	VO_LCD_CTL	Reserved
0x140	VO_LAYER0_CTL	Layer0 enable
0x144	VO_LAYER0_SCALE_BLENTH	Layer0 line memory burst length
0x148	VO_LAYER0_IMG_IN_DAT_MODE	Layer0 input image data format: YUV420, YUV422 Only support YUV420

Offset Adress	Register	Description
0x14c	VO_LAYER0_BADDR0_Y	Don't use
0x150	VO_LAYER0_BADDR0_UV	Don't use
0x154	VO_LAYER0_BADDR1_Y	Don't use
0x158	VO_LAYER0_BADDR1_UV	Don't use
0x15c	VO_LAYER0_IMG_IN_PIX_SIZE	Layer0 in image size in pixel
0x160	VO_LAYER0_IMG_OUT_PIX_SIZE	Layer0 out image size in pixel
0x164	VO_LAYER0_IMG_IN_STRIDE	Layer0 in image size in 8 pixel
0x168	VO_LAYER0_IMG_IN_MEM_HSIZE	Reserved
0x16c	VO_LAYER0_IMG_IN_OFFSET	Don't use
0x170	VO_LAYER0_SCALE_CTL	Reserved
0x174	VO_LAYER0_VSCALE_STEP	Layer0 vscale step
0x178	VO_LAYER0_VSCALE_ST_PSTEP	Layer0 vscale start phase
0x17c	VO_LAYER0_HSCALE_STEP	Layer0 hscale step
0x180	VO_LAYER0_HSCALE_ST_PSTEP	Layer0 hscale start phase
0x184	VO_LAYER0_ADDR_SEL_MODE	Don't use
0x200	VO_LAYER2_CTL	Layer2 enable
0x204	VO_LAYER2_IMG_IN_BADDR0_Y	Layer 2 in image y base address0
0x208	VO_LAYER2_IMG_IN_BADDR0_UV	Layer 2 in image uv base address0
0x20c	VO_LAYER2_IMG_IN_BADDR1_Y	Layer 2 in image uv base address1
0x210	VO_LAYER2_IMG_IN_BADDR0_UV	Layer 2 in image uv base address1
0x214	VO_LAYER2_IMG_IN_OFFSET	Layer 2 in image horizontal offset(in 8 pixel) & vertical offset(in line)
0x218	VO_LAYER2_IMG_IN_BLENTH	Layer 2 burst length of DMA access
0x21c	VO_LAYER2_IMG_IN_STRIDE	Layer 2 in image horizontal size(in 8 pixel) & vertical size(in line)
0x220	VO_LAYER2_IMG_IN_PIX_SIZE	Layer 2 in image horizontal size(in 1 pixel) & vertical size(in line)
0x224	VO_LAYER2_ADDR_SEL_MODE	Layer 2 base address select mode
0x240	VO_LAYER3_CTL	Layer3 enable
0x244	VO_LAYER3_IMG_IN_BADDR0_Y	Layer 3 in image y base address0
0x248	VO_LAYER3_IMG_IN_BADDR0_UV	Layer 3 in image uv base address0
0x24c	VO_LAYER3_IMG_IN_BADDR1_Y	Layer 3 in image uv base address1
0x250	VO_LAYER3_IMG_IN_BADDR0_UV	Layer 3 in image uv base address1

Offset Adress	Register	Description
0x254	VO_LAYER3_IMG_IN_OFFSET	Layer 3 in image horizontal offset(in 8 pixel) & vertical offset(in line)
0x258	VO_LAYER3_IMG_IN_BLENTH	Layer 3 burst length of DMA access
0x25c	VO_LAYER3_IMG_IN_STRIDE	Layer 3 in image horizontal size(in 8 pixel) & vertical size(in line)
0x260	VO_LAYER3_IMG_IN_PIX_SIZE	Layer 3 in image horizontal size(in 1 pixel) & vertical size(in line)
0x264	VO_LAYER3_ADDR_SEL_MODE	Layer 3 base address select mode
0x0280	VO_DISP_OSD0_INFO	OSD 0 image data format
0x0284	VO_DISP_OSD0_SIZE	OSD 0 image size
0x0288	VO_DISP_OSD0_VLU_ADDR0	OSD 0 RGB base address
0x028C	VO_DISP_OSD0_ALP_ADDR0	OSD 0 alpha base address
0x0290	VO_DISP_OSD0_VLU_ADDR1	OSD 0 RGB base address
0x0294	VO_DISP_OSD0_ALP_ADDR1	OSD 0 alpha base address
0x0298	VO_DISP_OSD0_DMA_CTRL	OSD 0 DMA contorl
0x029C	VO_DISP_OSD0_STRIDE	OSD 0 frame address space stride
0x02A0	VO_OSD0_ADDR_SEL_MODE	OSD 0 base address select mode
0x02C0	VO_DISP_OSD1_INFO	OSD 1 image data format
0x02C4	VO_DISP_OSD1_SIZE	OSD 1 image size
0x02C8	VO_DISP_OSD1_VLU_ADDR0	OSD 1 RGB base address
0x02CC	VO_DISP_OSD1_ALP_ADDR0	OSD 1 alpha base address
0x02D0	VO_DISP_OSD1_VLU_ADDR1	OSD 1 RGB base address
0x02D4	VO_DISP_OSD1_ALP_ADDR1	OSD 1 alpha base address
0x02D8	VO_DISP_OSD1_DMA_CTRL	OSD 1 DMA contorl
0x02DC	VO_DISP_OSD1_STRIDE	OSD 1 frame address space stride
0x02E0	VO_OSD1_ADDR_SEL_MODE	OSD 1 base address select mode
0x0300	VO_DISP_OSD2_INFO	OSD 2 image data format
0x0304	VO_DISP_OSD2_SIZE	OSD 2 image size
0x0308	VO_DISP_OSD2_VLU_ADDR0	OSD 2 RGB base address
0x030C	VO_DISP_OSD2_ALP_ADDR0	OSD 2 alpha base address
0x0310	VO_DISP_OSD2_VLU_ADDR1	OSD 2 RGB base address
0x0314	VO_DISP_OSD2_ALP_ADDR1	OSD 2 alpha base address
0x0318	VO_DISP_OSD2_DMA_CTRL	OSD 2 DMA contorl
0x031C	VO_DISP_OSD2_STRIDE	OSD 2 frame address space stride

Offset Address	Register	Description
0x0320	VO_OSD2_ADDR_SEL_MODE	OSD 2 base address select mode
0x0850	VO_DISP_OSD3_INFO	OSD 3 image data format
0x0854	VO_DISP_OSD3_SIZE	OSD 3 image size
0x0858	VO_DISP_OSD3_VLU_ADDR0	OSD 3 RGB base address
0x085C	VO_DISP_OSD3_ALP_ADDR0	OSD 3 alpha base address
0x0860	VO_DISP_OSD3_VLU_ADDR1	OSD 3 RGB base address
0x0864	VO_DISP_OSD3_ALP_ADDR1	OSD 3 alpha base address
0x0868	VO_DISP_OSD3_DMA_CTRL	OSD 3 DMA control
0x086C	VO_DISP_OSD3_STRIDE	OSD 3 frame address space stride
0x0870	VO_OSD3_ADDR_SEL_MODE	OSD 3 base address select mode
0x0880	VO_DISP_OSD4_INFO	OSD 4 image data format
0x0884	VO_DISP_OSD4_SIZE	OSD 4 image size
0x0888	VO_DISP_OSD4_VLU_ADDR0	OSD 4 RGB base address
0x088C	VO_DISP_OSD4_ALP_ADDR0	OSD 4 alpha base address
0x0890	VO_DISP_OSD4_VLU_ADDR1	OSD 4 RGB base address
0x0894	VO_DISP_OSD4_ALP_ADDR1	OSD 4 alpha base address
0x0898	VO_DISP_OSD4_DMA_CTRL	OSD 4 DMA control
0x089C	VO_DISP_OSD4_STRIDE	OSD 4 frame address space stride
0x08A0	VO_OSD4_ADDR_SEL_MODE	OSD 4 base address select mode
0x08B0	VO_DISP_OSD5_INFO	OSD 5 image data format
0x08B4	VO_DISP_OSD5_SIZE	OSD 5 image size
0x08B8	VO_DISP_OSD5_VLU_ADDR0	OSD 5 RGB base address
0x08BC	VO_DISP_OSD5_ALP_ADDR0	OSD 5 alpha base address
0x08C0	VO_DISP_OSD5_VLU_ADDR1	OSD 5 RGB base address
0x08C4	VO_DISP_OSD5_ALP_ADDR1	OSD 5 alpha base address
0x08C8	VO_DISP_OSD5_DMA_CTRL	OSD 5 DMA control
0x08CC	VO_DISP_OSD5_STRIDE	OSD 5 frame address space stride
0x08D0	VO_OSD5_ADDR_SEL_MODE	OSD 5 base address select mode
0x08E0	VO_DISP_OSD6_INFO	OSD 6 image data format
0x08E4	VO_DISP_OSD6_SIZE	OSD 6 image size
0x08E8	VO_DISP_OSD6_VLU_ADDR0	OSD 6 RGB base address
0x08EC	VO_DISP_OSD6_ALP_ADDR0	OSD 6 alpha base address
0x08F0	VO_DISP_OSD6_VLU_ADDR1	OSD 6 RGB base address

Offset Adress	Register	Description
0x08F4	VO_DISP_OSD6_ALP_ADDR1	OSD 6 alpha base address
0x08F8	VO_DISP_OSD6_DMA_CTRL	OSD 6 DMA control
0x08FC	VO_DISP_OSD6_STRIDE	OSD 6 frame address space stride
0x0900	VO_OSD6_ADDR_SEL_MODE	OSD 6 base address select mode
0x0910	VO_DISP_OSD7_INFO	OSD 7 image data format
0x0914	VO_DISP_OSD7_SIZE	OSD 7 image size
0x0918	VO_DISP_OSD7_VLU_ADDR0	OSD 7 RGB base address
0x091C	VO_DISP_OSD7_ALP_ADDR0	OSD 7 alpha base address
0x0920	VO_DISP_OSD7_VLU_ADDR1	OSD 7 RGB base address
0x0924	VO_DISP_OSD7_ALP_ADDR1	OSD 7 alpha base address
0x0928	VO_DISP_OSD7_DMA_CTRL	OSD 7 DMA control
0x092C	VO_DISP_OSD7_STRIDE	OSD 7 frame address space stride
0x0930	VO_OSD7_ADDR_SEL_MODE	OSD 7 base address select mode
0x0340	VO_OSD_RGB2YUV_CTL	Osd layer RGB2YUV enable
0x0344	VO_OSD_RGB2YUV_COEFF00	RGB2YUV coefficient
0x0348	VO_OSD_RGB2YUV_COEFF01	RGB2YUV coefficient
0x034C	VO_OSD_RGB2YUV_COEFF02	RGB2YUV coefficient
0x0350	VO_OSD_RGB2YUV_COEFF03	RGB2YUV coefficient
0x0354	VO_OSD_RGB2YUV_COEFF10	RGB2YUV coefficient
0x0358	VO_OSD_RGB2YUV_COEFF11	RGB2YUV coefficient
0x035C	VO_OSD_RGB2YUV_COEFF12	RGB2YUV coefficient
0x0360	VO_OSD_RGB2YUV_COEFF13	RGB2YUV coefficient
0x0364	VO_OSD_RGB2YUV_COEFF20	RGB2YUV coefficient
0x0368	VO_OSD_RGB2YUV_COEFF21	RGB2YUV coefficient
0x036C	VO_OSD_RGB2YUV_COEFF22	RGB2YUV coefficient
0x0370	VO_OSD_RGB2YUV_COEFF23	RGB2YUV coefficient
0x0380	VO_DISP_YUV2RGB_CTL	After alpha blending, YUV2RGB conversion enable
0x0384	VO_DISP_YUV2RGB_COEFF00	YUV2RGB coefficient
0x0388	VO_DISP_YUV2RGB_COEFF01	YUV2RGB coefficient
0x038C	VO_DISP_YUV2RGB_COEFF02	YUV2RGB coefficient
0x0390	VO_DISP_YUV2RGB_COEFF03	YUV2RGB coefficient
0x0394	VO_DISP_YUV2RGB_COEFF10	YUV2RGB coefficient
0x0398	VO_DISP_YUV2RGB_COEFF11	YUV2RGB coefficient

Offset Adress	Register	Description
0x039C	VO_DISP_YUV2RGB_COEFF12	YUV2RGB coefficient
0x03A0	VO_DISP_YUV2RGB_COEFF13	YUV2RGB coefficient
0x03A4	VO_DISP_YUV2RGB_COEFF20	YUV2RGB coefficient
0x03A8	VO_DISP_YUV2RGB_COEFF21	YUV2RGB coefficient
0x03AC	VO_DISP_YUV2RGB_COEFF22	YUV2RGB coefficient
0x03B0	VO_DISP_YUV2RGB_COEFF23	YUV2RGB coefficient
0x03C4	VO_DISP_MIX_LAYER_GLB_ALPHA0	Layer0~3 layer global alpha
0x03C8	VO_DISP_MIX_LAYER_GLB_ALPHA1	Osd0~3 layer global alpha
0x0940	VO_DISP_MIX_LAYER_GLB_ALPHA2	Osd4~7 layer global alpha
0x03CC	VO_DISP_MIX_SEL0	Display mixing layer sequence selection
0x0950	VO_DISP_MIX_SEL1	Display mixing layer sequence selection
0x03D0	VO_DISP_BACKGROUND	Background RGB color value
0x03D4	VO_DISP_DITH_CTL	Display dither enable
0x03D8	VO_DISP_CLUT_CTL	Gamma correction enable
0x03E0	VO_DISP_IRQ0_CTL	Display irq0 control
0x03E4	VO_DISP_IRQ1_CTL	Display irq1 control
0x03E8	VO_DISP_IRQ2_CTL	Display irq2 control
0x03EC	VO_DISP_IRQ	Display irq clear & status
0x0400	VO_WB_DMA_CH_WR_EN	Write Back DMA enable
0x0404	VO_WB_MODE	Write Back mode
0x0408	VO_WB_PIXEL_FORMAT_OUT	Write Back Pixel format
0x040C	VO_WB_FRM_BASE_OUT_ADDR0_Y	Write Back base address(Y0)
0x0410	VO_WB_FRM_BASE_OUT_ADDR1_Y	Write Back base address(Y1)
0x0414	VO_WB_FRM_BASE_OUT_ADDR0_UV	Write Back base address(UV0)
0x0418	VO_WB_FRM_BASE_OUT_ADDR1_UV	Write Back base address(UV1)
0x041C	VO_WB_HSTRIDE_OUT_Y	Write Back line stride (Y)
0x0420	VO_WB_HSTRIDE_OUT_UV	Write Back line stride (UV)
0x0424	VO_WB_FRM_BUF_SIZE_OUT	Write Back frame buffer size, 64KByte unit,64K*N/Stride=Int (N=1,2,4,8)
0x0428	VO_WB_V_SIZE_ACTIVE_OUT	Write Back Active vertical size,(actual -1),
0x042C	VO_WB_H_SIZE_ACTIVE_OUT	Write Back Active horizontal size,(actual -1),
0x0440	VO_WB_CH0_DMA_MODE	Write Back DMA control mode.
0x0444	VO_WB_CH0_ERROR_UNIT	Write Back DMA Error check Unit

Offset Adress	Register	Description
0x0448	VO_WB_CH0_ERROR_TH	Write Back DMA Error check Threshold
0x044C	VO_WB_CH0_INFO_CLR	Write Back DMA Info Clear
0x0460	VO_WB_CH0_RST_YEQ	Write Back SW reset request
0x0464	VO_WB_CH1_ERROR_UNIT	Write Back DMA Error check Unit
0x0468	VO_WB_CH1_ERROR_TH	Write Back DMA Error check Threshold
0x046C	VO_WB_CH1_INFO_CLR	Write Back DMA Info Clear
0x0470	VO_WB_CH1_RST_YEQ	Write Back SW reset request
0x0480	VO_WB_Y_WR_OUT_IDLE	Write Back DMA Channel Y Idle
0x0484	VO_WB_UV_WR_OUT_IDLE	Write Back DMA Channel UV Idle
0x0488	VO_WB_Y_WR_OUT_INT	Write Back DMA Channel Y Int
0x048C	VO_WB_UV_WR_OUT_INT	Write Back DMA Channel UV Int
0x0490	VO_WB_Y_WR_OUT_FUNC_0	Write Back DMA Channel Y Read Information[31:0]
0x0494	VO_WB_Y_WR_OUT_FUNC_1	Write Back DMA Channel Y Read Information[63:32]
0x0498	VO_WB_UV_WR_OUT_FUNC_0	Write Back DMA Channel UV Read Information[31:0]
0x049C	VO_WB_UV_WR_OUT_FUNC_1	Write Back DMA Channel UV Read Information[63:32]

8.2.5 Register Description

VO_SOFT_RST_CTL Offset Address: 0x0 Total Reset Value: 0x0000_0000

bits	Access	Name	Descriptions	Reset
[31:4]	-	reserved	reserved	0x0
[3]	RW	VO_SF_RST_CORE	When '1', reset core clock domain logic	0x0
[2]	RW	VO_SF_RST_APB	When '1', reset apb clock domain logic	0x0
[1]	RW	VO_SF_RST_AXI	When '1', reset axi clock domain logic	0x0
[0]	RW	VO_SF_RST_DISP	When '1', reset display clock domain logic	0x0

VO_REG_LOAD_CTL Offset Address: 0x4 Total Reset Value: 0x0000_0000

bits	Access	Name	Descriptions	Reset
[31:5]	-	reserved	reserved	0x0
[4]	RW	VO_REG_FIRST_LOAD	When '1', means VO registers config done	0x0
[3:1]	-	reserved	reserved	0x0
[0]	RW	VO_REG_READY	When '1', means VO registers config done	0x0

VO_DMA_SW_CTL Offset Address: 0x8 Total Reset Value: 0x0000_0011

bits	Access	Name	Descriptions	Reset

bits	Access	Name	Descriptions	Reset
[31:5]	-	reserved	reserved	0x0
[4]	RW	VO_AXI_RST_REQ_RD	Setting to 0 means request reset for AXI read port.	0x1
[3:1]	-	reserved	reserved	0x0
[0]	RW	VO_AXI_RST_REQ_WR	Setting to 0 means request reset for AXI write port	0x1

VO_DMA_RD_CTL_OUT Offset Address: 0xc Total Reset Value: 0x0000_0000

bits	Access	Name	Descriptions	Reset
[31:8]	-	reserved	reserved	0x0
[7]	RO	VO_AXI_RD_RST_RDY	AXI read port reset is ready.	0x0
[6:4]	RO	VO_AXI_RD_ST		0x0
[3]	RO	VO_AXI_WR_RST_RDY	AXI write port reset is ready.	0x0
[2:0]	RO	VO_AXI_WR_ST		0x0

VO_DMA_ARB_MODE Offset Address: 0x10 Total Reset Value: 0x0000_0000

bits	Access	Name	Descriptions	Reset
[31:2]	-	reserved	reserved	0x0
[1:0]	RW	VO_DMA_RAB_MODE	0: read channel arbiter mode. When '0', use priority configured by CPU When '1', use weight + round-robin	

1 : write channel arbiter mode.

When '0', use priority configured by CPU When '1', use weight + round-robin | 0x0 **VO_DMA_WEIGHT_RD0** Offset Address: 0x24 Total Reset Value: 0xffff_ffff

bits	Access	Name	Descriptions	Reset
[31:24]	RW	CH3_READ_WEIGHT	Chanel 3 read weight	0xff
[23:16]	RW	CH2_READ_WEIGHT	Chanel 2 read weight	0xff
[15:8]	RW	CH1_READ_WEIGHT	Chanel 1 read weight	0xff
[7:0]	RW	CH0_READ_WEIGHT	Chanel 0 read weight	0xff

VO_DMA_WEIGHT_RD1 Offset Address: 0x28 Total Reset Value: 0xffff_ffff

bits	Access	Name	Descriptions	Reset
[31:24]	RW	CH7_READ_WEIGHT	Chanel 7 read weight	0xff
[23:16]	RW	CH6_READ_WEIGHT	Chanel 6 read weight	0xff
[15:8]	RW	CH5_READ_WEIGHT	Chanel 5 read weight	0xff
[7:0]	RW	CH4_READ_WEIGHT	Chanel 4 read weight	0xff

VO_DMA_WEIGHT_RD2 Offset Address: 0x2c Total Reset Value: 0xffff_ffff

bits	Access	Name	Descriptions	Reset
[31:24]	RW	CH11_READ_WEIGHT	Chanel 11 read weight	0xff
[23:16]	RW	CH10_READ_WEIGHT	Chanel 10 read weight	0xff
[15:8]	RW	CH9_READ_WEIGHT	Chanel 9 read weight	0xff
[7:0]	RW	CH8_READ_WEIGHT	Chanel 8 read weight	0xff

VO_DMA_WEIGHT_RD3 Offset Address: 0x30 Total Reset Value: 0xffff_fff

bits	Access	Name	Descriptions	Reset
[31:24]	RW	CH15_READ_WEIGHT	Chanel 15 read weight	0xff
[23:16]	RW	CH14_READ_WEIGHT	Chanel 14 read weight	0xff
[15:8]	RW	CH13_READ_WEIGHT	Chanel 13 read weight	0xff
[7:0]	RW	CH12_READ_WEIGHT	Chanel 12 read weight	0xff

VO_DMA_PRIORITY_RD_0 Offset Address: 0x3c Total Reset Value: 0x7654_3210

bits	Access	Name	Descriptions	Reset
[31:28]	RW	CH7_READ_PR	Chanel 7 read priority	0x7
[27:24]	RW	CH6_READ_PR	Chanel 6 read priority	0x6
[23:20]	RW	CH5_READ_PR	Chanel 5 read priority	0x5
[19:16]	RW	CH4_READ_PR	Chanel 4 read priority	0x4
[15:12]	RW	CH3_READ_PR	Chanel 3 read priority	0x3
[11:8]	RW	CH2_READ_PR	Chanel 2 read priority	0x2
[7:4]	RW	CH1_READ_PR	Chanel 1 read priority	0x1
[3:0]	RW	CH0_READ_PR	Chanel 0 read priority	0x0

VO_DMA_PRIORITY_RD_1 Offset Address: 0x40 Total Reset Value: 0xfedc_ba98

bits	Access	Name	Descriptions	Reset
[31:28]	RW	CH15_READ_PR	Chanel 15 read priority	0xf
[27:24]	RW	CH14_READ_PR	Chanel 14 read priority	0xe
[23:20]	RW	CH13_READ_PR	Chanel 13 read priority	0xd
[19:16]	RW	CH12_READ_PR	Chanel 12 read priority	0xc
[15:12]	RW	CH11_READ_PR	Chanel 11 read priority	0xb
[11:8]	RW	CH10_READ_PR	Chanel 10 read priority	0xa
[7:4]	RW	CH9_READ_PR	Chanel 8 read priority	0x9
[3:0]	RW	CH8_READ_PR	Chanel 8 read priority	0x8

VO_DMA_ID_RD_0 Offset Address: 0x4c Total Reset Value: 0x7654_3210

bits	Access	Name	Descriptions	Reset
[31:28]	RW	CH7_READ_ID	Chanel 7 read priority	0x7
[27:24]	RW	CH6_READ_ID	Chanel 6 read priority	0x6
[23:20]	RW	CH5_READ_ID	Chanel 5 read priority	0x5
[19:16]	RW	CH4_READ_ID	Chanel 4 read priority	0x4
[15:12]	RW	CH3_READ_ID	Chanel 3 read priority	0x3
[11:8]	RW	CH2_READ_ID	Chanel 2 read priority	0x2
[7:4]	RW	CH1_READ_ID	Chanel 1 read priority	0x1
[3:0]	RW	CH0_READ_ID	Chanel 0 read priority	0x0

VO_DMA_ID_RD_1 Offset Address: 0x50 Total Reset Value: 0xfedc_ba98

bits	Access	Name	Descriptions	Reset
[31:28]	RW	CH15_READ_ID	Chanel 15 read priority	0xf
[27:24]	RW	CH14_READ_ID	Chanel 14 read priority	0xe
[23:20]	RW	CH13_READ_ID	Chanel 13 read priority	0xd
[19:16]	RW	CH12_READ_ID	Chanel 12 read priority	0xc
[15:12]	RW	CH11_READ_ID	Chanel 11 read priority	0xb
[11:8]	RW	CH10_READ_ID	Chanel 10 read priority	0xa
[7:4]	RW	CH9_READ_ID	Chanel 9 read priority	0x9
[3:0]	RW	CH8_READ_ID	Chanel 8 read priority	0x8

VO_LAYER2_BD_CTL Offset Address: 0x94 Total Reset Value: 0x701

bits	Access	Name	Descriptions	Reset
[31:13]	-	-	-	Reserved
[12:8]	RW	VO_LAYER2_OUTSTAND	Layer 2 DMA access outstanding numbe(actual outstanding value– 1)	0x7
[7:1]	-	-	-	Reserved
[0]	RW	VO_LAYER2_LIMIT_EN	When '1', layer 2 DMA access outstanding number limit enable	0x1

VO_LAYER3_BD_CTL Offset Address: 0x98 Total Reset Value: 0x701

bits	Access	Name	Descriptions	Reset
[31:13]	-	-	-	Reserved
[12:8]	RW	VO_LAYER3_OUTSTAND	Layer 3 DMA access outstanding numbe(actual outstanding value– 1)	0x7
[7:1]	-	-	-	Reserved

bits	Access	Name	Descriptions	Reset
[0]	RW	VO_LAYER3_LIMIT_EN	When '1', layer 3 DMA access outstanding number limit enable	0x1

VO_OSD0_BD_CTL Offset Address: 0x9c Total Reset Value: 0x701

bits	Access	Name	Descriptions	Reset
[31:13]	-	-	-	Reserved
[12:8]	RW	VO_OSD0_OUTSTAND	OSD0 DMA access outstanding numbe(actual outstanding value– 1)	0x7
[7:1]	-	-	-	Reserved
[0]	RW	VO_OSD0_LIMIT_EN	When '1', OSD0 access outstanding number limit enable	0x1

VO_OSD1_BD_CTL Offset Address: 0xa0 Total Reset Value: 0x701

bits	Access	Name	Descriptions	Reset
[31:13]	-	-	-	Reserved
[12:8]	RW	VO_OSD1_OUTSTAND	OSD0 DMA access outstanding numbe(actual outstanding value– 1)	0x7
[7:1]	-	-	-	Reserved
[0]	RW	VO_OSD1_LIMIT_EN	When '1', OSD0 access outstanding number limit enable	0x1

VO_OSD2_BD_CTL Offset Address: 0xa4 Total Reset Value: 0x701

bits	Access	Name	Descriptions	Reset
[31:13]	-	-	-	Reserved
[12:8]	RW	VO_OSD2_OUTSTAND	OSD0 DMA access outstanding numbe(actual outstanding value– 1)	0x7
[7:1]	-	-	-	Reserved
[0]	RW	VO_OSD2_LIMIT_EN	When '1', OSD0 access outstanding number limit enable	0x1

VO_OSD3_BD_CTL Offset Address: 0x800 Total Reset Value: 0x701

bits	Access	Name	Descriptions	Reset
[31:13]	-	-	-	Reserved
[12:8]	RW	VO_OSD3_OUTSTAND	OSD0 DMA access outstanding numbe(actual outstanding value– 1)	0x7
[7:1]	-	-	-	Reserved
[0]	RW	VO_OSD3_LIMIT_EN	When '1', OSD0 access outstanding number limit enable	0x1

VO_OSD4_BD_CTL Offset Address: 0x804 Total Reset Value: 0x701

bits	Access	Name	Descriptions	Reset
[31:13]	-	-	-	Reserved
[12:8]	RW	VO_OSD4_OUTSTAND	OSD0 DMA access outstanding numbe(actual outstanding value– 1)	0x7
[7:1]	-	-	-	Reserved
[0]	RW	VO_OSD4_LIMIT_EN	When '1', OSD0 access outstanding number limit enable	0x1

VO_OSD5_BD_CTL Offset Address: 0x808 Total Reset Value: 0x701

bits	Access	Name	Descriptions	Reset
[31:13]	-	-	-	Reserved
[12:8]	RW	VO_OSD5_OUTSTAND	OSD0 DMA access outstanding numbe(actual outstanding value– 1)	0x7
[7:1]	-	-	-	Reserved
[0]	RW	VO_OSD5_LIMIT_EN	When '1', OSD0 access outstanding number limit enable	0x1

VO_OSD6_BD_CTL Offset Address: 0x80c Total Reset Value: 0x701

bits	Access	Name	Descriptions	Reset
[31:13]	-	-	-	Reserved
[12:8]	RW	VO_OSD6_OUTSTAND	OSD0 DMA access outstanding numbe(actual outstanding value– 1)	0x7
[7:1]	-	-	-	Reserved
[0]	RW	VO_OSD6_LIMIT_EN	When '1', OSD0 access outstanding number limit enable	0x1

VO_OSD7_BD_CTL Offset Address: 0x810 Total Reset Value: 0x701

bits	Access	Name	Descriptions	Reset
[31:13]	-	-	-	Reserved
[12:8]	RW	VO_OSD7_OUTSTAND	OSD0 DMA access outstanding numbe(actual outstanding value– 1)	0x7
[7:1]	-	-	-	Reserved
[0]	RW	VO_OSD7_LIMIT_EN	When '1', OSD0 access outstanding number limit enable	0x1

VO_DISP_XZONE_CTL Offset Address: 0xc0 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[15:13]	-	-	-	Reserved

bits	Access	Name	Descriptions	Reset
[28:16]	RW	VO_DISP_ZONE_XSTOP	Display active zone horizontal stop pixel address (min pixeladdress is 1)	0x0
[15:13]	-	-	-	Reserved
[12:0]	RW	VO_DISP_ZONE_XSTART	Display active zone horizontal start pixel address (min pixel address is 1)	0x0

VO_DISP_YZONE_CTL Offset Address: 0xc4 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[15:13]	-	-	-	Reserved
[28:16]	RW	VO_DISP_ZONE_YSTOP	Display active zone vertical stop pixel address (min pixeladdress is 1)	0x0
[15:13]	-	-	-	Reserved
[12:0]	RW	VO_DISP_ZONE_YSTART	Display active zone vertical start pixel address (min pixel address is 1)	0x0

VO_DISP_LAYER0_XCTL Offset Address: 0xc8 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[15:13]	-	-	-	Reserved
[28:16]	RW	VO_DISP_LAYER0_XSTOP	Display layer 0 horizontal start pixel address (min pixel address is 1)	0x0
[15:13]	-	-	-	Reserved
[12:0]	RW	VO_DISP_LAYER0_XSTART	Display layer 0 horizontal start pixel address (min pixel address is 1)	0x0

VO_DISP_LAYER0_YCTL Offset Address: 0xcc Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[15:13]	-	-	-	Reserved
[28:16]	RW	VO_DISP_LAYER0_YSTOP	Display layer 0 vertical start pixel address (min pixel address is 1)	0x0
[15:13]	-	-	-	Reserved
[12:0]	RW	VO_DISP_LAYER0_YSTART	Display layer 0 vertical start pixel address (min pixel address is 1)	0x0

VO_DISP_LAYER1_XCTL Offset Address: 0xd0 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[15:13]	-	-	-	Reserved
[28:16]	RW	VO_DISP_LAYER1_XSTOP	Display layer 1 horizontal start pixel address (min pixel address is 1)	0x0

bits	Access	Name	Descriptions	Reset
[15:13]	-	-	-	Reserved
[12:0]	RW	VO_DISP_LAYER1_XSTART	Display layer 1 horizontal start pixel address (min pixel address is 1)	0x0

VO_DISP_LAYER1_YCTL Offset Address: 0xd4 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[15:13]	-	-	-	Reserved
[28:16]	RW	VO_DISP_LAYER1_YSTOP	Display layer 1 vertical start pixel address (min pixel address is 1)	0x0
[15:13]	-	-	-	Reserved
[12:0]	RW	VO_DISP_LAYER1_YSTART	Display layer 1 vertical start pixel address (min pixel address is 1)	0x0

VO_DISP_LAYER2_XCTL Offset Address: 0xd8 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[15:13]	-	-	-	Reserved
[28:16]	RW	VO_DISP_LAYER2_XSTOP	Display layer 2 horizontal start pixel address (min pixel address is 1)	0x0
[15:13]	-	-	-	Reserved
[12:0]	RW	VO_DISP_LAYER2_XSTART	Display layer 2 horizontal start pixel address (min pixel address is 1)	0x0

VO_DISP_LAYER2_YCTL Offset Address: 0xdc Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[15:13]	-	-	-	Reserved
[28:16]	RW	VO_DISP_LAYER2_YSTOP	Display layer 2 vertical start pixel address (min pixel address is 1)	0x0
[15:13]	-	-	-	Reserved
[12:0]	RW	VO_DISP_LAYER2_YSTART	Display layer 2 vertical start pixel address (min pixel address is 1)	0x0

VO_DISP_LAYER3_XCTL Offset Address: 0xe0 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[15:13]	-	-	-	Reserved
[28:16]	RW	VO_DISP_LAYER3_XSTOP	Display layer 3 horizontal start pixel address (min pixel address is 1)	0x0
[15:13]	-	-	-	Reserved

bits	Access	Name	Descriptions	Reset
[12:0]	RW	VO_DISP_LAYER3_XSTART	Display layer 3 horizontal start pixel address (min pixel address is 1)	0x0

VO_DISP_LAYER3_YCTL Offset Address: 0xe4 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[15:13]	-	-	-	Reserved
[28:16]	RW	VO_DISP_LAYER3_YSTOP	Display layer 3 vertical start pixel address (min pixel address is 1)	0x0
[15:13]	-	-	-	Reserved
[12:0]	RW	VO_DISP_LAYER3_YSTART	Display layer 3 vertical start pixel address (min pixel address is 1)	0x0

VO_DISP_OSD0_XCTL Offset Address: 0xe8 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[15:13]	-	-	-	Reserved
[28:16]	RW	VO_DISP_OSD0_XSTOP	OSD0 (OSD layer) horizontal start pixel address (min pixel address is 1)	0x0
[15:13]	-	-	-	Reserved
[12:0]	RW	VO_DISP_OSD0_XSTOP	OSD0 (OSD layer) horizontal stop pixel address (min pixel address is 1)	0x0

VO_DISP_OSD0_YCTL Offset Address: 0xec Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[15:13]	-	-	-	Reserved
[28:16]	RW	VO_DISP_OSD0_YSTOP	OSD0 (OSD layer) vertical start pixel address (min pixel address is 1)	0x0
[15:13]	-	-	-	Reserved
[12:0]	RW	VO_DISP_OSD0_YSTOP	OSD0 (OSD layer) vertical stop pixel address (min pixel address is 1)	0x0

VO_DISP_OSD1_XCTL Offset Address: 0xf0 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[15:13]	-	-	-	Reserved
[28:16]	RW	VO_DISP_OSD1_XSTOP	OSD1 (OSD layer) horizontal start pixel address (min pixel address is 1)	0x0
[15:13]	-	-	-	Reserved
[12:0]	RW	VO_DISP_OSD1_XSTOP	OSD1 (OSD layer) horizontal stop pixel address (min pixel address is 1)	0x0

VO_DISP_OSD1_YCTL Offset Address: 0xf4 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[15:13]	-	-	-	Reserved
[28:16]	RW	VO_DISP_OSD1_YSTOP	OSD1 (OSD layer) vertical start pixel address (min pixel address is 1)	0x0
[15:13]	-	-	-	Reserved
[12:0]	RW	VO_DISP_OSD1_YSTOP	OSD1 (OSD layer) vertical stop pixel address (min pixel address is 1)	0x0

VO_DISP_OSD2_XCTL Offset Address: 0xf8 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[15:13]	-	-	-	Reserved
[28:16]	RW	VO_DISP_OSD2_XSTOP	OSD2 (OSD layer) horizontal start pixel address (min pixel address is 1)	0x0
[15:13]	-	-	-	Reserved
[12:0]	RW	VO_DISP_OSD2_XSTOP	OSD2 (OSD layer) horizontal stop pixel address (min pixel address is 1)	0x0

VO_DISP_OSD2_YCTL Offset Address: 0xfc Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[15:13]	-	-	-	Reserved
[28:16]	RW	VO_DISP_OSD2_YSTOP	OSD2 (OSD layer) vertical start pixel address (min pixel address is 1)	0x0
[15:13]	-	-	-	Reserved
[12:0]	RW	VO_DISP_OSD2_YSTOP	OSD2 (OSD layer) vertical stop pixel address (min pixel address is 1)	0x0

VO_DISP_OSD3_XCTL Offset Address: 0x820 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[15:13]	-	-	-	Reserved
[28:16]	RW	VO_DISP_OSD3_XSTOP	OSD3 (OSD layer) horizontal start pixel address (min pixel address is 1)	0x0
[15:13]	-	-	-	Reserved
[12:0]	RW	VO_DISP_OSD3_XSTOP	OSD3 (OSD layer) horizontal stop pixel address (min pixel address is 1)	0x0

VO_DISP_OSD3_YCTL Offset Address: 0x824 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[15:13]	-	-	-	Reserved

bits	Access	Name	Descriptions	Reset
[28:16]	RW	VO_DISP_OSD3_YSTOP	OSD3 (OSD layer) vertical start pixel address (min pixel address is 1)	0x0
[15:13]	-	-	-	Reserved
[12:0]	RW	VO_DISP_OSD3_YSTOP	OSD3 (OSD layer) vertical stop pixel address (min pixel address is 1)	0x0

VO_DISP_OSD4_XCTL Offset Address: 0x828 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[15:13]	-	-	-	Reserved
[28:16]	RW	VO_DISP_OSD4_XSTOP	OSD4 (OSD layer) horizontal start pixel address (min pixel address is 1)	0x0
[15:13]	-	-	-	Reserved
[12:0]	RW	VO_DISP_OSD4_XSTOP	OSD4 (OSD layer) horizontal stop pixel address (min pixel address is 1)	0x0

VO_DISP_OSD4_YCTL Offset Address: 0x82c Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[15:13]	-	-	-	Reserved
[28:16]	RW	VO_DISP_OSD4_YSTOP	OSD4 (OSD layer) vertical start pixel address (min pixel address is 1)	0x0
[15:13]	-	-	-	Reserved
[12:0]	RW	VO_DISP_OSD4_YSTOP	OSD4 (OSD layer) vertical stop pixel address (min pixel address is 1)	0x0

VO_DISP_OSD5_XCTL Offset Address: 0x830 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[15:13]	-	-	-	Reserved
[28:16]	RW	VO_DISP_OSD5_XSTOP	OSD5 (OSD layer) horizontal start pixel address (min pixel address is 1)	0x0
[15:13]	-	-	-	Reserved
[12:0]	RW	VO_DISP_OSD5_XSTOP	OSD5 (OSD layer) horizontal stop pixel address (min pixel address is 1)	0x0

VO_DISP_OSD5_YCTL Offset Address: 0x834 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[15:13]	-	-	-	Reserved
[28:16]	RW	VO_DISP_OSD5_YSTOP	OSD5 (OSD layer) vertical start pixel address (min pixel address is 1)	0x0

bits	Access	Name	Descriptions	Reset
[15:13]	-	-	-	Reserved
[12:0]	RW	VO_DISP_OSD5_YSTOP	OSD5 (OSD layer) vertical stop pixel address (min pixel address is 1)	0x0

VO_DISP_OSD6_XCTL Offset Address: 0x838 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[15:13]	-	-	-	Reserved
[28:16]	RW	VO_DISP_OSD6_XSTOP	OSD6 (OSD layer) horizontal start pixel address (min pixel address is 1)	0x0
[15:13]	-	-	-	Reserved
[12:0]	RW	VO_DISP_OSD6_XSTOP	OSD6 (OSD layer) horizontal stop pixel address (min pixel address is 1)	0x0

VO_DISP_OSD6_YCTL Offset Address: 0x83c Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[15:13]	-	-	-	Reserved
[28:16]	RW	VO_DISP_OSD6_YSTOP	OSD6 (OSD layer) vertical start pixel address (min pixel address is 1)	0x0
[15:13]	-	-	-	Reserved
[12:0]	RW	VO_DISP_OSD6_YSTOP	OSD6 (OSD layer) vertical stop pixel address (min pixel address is 1)	0x0

VO_DISP_OSD7_XCTL Offset Address: 0x840 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[15:13]	-	-	-	Reserved
[28:16]	RW	VO_DISP_OSD7_XSTOP	OSD7 (OSD layer) horizontal start pixel address (min pixel address is 1)	0x0
[15:13]	-	-	-	Reserved
[12:0]	RW	VO_DISP_OSD7_XSTOP	OSD7 (OSD layer) horizontal stop pixel address (min pixel address is 1)	0x0

VO_DISP_OSD7_YCTL Offset Address: 0x844 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[15:13]	-	-	-	Reserved
[28:16]	RW	VO_DISP_OSD7_YSTOP	OSD7 (OSD layer) vertical start pixel address (min pixel address is 1)	0x0
[15:13]	-	-	-	Reserved

bits	Access	Name	Descriptions	Reset
[12:0]	RW	VO_DISP_OSD7_YSTOP	OSD7 (OSD layer) vertical stop pixel address (min pixel address is 1)	0x0

VO_DISP_HSYNC_CTL Offset Address: 0x100 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[15:13]	-	-	-	Reserved
[28:16]	RW	VO_DISP_HSYNC_STOP	Horizontal stop pixel position of Hsync (min pixel address is 1)	0x0
[15:13]	-	-	-	Reserved
[12:0]	RW	VO_DISP_HSYNC_START	Horizontal start pixel position of Hsync (min pixel address is 1)	0x0

VO_DISP_HSYNC1_CTL Offset Address: 0x104 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[15:13]	-	-	-	Reserved
[28:16]	RW	VO_DISP_HSYNC1_STOP	Horizontal stop pixel position of Hsync1 (min pixel address is 1)	0x0
[15:13]	-	-	-	Reserved
[12:0]	RW	VO_DISP_HSYNC1_START	Horizontal start pixel position of Hsync1 (min pixel address is 1)	0x0

VO_DISP_VSYNC1_CTL Offset Address: 0x108 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[15:13]	-	-	-	Reserved
[28:16]	RW	VO_DISP_VSYNC1_STOP	Vertical stop pixel position of Vsync1 (min line address is VO_DISP_VCNT_BIT0)	0x0
[15:13]	-	-	-	Reserved
[12:0]	RW	VO_DISP_VSYNC1_START	Vertical start pixel position of Vsync1 (min line address is VO_DISP_VCNT_BIT0)	0x0

VO_DISP_HSYNC2_CTL Offset Address: 0x10c Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[15:13]	-	-	-	Reserved
[28:16]	RW	VO_DISP_HSYNC2_STOP	Horizontal stop pixel position of Hsync2 (min pixel address is 1)	0x0
[15:13]	-	-	-	Reserved
[12:0]	RW	VO_DISP_HSYNC2_START	Horizontal start pixel position of Hsync2 (min pixel address is 1)	0x0

VO_DISP_VSYNC2_CTL Offset Address: 0x110 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[15:13]	-	-	-	Reserved
[28:16]	RW	VO_DISP_VSYNC2_STOP	Vertical start line position of Vsync2(min line address is VO_DISP_VCNT_BIT0)	0x0
[15:13]	-	-	-	Reserved
[12:0]	RW	VO_DISP_VSYNC2_START	Vertical start line position of Vsync2(min line address is VO_DISP_VCNT_BIT0)	0x0

VO_DISP_CTL Offset Address: 0x114 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:24]	-	-	-	Reserved
[23:16]	RW	VO_FRAME_SYNC_DLY	Offset line number of frame reset for sync between two display control, when two display output are supported.	0x2
[15:10]	-	-	-	Reserved
[9:8]	RW	VO_DISP_2NDF_OFFSET	Offset line number of second field, when display in interlaced output mode.	0x2
[7:1]	-	-	-	Reserved
[0]	RW	VO_DISP_VCNT_BIT0	When '1', Display line cnt start line number is 1.	
When '0', Display line cnt start line number is 0.	0x0			

VO_DISP_ENABLE Offset Address: 0x118 Total Reset Value: 0x10001

bits	Access	Name	Descriptions	Reset
[31:17]	-	-	-	Reserved
[16]	RW	VO_DISP_DISABLE	When '1', display output is disable.	0x0
[15:12]	-	-	-	Reserved
[11]	RW	VO_DISP_OSD7_EN	When '1', osd7 display enable.	0x0
[10]	RW	VO_DISP_OSD6_EN	When '1', osd6 display enable.	0x0
[9]	RW	VO_DISP_OSD5_EN	When '1', osd5 display enable.	0x0
[8]	RW	VO_DISP_OSD4_EN	When '1', osd4 display enable.	0x0
[7]	RW	VO_DISP_OSD3_EN	When '1', osd3 display enable.	0x0
[6]	RW	VO_DISP_OSD2_EN	When '1', osd2 display enable.	0x0

bits	Access	Name	Descriptions	Reset
[5]	RW	VO_DISP_OSD1_EN	When '1', osd1 display enable.	0x0
[4]	RW	VO_DISP_OSD0_EN	When '1', osd0 display enable.	0x0
[3]	RW	VO_DISP_LAYER3_EN	When '1', layer 3 display enable.	0x0
[2]	RW	VO_DISP_LAYER2_EN	When '1', layer 2 display enable.	0x0
[1]	RW	VO_DISP_LAYER1_EN	When '1', layer 1 display enable.	0x0
[0]	RW	VO_DISP_LAYER0_EN	When '1', layer 0 display enable.	0x1

VO_DISP_SIZE Offset Address: 0x11c Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:29]	-	-	-	Reserved
[28:16]	RW	VO_DISP_VSIZE	Horizontal size of display frame, total horizontal count.	0x0
[15:13]	-	-	-	Reserved
[12:0]	RW	VO_DISP_HSIZE	Vertical size of display frame, total vertical count.	0x0

VO_LAYER0_CTL Offset Address: 0x140 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:1]	-	-	-	Reserved
[0]	RW	VO_LAYER0_ENABLE	When '1', layer 0 is enable When '0', layer 0 is disable	0x0

VO_LAYER0_SCALE_BLENTH Offset Address: 0x144 Total Reset Value: 0xf

bits	Access	Name	Descriptions	Reset
[31:4]	-	-	-	Reserved
[3:0]	RW	VO_LAYER0_SCALE_BLENTH	DMA burst length of layer 0 scale line memory 0~4(min length is 0)	0xf

VO_LAYER0_IMG_IN_PIX_SIZE Offset Address: 0x15c Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:29]	-	-	-	Reserved
[28:16]	RW	VO_LAYER0_IMG_IN_PIX_VSIZE	Actual reading Vertical line size of layer 0 source, in line unit. For example, reading image of 800x600, this field is 599.	0x0
[15:13]	-	-	-	Reserved
[12:0]	RW	VO_LAYER0_IMG_IN_PIX_HSIZE	Actual reading horizontal size of layer 0 source, in pixel unit. For example, reading image size of 800x600, this field is 799.	0x0

VO_LAYER0_IMG_OUT_PIX_SIZE Offset Address: 0x160 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:29]	-	-	-	Reserved
[28:16]	RW	Vertical line size of layer 0 scaler output. As example, for image of 800x600, is 599.	0x0	
[15:13]	-	-	-	Reserved
[12:0]	RW	VO_LAYER0_IMG_OUT_PIX_HSIZE	Horizontal size in pixel of layer 0 scaler output. As example, for image of 800x600, is 799.	0x0

VO_LAYER0_IMG_IN_STRIDE Offset Address: 0x164 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:29]	-	-	-	Reserved
[28:16]	RW	VO_LAYER0_IMG_IN_VSIZE	Vertical line size of layer 0 image in DDR.	0x0
[15:13]	-	-	-	Reserved
[12:0]	RW	VO_LAYER0_IMG_IN_HSTRIDE	Horizontal stride of layer 0 image in DDR, in double word(8-pixel).As example, for image of 800x600, this field is 99. (800/8 – 1)	0x0

VO_LAYER0_IMG_IN_OFFSET Offset Address: 0x16c Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:29]	-	-	-	Reserved
[28:16]	RW	VO_LAYER0_IMG_IN_VSIZE	Vertical offset line number relative to start line (0) of layer 0 input image.	0x0
[15:13]	-	-	-	Reserved
[12:0]	RW	VO_LAYER0_IMG_IN_HOFFSET	Horizontal offset in double word(8-pixel) relative to start address(0) of layer 0 input image.	0x0

VO_LAYER0_VSCALE_STEP Offset Address: 0x174 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:29]	-	-	-	Reserved
[28:16]	RW	VO_LAYER0_VSCALE_DSTEP	Decimal part of vertical scale ratio.	

As example, for scale ratio of 2.25, it is $0.25 * 2^{16}$. And, scale ratio = in size /out size.| 0x0 [15:13]| - | - | Reserved [12:0]|RW |VO_LAYER0_VSCALE_PSTEP|Integer part of vertical scale ratio. As example, for scale ratio of 2.25, it is 2. And, scale ratio = in size /out size.| 0x0 **VO_LAYER0_VSCALE_ST_PSTEP** Offset Address: 0x178 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:16]	-	-	-	Reserved

bits	Access	Name	Descriptions	Reset
[15:0]	RW	VO_LAYER0_VSCALE_ST_PSTEP	Start phase of first line position of vertical scale output. It is used to indicate relative position between line1 and line2(4 taps) in input image, when scaler calculates first output line. Min value is '0', Max value is $2^{16}-1$.	0x0

VO_LAYER0_HSCALE_STEP Offset Address: 0x17c Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:16]	RW	VO_LAYER0_HSCALE_DSTEP	Decimal part of horizontal scale ratio.	

As example, for scale ratio of 2.25, it is $0.25 * 2^{16}$. And, scale ratio = in size /out size. | 0x0 [15:8] | - | - | Reserved [7:0] | RW | VO_LAYER0_HSCALE_PSTEP | Integer part of horizontal scale ratio. As example, for scale ratio of 2.25, it is 2. And, scale ratio = in size /out size. | 0x0 **VO_LAYER0_HSCALE_ST_PSTEP** Offset Address: 0x180 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:16]	-	-	-	Reserved
[15:0]	RW	VO_LAYER0_HSCALE_ST_PSTEP	Start phase of first pixel position of horizontal scale output. It is used to indicate relative position between pixel3 and pixel4(8 taps) in input image, when scaler calculates first output pixel. Min value is '0', Max value is $2^{16}-1$.	0x0

VO_LAYER0_ADDR_SEL_MODE Offset Address: 0x184 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:13]	-	-	-	Reserved
[12]	RW	VO_LAYER0_ADDR_FLAG	Indicate which address set is current frame used	0x0
[11:9]	-	-	-	Reserved
[8]	RW	VO_LAYER0_ADDR_BASE1_MASK	0 : hw is NOT allowed to read base1, not start or repeat base0 1 :hw is allowed to read base1	0x0
[7:5]	-	-	-	Reserved
[4]	RW	VO_LAYER0_ADDR_BASE0_MASK	0 : hw is NOT allowed to read base0, not start or repeat base1 1 :hw is allowed to read base0	0x0
[3:1]	-	-	-	Reserved

bits	Access	Name	Descriptions	Reset
[0]	RW	VO_LAYER0_ADDR_SEL_MODE	0: video layer 0 read buffer base0 and base1, Hw switch the pingpong address automatically. 1: video layer 0 read buffer base0 and base1, Hw switch the pingpong address according to register VO_LAYER0_ADDR_BASE0_MASK and VO_LAYER0_ADDR_BASE1_MASK	0x0

VO_LAYER2_CTL Offset Address: 0x200 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:29]	-	-	-	Reserved
[28]	RW	VO_LAYER2_UV_SWAP	Toggle this flag to swap cb cr output channel	0x0
[27:19]	-	-	-	Reserved
[18:16]	RW	VO_LAYER2_ENDIAN_MODE_UV	UV data input endian (64-bit input for example) 0: U2 V2 U3 V3 U0 V0 U1 V1 1: V2 U2 V3 U3 V0 U0 V1 U1 2: U3 V3 U2 V2 U1 V1 U0 V0 3: V3 U3 V2 U2 V1 U1 V0 U0 4: U0 V0 U1 V1 U2 V2 U3 V3 5: V0 U0 V1 U1 V2 U2 V3 U3 6: U1 V1 U0 V0 U3 V3 U2 V2 7: V1 U1 V0 U0 V3 U3 V2 U2	0x3
[15:14]	-	-	-	Reserved
[13:12]	RW	VO_LAYER2_ENDIAN_MODE_Y	Y data input endian (64-bit input for example) 0: Y4 Y5 Y6 Y7 Y0 Y1 Y2 Y3 1: Y7 Y6 Y5 Y4 Y3 Y2 Y1 Y0 2: Y0 Y1 Y2 Y3 Y4 Y5 Y6 Y7 3: Y3 Y2 Y1 Y0 Y7 Y6 Y5 Y4	0x1
[11:9]	-	-	-	Reserved
[8]	RW	VO_LAYER2_YUV420_EN	Set to '1', when layer 2 image is YUV4:2:0 format.	0x0
[7:5]	-	-	-	Reserved
[4]	RW	VO_LAYER2_YUV422_EN	Set to '1', when layer 2 image is YUV4:2:2 format.	0x0
[3:1]	-	-	-	Reserved
[0]	RW	VO_LAYER2_EN	When '1', layer 2 is enabled. When '0', layer 2 is disabled.	0x0

VO_LAYER2_IMG_IN_BADDR0_Y Offset Address: 0x204 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	VO_LAER2_IMG_IN_BADDR_Y	Layer 2 Y frame base address of input image	0x0

VO_LAYER2_IMG_IN_BADDR0_UV

Offset Address: 0x208 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	VO_LAER2_IMG_IN_BADDR_UV	Layer 2 UV frame base address of input image	0x0

VO_LAYER2_IMG_IN_BADDR1_Y

Offset Address: 0x20c Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	VO_LAER2_IMG_IN_BADDR_Y	Layer 2 Y frame base address of input image	0x0

VO_LAYER2_IMG_IN_BADDR1_UV

Offset Address: 0x210 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	VO_LAER2_IMG_IN_BADDR_UV	Layer 2 UV frame base address of input image	0x0

VO_LAYER2_IMG_IN_OFFSET Offset Address: 0x214 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:29]	-	-	-	Reserved
[28:16]	RW	VO_LAYER2_IMG_IN_VSIZE	Vertical offset line number relative to start line (0) of layer 2 input image.	0x0
[15:13]	-	-	-	Reserved
[12:0]	RW	VO_LAYER2_IMG_IN_HOFFSET	Horizontal offset in double word(8-pixel) relative to start address(0) of layer 2 input image.	0x0

VO_LAYER2_IMG_IN_BLENTH Offset Address: 0x218 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:4]	-	-	-	Reserved
[3:0]	RW	VO_LAYER2_IMG_IN_BLENTH	Layer 2 DMA burst length	0x0

VO_LAYER2_IMG_IN_STRIDE Offset Address: 0x21c Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:29]	-	-	-	Reserved
[28:16]	RW	VO_LAYER2_IMG_IN_VSIZE	Vertical line size of layer 2 image in DDR.	0x0

bits	Access	Name	Descriptions	Reset
[15:13]	-	-	-	Reserved
[12:0]	RW	VO_LAYER2_IMG_IN_HSTRIDE	Horizontal stride of layer 2 image in DDR, in double word(8-pixel).As example, for image of 800x600, this field is 99. (800/8 – 1)	0x0

VO_LAYER2_IMG_IN_PIX_SIZE Offset Address: 0x220 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:29]	-	-	-	Reserved
[28:16]	RW	VO_LAYER2_IMG_IN_VSIZE	Actually reading vertical size of layer 2 image, in line unit. As example, for image of 800x600, is 599.	0x0
[15:13]	-	-	-	Reserved
[12:0]	RW	VO_LAYER2_IMG_IN_PIX_HSIZE	Actually reading horizontal size of layer 2 image, in pixel unit. As example, for image of 800x600, is 799.	0x0

VO_LAYER2_ADDR_SEL_MODE Offset Address: 0x224 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:13]	-	-	-	Reserved
[12]	RW	VO_LAYER2_ADDR_FLAG	Indicate which address set is current frame used	0x0
[11:9]	-	-	-	Reserved
[8]	RW	VO_LAYER2_ADDR_BASE1_MASK	0 : hw is NOT allowed to read base1,not start or repeat base0 1 :hw is allowed to read base1	0x0
[7:5]	-	-	-	Reserved
[4]	RW	VO_LAYER2_ADDR_BASE0_MASK	0 : hw is NOT allowed to read base0,not start or repeat base1 1 :hw is allowed to read base0	0x0
[3:1]	-	-	-	Reserved
[0]	RW	VO_LAYER2_ADDR_SEL_MODE	0: video layer 2 read buffer base0 and base1, Hw switch the pingpong address automatically. 1: video layer 2 read buffer base0 and base1, Hw switch the pingpong address according to register VO_LAYER2_ADDR_BASE0_MASK and VO_LAYER2_ADDR_BASE1_MASK	0x0

VO_LAYER3_CTL Offset Address: 0x240 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:29]	-	-	-	Reserved
[28]	RW	VO_LAYER3_UV_SWAP	Toggle this flag to swap cb cr output channel	0x0
[27:19]	-	-	-	Reserved
[18:16]	RW	VO_LAYER3_ENDIAN_MODE_UV	UV data input endian (64-bit input for example) 0: U2 V2 U3 V3 U0 V0 U1 V1 1: V2 U2 V3 U3 V0 U0 V1 U1 2: U3 V3 U2 V2 U1 V1 U0 V0 3: V3 U3 V2 U2 V1 U1 V0 U0 4: U0 V0 U1 V1 U2 V2 U3 V3 5: V0 U0 V1 U1 V2 U2 V3 U3 6: U1 V1 U0 V0 U3 V3 U2 V2 7: V1 U1 V0 U0 V3 U3 V2 U2	0x3
[15:14]	-	-	-	Reserved
[13:12]	RW	VO_LAYER3_ENDIAN_MODE_Y	Y data input endian (64-bit input for example) 0: Y4 Y5 Y6 Y7 Y0 Y1 Y2 Y3 1: Y7 Y6 Y5 Y4 Y3 Y2 Y1 Y0 2: Y0 Y1 Y2 Y3 Y4 Y5 Y6 Y7 3: Y3 Y2 Y1 Y0 Y7 Y6 Y5 Y4	0x1
[11:9]	-	-	-	Reserved
[8]	RW	VO_LAYER3_YUV420_EN	Set to '1', when layer 3 image is YUV4:2:0 format.	0x0
[7:5]	-	-	-	Reserved
[4]	RW	VO_LAYER3_YUV422_EN	Set to '1', when layer 3 image is YUV4:2:2 format.	0x0
[3:1]	-	-	-	Reserved
[0]	RW	VO_LAYER3_EN	When '1', layer 3 is enabled. When '0', layer 2 is disabled.	0x0

VO_LAYER3_IMG_IN_BADDR0_Y Offset Address: 0x244 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	VO_LAYER3_IMG_IN_BADDR_Y	Layer 3 Y frame base address of input image	0x0

VO_LAYER3_IMG_IN_BADDR0_UV Offset Address: 0x248 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	VO_LAYER3_IMG_IN_BADDR_UV	Layer 3 UV frame base address of input image	0x0

VO_LAYER3_IMG_IN_BADDR1_Y Offset Address: 0x24c Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset

bits	Access	Name	Descriptions	Reset
[31:0]	RW	VO_LAER3_IMG_IN_BADDR_Y	Layer 3 Y frame base address of input image	0x0

VO_LAYER3_IMG_IN_BADDR1_UV Offset Address: 0x250 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	VO_LAER3_IMG_IN_BADDR_UV	Layer 3 UV frame base address of input image	0x0

VO_LAYER3_IMG_IN_OFFSET Offset Address: 0x254 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:29]	-	-	-	Reserved
[28:16]	RW	VO_LAYER3_IMG_IN_VSIZE	Vertical offset line number relative to start line (0) of layer 3 input image.	0x0
[15:13]	-	-	-	Reserved
[12:0]	RW	VO_LAYER3_IMG_IN_HOFFSET	Horizontal offset in double word(8-pixel) relative to start address(0) of layer 3 input image.	0x0

VO_LAYER3_IMG_IN_BLENTH Offset Address: 0x258 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:4]	-	-	-	Reserved
[3:0]	RW	VO_LAYER3_IMG_IN_BLENTH	Layer 3 DMA burst length	0x0

VO_LAYER3_IMG_IN_STRIDE Offset Address: 0x25c Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:29]	-	-	-	Reserved
[28:16]	RW	VO_LAYER3_IMG_IN_VSIZE	Vertical line size of layer 3 image in DDR.	0x0
[15:13]	-	-	-	Reserved
[12:0]	RW	VO_LAYER3_IMG_IN_HSTRIDE	Horizontal stride of layer 3 image in DDR, in double word(8-pixel).As example, for image of 800x600, this field is 99. (800/8 – 1)	0x0

VO_LAYER3_IMG_IN_PIX_SIZE Offset Address: 0x260 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:29]	-	-	-	Reserved
[28:16]	RW	VO_LAYER3_IMG_IN_VSIZE	Actually reading vertical size of layer 3 image, in line unit. As example, for image of 800x600, is 599.	0x0
[15:13]	-	-	-	Reserved

bits	Access	Name	Descriptions	Reset
[12:0]	RW	VO_LAYER3_IMG_IN_PIX_HSIZE	Actually reading horizontal size of layer 3 image, in pixel unit. As example, for image of 800x600, is 799.	0x0

VO_LAYER3_ADDR_SEL_MODE Offset Address: 0x264 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:13]	-	-	-	Reserved
[12]	RW	VO_LAYER3_ADDR_FLAG	Indicate which address set is current frame used	0x0
[11:9]	-	-	-	Reserved
[8]	RW	VO_LAYER3_ADDR_BASE1_MASK	0 : hw is NOT allowed to read base1,not start or repeat base0 1 :hw is allowed to read base1	0x0
[7:5]	-	-	-	Reserved
[4]	RW	VO_LAYER3_ADDR_BASE0_MASK	0 : hw is NOT allowed to read base0,not start or repeat base1 1 :hw is allowed to read base0	0x0
[3:1]	-	-	-	Reserved
[0]	RW	VO_LAYER3_ADDR_SEL_MODE	0: video layer 3 read buffer base0 and base1, Hw switch the pingpong address automatically. 1: video layer 2 read buffer base0 and base1, Hw switch the pingpong address according to register VO_LAYER3_ADDR_BASE0_MASK and VO_LAYER3_ADDR_BASE1_MASK	0x0

VO_DISP_OSD0_INFO Offset Address: 0x280 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:7]	-	-	-	Reserved
[6:4]	RW	ALPHA_TPYE	when 0: Alpha data is a fixed value	
when 1: Alpha data is at alpha data block in L2_ALPHA_ADDR. when 2: Alpha data interval is aligned with R channel of OSD when 3: Alpha data interval is aligned with G channel of OSD when 4: Alpha data interval is aligned with B channel of OSD when 5: Alpha data interval is aligned with Alpha Channel of OSD for OSD type is 3, 4, and 5.	0x0			

bits	Access	Name	Descriptions	Reset
[3:0]	RW	OSD_TYPE	when 0: RGB-24bit when 1: Monochrome-8-bit when 2: RGB-16-bit (565) when 3: RGB-32-bit when 4: RGB4444 when 5: RGB1555 In mode 2, {R, G, B} value is { RGB-16-bit [15:11], 3'b100, RGB-16-bit [10:5], 2'b10, RGB-16-bit [4:0], 3'b100} In mode of 3, 4, 5, the alpha value is at the highest bits of one pixel. In mode 4, A,R,G,B value is { RGB4444[15:12], 4'b1000},{ RGB4444[11:8], 4'b1000}, { RGB4444[7:4], 4'b1000}, { RGB4444[3:0], 4'b1000}. In mode 5, alpha value is 8'hFF if MSB of RGB1555 is 1 or it is 8'h00.	0x0

VO_DISP_OSD0_SIZE Offset Address: 0x284 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:16]	RW	OSD_HEIGHT	OSD image height (in unit of pixel)	0x0
[15:0]	RW	OSD_WIDTH	OSD image width (in unit of pixel)	0x0

VO_DISP_OSD0_VLU_ADDR0 Offset Address: 0x288 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	VO_DISP_OSD_VLU_ADDR	OSD RGB image base address in DDR (should be 8-byte aligned and must be start address of new pixel).	0x0

VO_DISP_OSD0_ALP_ADDR0 Offset Address: 0x28c Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	VO_DISP_OSD_ALP_ADDR	OSD ALPHA image base address in DDR (should be 8-byte aligned and must be start address of new pixel).	0x0

VO_DISP_OSD0_VLU_ADDR1 Offset Address: 0x290 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	VO_DISP_OSD_VLU_ADDR	OSD RGB image base address in DDR (should be 8-byte aligned and must be start address of new pixel).	0x0

VO_DISP_OSD0_ALP_ADDR1

Offset Address: 0x294 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	VO_DISP_OSD_ALP_ADDR	OSD ALPHA image base address in DDR (should be 8-byte aligned and must be start address of new pixel).	0x0

VO_DISP_OSD0_DMA_CTRL Offset Address: 0x298 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:7]	-	-	-	Reserved
[6]	RW	RGB_REV	0: mem[OSD_ADDR] is R 1: mem[OSD_ADDR] is B	0x7
[5:4]	RW	DMA_MAP	DMA data map manually in a d-word (32-bit) by byte (8-bit). 00: original order 10: 1-2-3-4 to 2-1-4-3 11: 1-2-3-4 to 4-3-2-1	0x0
[3:0]	RW	DMA_REQUEST LENGHT	Beat number for DMA read request is [DMA request length] + 1.	0x7

VO_DISP_OSD0_STRIDE Offset Address: 0x29c Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:16]	RW	VO_DISP_ALP_STRIDE	ALPHA stride in unit of 8-byte for each pixel line	0x0
[15:0]	RW	VO_DISP_OSD_STRIDE	OSD stride in 8-byte unit for each pixel line	0x0

VO_OSD0_ADDR_SEL_MODE Offset Address: 0x2a0 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:13]	-	-	-	Reserved
[12]	RW	VO_OSD0_ADDR_FLAG	Indicate which address set is current frame used	0x0
[11:9]	-	-	-	Reserved
[8]	RW	VO_OSD0_ADDR_BASE1_MASK	0 : hw is NOT allowed to read base1,not start or repeat base0 1 :hw is allowed to read base1	0x0
[7:5]	-	-	-	Reserved
[4]	RW	VO_OSD0_ADDR_BASE0_MASK	0 : hw is NOT allowed to read base0,not start or repeat base1 1 :hw is allowed to read base0	0x0
[3:1]	-	-	-	Reserved
[0]	RW	VO_OSD0_ADDR_SEL_MODE	0: video OSD0 read buffer base0 and base1, Hw switch the pingpong address automatically. 1: video OSD0 read buffer base0 and base1, Hw switch the pingpong address according to register VO_OSD0_ADDR_BASE0_MASK and VO_OSD0_ADDR_BASE1_MASK	0x0

VO_DISP_OSD0_INFO Offset Address: 0x280 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:7]	-	-	-	Reserved
[6:4]	RW	ALPHA_TPYE	when 0: Alpha data is a fixed value	
when 1: Alpha data is at alpha data block in L2_ALPHA_ADDR. when 2: Alpha data interval is aligned with R channel of OSD when 3: Alpha data interval is aligned with G channel of OSD when 4: Alpha data interval is aligned with B channel of OSD when 5: Alpha data interval is aligned with Alpha Channel of OSD for OSD type is 3, 4, and 5.	0x0			

bits	Access	Name	Descriptions	Reset
[3:0]	RW	OSD_TYPE	when 0: RGB-24bit when 1: Monochrome-8-bit when 2: RGB-16-bit (565) when 3: RGB-32-bit when 4: RGB4444 when 5: RGB1555 In mode 2, {R, G, B} value is { RGB-16-bit [15:11], 3'b100, RGB-16-bit [10:5], 2'b10, RGB-16-bit [4:0], 3'b100} In mode of 3, 4, 5, the alpha value is at the highest bits of one pixel. In mode 4, A,R,G,B value is { RGB4444[15:12], 4'b1000},{ RGB4444[11:8], 4'b1000}, { RGB4444[7:4], 4'b1000}, { RGB4444[3:0], 4'b1000}. In mode 5, alpha value is 8'hFF if MSB of RGB1555 is 1 or it is 8'h00.	0x0

VO_DISP_OSD0_SIZE Offset Address: 0x284 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:16]	RW	OSD_HEIGHT	OSD image height (in unit of pixel)	0x0
[15:0]	RW	OSD_WIDTH	OSD image width (in unit of pixel)	0x0

VO_DISP_OSD0_VLU_ADDR0 Offset Address: 0x288 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	VO_DISP_OSD_VLU_ADDR	OSD RGB image base address in DDR (should be 8-byte aligned and must be start address of new pixel).	0x0

VO_DISP_OSD0_ALP_ADDR0 Offset Address: 0x28c Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	VO_DISP_OSD_ALP_ADDR	OSD ALPHA image base address in DDR (should be 8-byte aligned and must be start address of new pixel).	0x0

VO_DISP_OSD0_VLU_ADDR1 Offset Address: 0x290 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	VO_DISP_OSD_VLU_ADDR	OSD RGB image base address in DDR (should be 8-byte aligned and must be start address of new pixel).	0x0

VO_DISP_OSD0_ALP_ADDR1 Offset Address: 0x294 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	VO_DISP_OSD_ALP_ADDR	OSD ALPHA image base address in DDR (should be 8-byte aligned and must be start address of new pixel).	0x0

VO_DISP_OSD0_DMA_CTRL Offset Address: 0x298 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:7]	-	-	-	Reserved
[6]	RW	RGB_REV	0: mem[OSD_ADDR] is R 1: mem[OSD_ADDR] is B	0x7
[5:4]	RW	DMA_MAP	DMA data map manually in a d-word (32-bit) by byte (8-bit). 00: original order 10: 1-2-3-4 to 2-1-4-3 11: 1-2-3-4 to 4-3-2-1	0x0
[3:0]	RW	DMA_REQUEST LENGHT	Beat number for DMA read request is [DMA request length] + 1.	0x7

VO_DISP_OSD0_STRIDE Offset Address: 0x29c Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:16]	RW	VO_DISP_ALP_STRIDE	ALPHA stride in unit of 8-byte for each pixel line	0x0
[15:0]	RW	VO_DISP_OSD_STRIDE	OSD stride in 8-byte unit for each pixel line	0x0

VO_OSD0_ADDR_SEL_MODE Offset Address: 0x2a0 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:13]	-	-	-	Reserved

bits	Access	Name	Descriptions	Reset
[12]	RW	VO_OSD0_ADDR_FLAG	Indicate which address set is current frame used	0x0
[11:9]	-	-	-	Reserved
[8]	RW	VO_OSD0_ADDR_BASE1_MASK	0 : hw is NOT allowed to read base1,not start or repeat base0 1 :hw is allowed to read base1	0x0
[7:5]	-	-	-	Reserved
[4]	RW	VO_OSD0_ADDR_BASE0_MASK	0 : hw is NOT allowed to read base0,not start or repeat base1 1 :hw is allowed to read base0	0x0
[3:1]	-	-	-	Reserved
[0]	RW	VO_OSD0_ADDR_SEL_MODE	0: video OSD0 read buffer base0 and base1, Hw switch the pingpang address automatically. 1: video OSD0 read buffer base0 and base1, Hw switch the pingpang address according to register VO_OSD0_ADDR_BASE0_MASK and VO_OSD0_ADDR_BASE1_MASK	0x0

VO_DISP_OSD1_INFO Offset Address: 0x2c0 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:7]	-	-	-	Reserved
[6:4]	RW	ALPHA_TPYE	when 0: Alpha data is a fixed value	
when 1: Alpha data is at alpha data block in L2_ALPHA_ADDR. when 2: Alpha data interval is aligned with R channel of OSD when 3: Alpha data interval is aligned with G channel of OSD when 4: Alpha data interval is aligned with B channel of OSD when 5: Alpha data interval is aligned with Alpha Channel of OSD for OSD type is 3, 4, and 5.	0x0			

bits	Access	Name	Descriptions	Reset
[3:0]	RW	OSD_TYPE	when 0: RGB-24bit when 1: Monochrome-8-bit when 2: RGB-16-bit (565) when 3: RGB-32-bit when 4: RGB4444 when 5: RGB1555 In mode 2, {R, G, B} value is { RGB-16-bit [15:11], 3'b100, RGB-16-bit [10:5], 2'b10, RGB-16-bit [4:0], 3'b100} In mode of 3, 4, 5, the alpha value is at the highest bits of one pixel. In mode 4, A,R,G,B value is { RGB4444[15:12], 4'b1000},{ RGB4444[11:8], 4'b1000}, { RGB4444[7:4], 4'b1000}, { RGB4444[3:0], 4'b1000}. In mode 5, alpha value is 8'hFF if MSB of RGB1555 is 1 or it is 8'h00.	0x0

VO_DISP_OSD1_SIZE Offset Address: 0x2c4 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:16]	RW	OSD_HEIGHT	OSD image height (in unit of pixel)	0x0
[15:0]	RW	OSD_WIDTH	OSD image width (in unit of pixel)	0x0

VO_DISP_OSD1_VLU_ADDR0 Offset Address: 0x2c8 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	VO_DISP_OSD_VLU_ADDR	OSD RGB image base address in DDR (should be 8-byte aligned and must be start address of new pixel).	0x0

VO_DISP_OSD1_ALP_ADDR0 Offset Address: 0x2cc Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	VO_DISP_OSD_ALP_ADDR	OSD ALPHA image base address in DDR (should be 8-byte aligned and must be start address of new pixel).	0x0

VO_DISP_OSD1_VLU_ADDR1 Offset Address: 0x2d0 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	VO_DISP_OSD_VLU_ADDR	OSD RGB image base address in DDR (should be 8-byte aligned and must be start address of new pixel).	0x0

VO_DISP_OSD1_ALP_ADDR1 Offset Address: 0x2d4 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	VO_DISP_OSD_ALP_ADDR	OSD ALPHA image base address in DDR (should be 8-byte aligned and must be start address of new pixel).	0x0

VO_DISP_OSD1_DMA_CTRL Offset Address: 0x2d8 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:7]	-	-	-	Reserved
[6]	RW	RGB_REV	0: mem[OSD_ADDR] is R 1: mem[OSD_ADDR] is B	0x7
[5:4]	RW	DMA_MAP	DMA data map manually in a d-word (32-bit) by byte (8-bit). 00: original order 10: 1-2-3-4 to 2-1-4-3 11: 1-2-3-4 to 4-3-2-1	0x0
[3:0]	RW	DMA_REQUEST_LENHT	Beat number for DMA read request is [DMA request length] + 1.	0x7

VO_DISP_OSD1_STRIDE Offset Address: 0x2dc Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:16]	RW	VO_DISP_ALP_STRIDE	ALPHA stride in unit of 8-byte for each pixel line	0x0
[15:0]	RW	VO_DISP_OSD_STRIDE	OSD stride in 8-byte unit for each pixel line	0x0

VO_OSD1_ADDR_SEL_MODE Offset Address: 0x2e0 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:13]	-	-	-	Reserved

bits	Access	Name	Descriptions	Reset
[12]	RW	VO_OSD1_ADDR_FLAG	Indicate which address set is current frame used	0x0
[11:9]	-	-	-	Reserved
[8]	RW	VO_OSD1_ADDR_BASE1_MASK	0 : hw is NOT allowed to read base1,not start or repeat base0 1 :hw is allowed to read base1	0x0
[7:5]	-	-	-	Reserved
[4]	RW	VO_OSD1_ADDR_BASE0_MASK	0 : hw is NOT allowed to read base0,not start or repeat base1 1 :hw is allowed to read base0	0x0
[3:1]	-	-	-	Reserved
[0]	RW	VO_OSD1_ADDR_SEL_MODE	0: video OSD0 read buffer base0 and base1, Hw switch the pingpang address automatically. 1: video OSD0 read buffer base0 and base1, Hw switch the pingpang address according to register VO_OSD0_ADDR_BASE0_MASK and VO_OSD0_ADDR_BASE1_MASK	0x0

VO_DISP_OSD2_INFO Offset Address: 0x300 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:7]	-	-	-	Reserved
[6:4]	RW	ALPHA_TPYE	when 0: Alpha data is a fixed value	
when 1: Alpha data is at alpha data block in L2_ALPHA_ADDR. when 2: Alpha data interval is aligned with R channel of OSD when 3: Alpha data interval is aligned with G channel of OSD when 4: Alpha data interval is aligned with B channel of OSD when 5: Alpha data interval is aligned with Alpha Channel of OSD for OSD type is 3, 4, and 5.	0x0			

bits	Access	Name	Descriptions	Reset
[3:0]	RW	OSD_TYPE	when 0: RGB-24bit when 1: Monochrome-8-bit when 2: RGB-16-bit (565) when 3: RGB-32-bit when 4: RGB4444 when 5: RGB1555 In mode 2, {R, G, B} value is { RGB-16-bit [15:11], 3'b100, RGB-16-bit [10:5], 2'b10, RGB-16-bit [4:0], 3'b100} In mode of 3, 4, 5, the alpha value is at the highest bits of one pixel. In mode 4, A,R,G,B value is { RGB4444[15:12], 4'b1000},{ RGB4444[11:8], 4'b1000}, { RGB4444[7:4], 4'b1000}, { RGB4444[3:0], 4'b1000}. In mode 5, alpha value is 8'hFF if MSB of RGB1555 is 1 or it is 8'h00.	0x0

VO_DISP_OSD2_SIZE Offset Address: 0x304 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:16]	RW	OSD_HEIGHT	OSD image height (in unit of pixel)	0x0
[15:0]	RW	OSD_WIDTH	OSD image width (in unit of pixel)	0x0

VO_DISP_OSD2_VLU_ADDR0 Offset Address: 0x308 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	VO_DISP_OSD_VLU_ADDR	OSD RGB image base address in DDR (should be 8-byte aligned and must be start address of new pixel).	0x0

VO_DISP_OSD2_ALP_ADDR0 Offset Address: 0x30c Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	VO_DISP_OSD_ALP_ADDR	OSD ALPHA image base address in DDR (should be 8-byte aligned and must be start address of new pixel).	0x0

VO_DISP_OSD2_VLU_ADDR1 Offset Address: 0x310 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	VO_DISP_OSD_VLU_ADDR	OSD RGB image base address in DDR (should be 8-byte aligned and must be start address of new pixel).	0x0

VO_DISP_OSD2_ALP_ADDR1 Offset Address: 0x314 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	VO_DISP_OSD_ALP_ADDR	OSD ALPHA image base address in DDR (should be 8-byte aligned and must be start address of new pixel).	0x0

VO_DISP_OSD2_DMA_CTRL Offset Address: 0x318 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:7]	-	-	-	Reserved
[6]	RW	RGB_REV	0: mem[OSD_ADDR] is R 1: mem[OSD_ADDR] is B	0x7
[5:4]	RW	DMA_MAP	DMA data map manually in a d-word (32-bit) by byte (8-bit). 00: original order 10: 1-2-3-4 to 2-1-4-3 11: 1-2-3-4 to 4-3-2-1	0x0
[3:0]	RW	DMA_REQUEST LENGHT	Beat number for DMA read request is [DMA request length] + 1.	0x7

VO_DISP_OSD2_STRIDE Offset Address: 0x31c Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:16]	RW	VO_DISP_ALP_STRIDE	ALPHA stride in unit of 8-byte for each pixel line	0x0
[15:0]	RW	VO_DISP_OSD_STRIDE	OSD stride in 8-byte unit for each pixel line	0x0

VO_OSD2_ADDR_SEL_MODE Offset Address: 0x320 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:13]	-	-	-	Reserved

bits	Access	Name	Descriptions	Reset
[12]	RW	VO_OSD2_ADDR_FLAG	Indicate which address set is current frame used	0x0
[11:9]	-	-	-	Reserved
[8]	RW	VO_OSD2_ADDR_BASE1_MASK	0 : hw is NOT allowed to read base1,not start or repeat base0 1 :hw is allowed to read base1	0x0
[7:5]	-	-	-	Reserved
[4]	RW	VO_OSD2_ADDR_BASE0_MASK	0 : hw is NOT allowed to read base0,not start or repeat base1 1 :hw is allowed to read base0	0x0
[3:1]	-	-	-	Reserved
[0]	RW	VO_OSD2_ADDR_SEL_MODE	0: video OSD0 read buffer base0 and base1, Hw switch the pingpang address automatically. 1: video OSD0 read buffer base0 and base1, Hw switch the pingpang address according to register VO_OSD0_ADDR_BASE0_MASK and VO_OSD0_ADDR_BASE1_MASK	0x0

VO_DISP_OSD3_INFO Offset Address: 0x850 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:7]	-	-	-	Reserved
[6:4]	RW	ALPHA_TPYE	when 0: Alpha data is a fixed value	
when 1: Alpha data is at alpha data block in L2_ALPHA_ADDR. when 2: Alpha data interval is aligned with R channel of OSD when 3: Alpha data interval is aligned with G channel of OSD when 4: Alpha data interval is aligned with B channel of OSD when 5: Alpha data interval is aligned with Alpha Channel of OSD for OSD type is 3, 4, and 5.	0x0			

bits	Access	Name	Descriptions	Reset
[3:0]	RW	OSD_TYPE	when 0: RGB-24bit when 1: Monochrome-8-bit when 2: RGB-16-bit (565) when 3: RGB-32-bit when 4: RGB4444 when 5: RGB1555 In mode 2, {R, G, B} value is { RGB-16-bit [15:11], 3'b100, RGB-16-bit [10:5], 2'b10, RGB-16-bit [4:0], 3'b100} In mode of 3, 4, 5, the alpha value is at the highest bits of one pixel. In mode 4, A,R,G,B value is { RGB4444[15:12], 4'b1000},{ RGB4444[11:8], 4'b1000}, { RGB4444[7:4], 4'b1000}, { RGB4444[3:0], 4'b1000}. In mode 5, alpha value is 8'hFF if MSB of RGB1555 is 1 or it is 8'h00.	0x0

VO_DISP_OSD3_SIZE Offset Address: 0x854 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:16]	RW	OSD_HEIGHT	OSD image height (in unit of pixel)	0x0
[15:0]	RW	OSD_WIDTH	OSD image width (in unit of pixel)	0x0

VO_DISP_OSD3_VLU_ADDR0 Offset Address: 0x858 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	VO_DISP_OSD_VLU_ADDR	OSD RGB image base address in DDR (should be 8-byte aligned and must be start address of new pixel).	0x0

VO_DISP_OSD3_ALP_ADDR0 Offset Address: 0x85c Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	VO_DISP_OSD_ALP_ADDR	OSD ALPHA image base address in DDR (should be 8-byte aligned and must be start address of new pixel).	0x0

VO_DISP_OSD3_VLU_ADDR1 Offset Address: 0x860 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	VO_DISP_OSD_VLU_ADDR	OSD RGB image base address in DDR (should be 8-byte aligned and must be start address of new pixel).	0x0

VO_DISP_OSD3_ALP_ADDR1 Offset Address: 0x864 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	VO_DISP_OSD_ALP_ADDR	OSD ALPHA image base address in DDR (should be 8-byte aligned and must be start address of new pixel).	0x0

VO_DISP_OSD3_DMA_CTRL Offset Address: 0x868 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:7]	-	-	-	Reserved
[6]	RW	RGB_REV	0: mem[OSD_ADDR] is R 1: mem[OSD_ADDR] is B	0x7
[5:4]	RW	DMA_MAP	DMA data map manually in a d-word (32-bit) by byte (8-bit). 00: original order 10: 1-2-3-4 to 2-1-4-3 11: 1-2-3-4 to 4-3-2-1	0x0
[3:0]	RW	DMA_REQUEST_LENIGHT	Beat number for DMA read request is [DMA request length] + 1.	0x7

VO_DISP_OSD3_STRIDE Offset Address: 0x86c Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:16]	RW	VO_DISP_ALP_STRIDE	ALPHA stride in unit of 8-byte for each pixel line	0x0
[15:0]	RW	VO_DISP_OSD_STRIDE	OSD stride in 8-byte unit for each pixel line	0x0

VO_OSD3_ADDR_SEL_MODE Offset Address: 0x870 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:13]	-	-	-	Reserved

bits	Access	Name	Descriptions	Reset
[12]	RW	VO_OSD0_ADDR_FLAG	Indicate which address set is current frame used	0x0
[11:9]	-	-	-	Reserved
[8]	RW	VO_OSD0_ADDR_BASE1_MASK	0 : hw is NOT allowed to read base1,not start or repeat base0 1 :hw is allowed to read base1	0x0
[7:5]	-	-	-	Reserved
[4]	RW	VO_OSD0_ADDR_BASE0_MASK	0 : hw is NOT allowed to read base0,not start or repeat base1 1 :hw is allowed to read base0	0x0
[3:1]	-	-	-	Reserved
[0]	RW	VO_OSD0_ADDR_SEL_MODE	0: video OSD0 read buffer base0 and base1, Hw switch the pingpong address automatically. 1: video OSD0 read buffer base0 and base1, Hw switch the pingpong address according to register VO_OSD0_ADDR_BASE0_MASK and VO_OSD0_ADDR_BASE1_MASK	0x0

VO_DISP_OSD4_INFO Offset Address: 0x880 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:7]	-	-	-	Reserved
[6:4]	RW	ALPHA_TPYE	when 0: Alpha data is a fixed value	
when 1: Alpha data is at alpha data block in L2_ALPHA_ADDR. when 2: Alpha data interval is aligned with R channel of OSD when 3: Alpha data interval is aligned with G channel of OSD when 4: Alpha data interval is aligned with B channel of OSD when 5: Alpha data interval is aligned with Alpha Channel of OSD for OSD type is 3, 4, and 5.	0x0			

bits	Access	Name	Descriptions	Reset
[3:0]	RW	OSD_TYPE	when 0: RGB-24bit when 1: Monochrome-8-bit when 2: RGB-16-bit (565) when 3: RGB-32-bit when 4: RGB4444 when 5: RGB1555 In mode 2, {R, G, B} value is { RGB-16-bit [15:11], 3'b100, RGB-16-bit [10:5], 2'b10, RGB-16-bit [4:0], 3'b100} In mode of 3, 4, 5, the alpha value is at the highest bits of one pixel. In mode 4, A,R,G,B value is { RGB4444[15:12], 4'b1000},{ RGB4444[11:8], 4'b1000}, { RGB4444[7:4], 4'b1000}, { RGB4444[3:0], 4'b1000}. In mode 5, alpha value is 8'hFF if MSB of RGB1555 is 1 or it is 8'h00.	0x0

VO_DISP_OSD4_SIZE Offset Address: 0x884 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:16]	RW	OSD_HEIGHT	OSD image height (in unit of pixel)	0x0
[15:0]	RW	OSD_WIDTH	OSD image width (in unit of pixel)	0x0

VO_DISP_OSD4_VLU_ADDR0 Offset Address: 0x888 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	VO_DISP_OSD_VLU_ADDR	OSD RGB image base address in DDR (should be 8-byte aligned and must be start address of new pixel).	0x0

VO_DISP_OSD4_ALP_ADDR0 Offset Address: 0x88c Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	VO_DISP_OSD_ALP_ADDR	OSD ALPHA image base address in DDR (should be 8-byte aligned and must be start address of new pixel).	0x0

VO_DISP_OSD4_VLU_ADDR1 Offset Address: 0x890 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	VO_DISP_OSD_VLU_ADDR	OSD RGB image base address in DDR (should be 8-byte aligned and must be start address of new pixel).	0x0

VO_DISP_OSD4_ALP_ADDR1 Offset Address: 0x894 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	VO_DISP_OSD_ALP_ADDR	OSD ALPHA image base address in DDR (should be 8-byte aligned and must be start address of new pixel).	0x0

VO_DISP_OSD4_DMA_CTRL Offset Address: 0x898 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:7]	-	-	-	Reserved
[6]	RW	RGB_REV	0: mem[OSD_ADDR] is R 1: mem[OSD_ADDR] is B	0x7
[5:4]	RW	DMA_MAP	DMA data map manually in a d-word (32-bit) by byte (8-bit). 00: original order 10: 1-2-3-4 to 2-1-4-3 11: 1-2-3-4 to 4-3-2-1	0x0
[3:0]	RW	DMA_REQUEST LENGHT	Beat number for DMA read request is [DMA request length] + 1.	0x7

VO_DISP_OSD4_STRIDE Offset Address: 0x89c Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:16]	RW	VO_DISP_ALP_STRIDE	ALPHA stride in unit of 8-byte for each pixel line	0x0
[15:0]	RW	VO_DISP_OSD_STRIDE	OSD stride in 8-byte unit for each pixel line	0x0

VO_OSD4_ADDR_SEL_MODE Offset Address: 0x8a0 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:13]	-	-	-	Reserved

bits	Access	Name	Descriptions	Reset
[12]	RW	VO_OSD4_ADDR_FLAG	Indicate which address set is current frame used	0x0
[11:9]	-	-	-	Reserved
[8]	RW	VO_OSD4_ADDR_BASE1_MASK	0 : hw is NOT allowed to read base1,not start or repeat base0 1 :hw is allowed to read base1	0x0
[7:5]	-	-	-	Reserved
[4]	RW	VO_OSD4_ADDR_BASE0_MASK	0 : hw is NOT allowed to read base0,not start or repeat base1 1 :hw is allowed to read base0	0x0
[3:1]	-	-	-	Reserved
[0]	RW	VO_OSD4_ADDR_SEL_MODE	0: video OSD0 read buffer base0 and base1, Hw switch the pingpong address automatically. 1: video OSD0 read buffer base0 and base1, Hw switch the pingpong address according to register VO_OSD0_ADDR_BASE0_MASK and VO_OSD0_ADDR_BASE1_MASK	0x0

VO_DISP_OSD5_INFO Offset Address: 0x8b0 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:7]	-	-	-	Reserved
[6:4]	RW	ALPHA_TPYE	when 0: Alpha data is a fixed value	
when 1: Alpha data is at alpha data block in L2_ALPHA_ADDR. when 2: Alpha data interval is aligned with R channel of OSD when 3: Alpha data interval is aligned with G channel of OSD when 4: Alpha data interval is aligned with B channel of OSD when 5: Alpha data interval is aligned with Alpha Channel of OSD for OSD type is 3, 4, and 5.	0x0			

bits	Access	Name	Descriptions	Reset
[3:0]	RW	OSD_TYPE	when 0: RGB-24bit when 1: Monochrome-8-bit when 2: RGB-16-bit (565) when 3: RGB-32-bit when 4: RGB4444 when 5: RGB1555 In mode 2, {R, G, B} value is { RGB-16-bit [15:11], 3'b100, RGB-16-bit [10:5], 2'b10, RGB-16-bit [4:0], 3'b100} In mode of 3, 4, 5, the alpha value is at the highest bits of one pixel. In mode 4, A,R,G,B value is { RGB4444[15:12], 4'b1000},{ RGB4444[11:8], 4'b1000}, { RGB4444[7:4], 4'b1000}, { RGB4444[3:0], 4'b1000}. In mode 5, alpha value is 8'hFF if MSB of RGB1555 is 1 or it is 8'h00.	0x0

VO_DISP_OSD5_SIZE Offset Address: 0x8b4 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:16]	RW	OSD_HEIGHT	OSD image height (in unit of pixel)	0x0
[15:0]	RW	OSD_WIDTH	OSD image width (in unit of pixel)	0x0

VO_DISP_OSD5_VLU_ADDR0 Offset Address: 0x8b8 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	VO_DISP_OSD_VLU_ADDR	OSD RGB image base address in DDR (should be 8-byte aligned and must be start address of new pixel).	0x0

VO_DISP_OSD5_ALP_ADDR0 Offset Address: 0x8bc Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	VO_DISP_OSD_ALP_ADDR	OSD ALPHA image base address in DDR (should be 8-byte aligned and must be start address of new pixel).	0x0

VO_DISP_OSD5_VLU_ADDR1 Offset Address: 0x8c0 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	VO_DISP_OSD_VLU_ADDR	OSD RGB image base address in DDR (should be 8-byte aligned and must be start address of new pixel).	0x0

VO_DISP_OSD5_ALP_ADDR1 Offset Address: 0x8c4 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	VO_DISP_OSD_ALP_ADDR	OSD ALPHA image base address in DDR (should be 8-byte aligned and must be start address of new pixel).	0x0

VO_DISP_OSD5_DMA_CTRL [Offset Address: 0x8c8 Total Reset Value: 0x0](#)

bits	Access	Name	Descriptions	Reset
[31:7]	-	-	-	Reserved
[6]	RW	RGB_REV	0: mem[OSD_ADDR] is R 1: mem[OSD_ADDR] is B	0x7
[5:4]	RW	DMA_MAP	DMA data map manually in a d-word (32-bit) by byte (8-bit). 00: original order 10: 1-2-3-4 to 2-1-4-3 11: 1-2-3-4 to 4-3-2-1	0x0
[3:0]	RW	DMA_REQUEST LENGHT	Beat number for DMA read request is [DMA request length] + 1.	0x7

VO_DISP_OSD5_STRIDE [Offset Address: 0x8cc Total Reset Value: 0x0](#)

bits	Access	Name	Descriptions	Reset
[31:16]	RW	VO_DISP_ALP_STRIDE	ALPHA stride in unit of 8-byte for each pixel line	0x0
[15:0]	RW	VO_DISP_OSD_STRIDE	OSD stride in 8-byte unit for each pixel line	0x0

VO_OSD5_ADDR_SEL_MODE Offset Address: 0x8d0 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:13]	-	-	-	Reserved

bits	Access	Name	Descriptions	Reset
[12]	RW	VO_OSD5_ADDR_FLAG	Indicate which address set is current frame used	0x0
[11:9]	-	-	-	Reserved
[8]	RW	VO_OSD5_ADDR_BASE1_MASK	0 : hw is NOT allowed to read base1,not start or repeat base0 1 :hw is allowed to read base1	0x0
[7:5]	-	-	-	Reserved
[4]	RW	VO_OSD5_ADDR_BASE0_MASK	0 : hw is NOT allowed to read base0,not start or repeat base1 1 :hw is allowed to read base0	0x0
[3:1]	-	-	-	Reserved
[0]	RW	VO_OSD5_ADDR_SEL_MODE	0: video OSD0 read buffer base0 and base1, Hw switch the pingpang address automatically. 1: video OSD0 read buffer base0 and base1, Hw switch the pingpang address according to register VO_OSD0_ADDR_BASE0_MASK and VO_OSD0_ADDR_BASE1_MASK	0x0

VO_DISP_OSD6_INFO Offset Address: 0x8e0 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:7]	-	-	-	Reserved
[6:4]	RW	ALPHA_TPYE	when 0: Alpha data is a fixed value	
when 1: Alpha data is at alpha data block in L2_ALPHA_ADDR. when 2: Alpha data interval is aligned with R channel of OSD when 3: Alpha data interval is aligned with G channel of OSD when 4: Alpha data interval is aligned with B channel of OSD when 5: Alpha data interval is aligned with Alpha Channel of OSD for OSD type is 3, 4, and 5.	0x0			

bits	Access	Name	Descriptions	Reset
[3:0]	RW	OSD_TYPE	when 0: RGB-24bit when 1: Monochrome-8-bit when 2: RGB-16-bit (565) when 3: RGB-32-bit when 4: RGB4444 when 5: RGB1555 In mode 2, {R, G, B} value is { RGB-16-bit [15:11], 3'b100, RGB-16-bit [10:5], 2'b10, RGB-16-bit [4:0], 3'b100} In mode of 3, 4, 5, the alpha value is at the highest bits of one pixel. In mode 4, A,R,G,B value is { RGB4444[15:12], 4'b1000},{ RGB4444[11:8], 4'b1000}, { RGB4444[7:4], 4'b1000}, { RGB4444[3:0], 4'b1000}. In mode 5, alpha value is 8'hFF if MSB of RGB1555 is 1 or it is 8'h00.	0x0

VO_DISP_OSD6_SIZE Offset Address: 0x8e4 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:16]	RW	OSD_HEIGHT	OSD image height (in unit of pixel)	0x0
[15:0]	RW	OSD_WIDTH	OSD image width (in unit of pixel)	0x0

VO_DISP_OSD6_VLU_ADDR0 Offset Address: 0x8e8 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	VO_DISP_OSD_VLU_ADDR	OSD RGB image base address in DDR (should be 8-byte aligned and must be start address of new pixel).	0x0

VO_DISP_OSD6_ALP_ADDR0 Offset Address: 0x8ec Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	VO_DISP_OSD_ALP_ADDR	OSD ALPHA image base address in DDR (should be 8-byte aligned and must be start address of new pixel).	0x0

VO_DISP_OSD6_VLU_ADDR1 Offset Address: 0x8f0 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	VO_DISP_OSD_VLU_ADDR	OSD RGB image base address in DDR (should be 8-byte aligned and must be start address of new pixel).	0x0

VO_DISP_OSD6_ALP_ADDR1 Offset Address: 0x8f4 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	VO_DISP_OSD_ALP_ADDR	OSD ALPHA image base address in DDR (should be 8-byte aligned and must be start address of new pixel).	0x0

VO_DISP_OSD6_DMA_CTRL Offset Address: 0x8f8 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:7]	-	-	-	Reserved
[6]	RW	RGB_REV	0: mem[OSD_ADDR] is R 1: mem[OSD_ADDR] is B	0x7
[5:4]	RW	DMA_MAP	DMA data map manually in a d-word (32-bit) by byte (8-bit). 00: original order 10: 1-2-3-4 to 2-1-4-3 11: 1-2-3-4 to 4-3-2-1	0x0
[3:0]	RW	DMA_REQUEST LENGHT	Beat number for DMA read request is [DMA request length] + 1.	0x7

VO_DISP_OSD6_STRIDE Offset Address: 0x8fc Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:16]	RW	VO_DISP_ALP_STRIDE	ALPHA stride in unit of 8-byte for each pixel line	0x0
[15:0]	RW	VO_DISP_OSD_STRIDE	OSD stride in 8-byte unit for each pixel line	0x0

VO_OSD6_ADDR_SEL_MODE Offset Address: 0x900 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:13]	-	-	-	Reserved

bits	Access	Name	Descriptions	Reset
[12]	RW	VO_OSD6_ADDR_FLAG	Indicate which address set is current frame used	0x0
[11:9]	-	-	-	Reserved
[8]	RW	VO_OSD6_ADDR_BASE1_MASK	0 : hw is NOT allowed to read base1,not start or repeat base0 1 :hw is allowed to read base1	0x0
[7:5]	-	-	-	Reserved
[4]	RW	VO_OSD6_ADDR_BASE0_MASK	0 : hw is NOT allowed to read base0,not start or repeat base1 1 :hw is allowed to read base0	0x0
[3:1]	-	-	-	Reserved
[0]	RW	VO_OSD6_ADDR_SEL_MODE	0: video OSD0 read buffer base0 and base1, Hw switch the pingpong address automatically. 1: video OSD0 read buffer base0 and base1, Hw switch the pingpong address according to register VO_OSD6_ADDR_BASE0_MASK and VO_OSD6_ADDR_BASE1_MASK	0x0

VO_DISP_OSD7_INFO Offset Address: 0x910 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:7]	-	-	-	Reserved
[6:4]	RW	ALPHA_TPYE	when 0: Alpha data is a fixed value	
when 1: Alpha data is at alpha data block in L2_ALPHA_ADDR. when 2: Alpha data interval is aligned with R channel of OSD when 3: Alpha data interval is aligned with G channel of OSD when 4: Alpha data interval is aligned with B channel of OSD when 5: Alpha data interval is aligned with Alpha Channel of OSD for OSD type is 3, 4, and 5.	0x0			

bits	Access	Name	Descriptions	Reset
[3:0]	RW	OSD_TYPE	when 0: RGB-24bit when 1: Monochrome-8-bit when 2: RGB-16-bit (565) when 3: RGB-32-bit when 4: RGB4444 when 5: RGB1555 In mode 2, {R, G, B} value is { RGB-16-bit [15:11], 3'b100, RGB-16-bit [10:5], 2'b10, RGB-16-bit [4:0], 3'b100} In mode of 3, 4, 5, the alpha value is at the highest bits of one pixel. In mode 4, A,R,G,B value is { RGB4444[15:12], 4'b1000},{ RGB4444[11:8], 4'b1000}, { RGB4444[7:4], 4'b1000}, { RGB4444[3:0], 4'b1000}. In mode 5, alpha value is 8'hFF if MSB of RGB1555 is 1 or it is 8'h00.	0x0

VO_DISP_OSD7_SIZE Offset Address: 0x914 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:16]	RW	OSD_HEIGHT	OSD image height (in unit of pixel)	0x0
[15:0]	RW	OSD_WIDTH	OSD image width (in unit of pixel)	0x0

VO_DISP_OSD7_VLU_ADDR0 Offset Address: 0x918 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	VO_DISP_OSD_VLU_ADDR	OSD RGB image base address in DDR (should be 8-byte aligned and must be start address of new pixel).	0x0

VO_DISP_OSD7_ALP_ADDR0 Offset Address: 0x91c Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	VO_DISP_OSD_ALP_ADDR	OSD ALPHA image base address in DDR (should be 8-byte aligned and must be start address of new pixel).	0x0

VO_DISP_OSD7_VLU_ADDR1 Offset Address: 0x920 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	VO_DISP_OSD_VLU_ADDR	OSD RGB image base address in DDR (should be 8-byte aligned and must be start address of new pixel).	0x0

VO_DISP_OSD7_ALP_ADDR1 Offset Address: 0x924 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	VO_DISP_OSD_ALP_ADDR	OSD ALPHA image base address in DDR (should be 8-byte aligned and must be start address of new pixel).	0x0

VO_DISP_OSD7_DMA_CTRL Offset Address: 0x928 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:7]	-	-	-	Reserved
[6]	RW	RGB_REV	0: mem[OSD_ADDR] is R 1: mem[OSD_ADDR] is B	0x7
[5:4]	RW	DMA_MAP	DMA data map manually in a d-word (32-bit) by byte (8-bit). 00: original order 10: 1-2-3-4 to 2-1-4-3 11: 1-2-3-4 to 4-3-2-1	0x0
[3:0]	RW	DMA_REQUEST_LENHT	Beat number for DMA read request is [DMA request length] + 1.	0x7

VO_DISP_OSD7_STRIDE Offset Address: 0x92c Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:16]	RW	VO_DISP_ALP_STRIDE	ALPHA stride in unit of 8-byte for each pixel line	0x0
[15:0]	RW	VO_DISP_OSD_STRIDE	OSD stride in 8-byte unit for each pixel line	0x0

VO_OSD7_ADDR_SEL_MODE Offset Address: 0x930 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:13]	-	-	-	Reserved

bits	Access	Name	Descriptions	Reset
[12]	RW	VO_OSD7_ADDR_FLAG	Indicate which address set is current frame used	0x0
[11:9]	-	-	-	Reserved
[8]	RW	VO_OSD7_ADDR_BASE1_MASK	0 : hw is NOT allowed to read base1,not start or repeat base0 1 :hw is allowed to read base1	0x0
[7:5]	-	-	-	Reserved
[4]	RW	VO_OSD7_ADDR_BASE0_MASK	0 : hw is NOT allowed to read base0,not start or repeat base1 1 :hw is allowed to read base0	0x0
[3:1]	-	-	-	Reserved
[0]	RW	VO_OSD7_ADDR_SEL_MODE	0: video OSD0 read buffer base0 and base1, Hw switch the pingpong address automatically. 1: video OSD0 read buffer base0 and base1, Hw switch the pingpong address according to register VO_OSD7_ADDR_BASE0_MASK and VO_OSD7_ADDR_BASE1_MASK	0x0

VO_LAYER0_CFG Offset Address: 0xa00 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:11]	-	-	-	Reserved
[10]	RW	layer0_gray_en	1: enable gray mode	0x0
[9]	RW	layer0_uv_swap	0: UV 1: VU	0x0
[8]	RW	layer0_yuv422_en	1: yuv 422 0 : yuv420	0x0
[7:6]	RW	layer0_mirror_mode	bit 6 x mirror bit 7 y mirror	0x0
[5:4]	RW	layer0_rotate_mode	0 : 0 1 : 90 2 180 3 : 270	0x0
[3]	RW	layer0_blenh_4or8	0: burst length 8 1: burst length 4	0x0
[2:0]	RW	layer0_max_outstanding	0: outstanding 1 1: outstanding 2 7: outstanding 8	0x0

VO_LAYER0_SIZE Offset Address: 0xa04 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:28]	-	-	-	Reserved
[27:16]	RW	layer0_vsize	graphic height	0x0
[15:12]	-	-	-	Reserved

bits	Access	Name	Descriptions	Reset
[11:0]	RW	layer0_hsize	graphic width	0x0

VO_LAYER0_STRIDE Offset Address: 0xa08 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:16]	RW	layer0_uv_stride	layer0 uv channel stride	0x0
[15:0]	RW	layer0_y_stride	layer0 y channel stride	0x0

VO_LAYER0_BADDR_Y0 Offset Address: 0xa0c Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	layer0_baddr_y0	layer0 y channel base address	0x0

VO_LAYER0_BADDR_UV0 Offset Address: 0xa10 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	layer0_baddr_uv0	layer0 uv channel base address	0x0

VO_LAYER0_BADDR_Y1 Offset Address: 0xa14 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	layer0_baddr_y1	layer0 y channel base address	0x0

VO_LAYER0_BADDR_UV1 Offset Address: 0xa18 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	layer0_baddr_uv1	layer0 uv channel base address	0x0

VO_LAYER0_ADDR_SEL_MODE Offset Address: 0xa1c Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:13]	-	-	-	Reserved
[12]	RW	VO_LAYER0_ADDR_FLAG	Indicate which address set is current frame used	0x0
[11:9]	-	-	-	Reserved
[8]	RW	VO_LAYER0_ADDR_BASE1_MASK	0 : hw is NOT allowed to read base1,not start or repeat base0 1 :hw is allowed to read base1	0x0
[7:5]	-	-	-	Reserved
[4]	RW	VO_LAYER0_ADDR_BASE0_MASK	0 : hw is NOT allowed to read base0,not start or repeat base1 1 :hw is allowed to read base0	0x0
[3:1]	-	-	-	Reserved

bits	Access	Name	Descriptions	Reset
[0]	RW	VO_LAYER0_ADDR_SEL_MODE	0: video OSD0 read buffer base0 and base1, Hw switch the pingpong address automatically. 1: video OSD0 read buffer base0 and base1, Hw switch the pingpong address according to register VO_LAYER0_ADDR_BASE0_MASK and VO_LAYER0_ADDR_BASE1_MASK	0x0

VO_LAYER1_CFG Offset Address: 0xa20 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:11]	-	-	-	Reserved
[10]	RW	layer0_gray_en	1: enable gray mode	0x0
[9]	RW	layer0_uv_swap	0: UV 1: VU	0x0
[8]	RW	layer0_yuv422_en	1: yuv 422 0 : yuv420	0x0
[7:6]	RW	layer0_mirror_mode	bit 6 x mirror bit 7 y mirror	0x0
[5:4]	RW	layer0_rotate_mode	0 : 0 1 : 90 2 180 3 : 270	0x0
[3]	RW	layer0_blenh_4or8	0: burst length 8 1: burst length 4	0x0
[2:0]	RW	layer0_max_outstanding	0: outstanding 1 1: outstanding 2 7: outstanding 8	0x0

VO_LAYER1_SIZE Offset Address: 0xa24 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:28]	-	-	-	Reserved
[27:16]	RW	layer0_vsize	graphic height	0x0
[15:12]	-	-	-	Reserved
[11:0]	RW	layer0_hsize	graphic width	0x0

VO_LAYER1_STRIDE Offset Address: 0xa28 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:16]	RW	layer0_uv_stride	layer0 uv channel stride	0x0
[15:0]	RW	layer0_y_stride	layer0 y channel stride	0x0

VO_LAYER1_BADDR_Y0 Offset Address: 0xa2c Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	layer0_baddr_y0	layer0 y channel base address	0x0

VO_LAYER1_BADDR_UV0 Offset Address: 0xa30 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	layer0_baddr_uv0	layer0 uv channel base address	0x0

VO_LAYER1_BADDR_Y1 Offset Address: 0xa34 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	layer0_baddr_y1	layer0 y channel base address	0x0

VO_LAYER1_BADDR_UV1 Offset Address: 0xa38 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	layer0_baddr_uv1	layer0 uv channel base address	0x0

VO_LAYER1_ADDR_SEL_MODE Offset Address: 0xa3c Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:13]	-	-	-	Reserved
[12]	RW	VO_LAYER0_ADDR_FLAG	Indicate which address set is current frame used	0x0
[11:9]	-	-	-	Reserved
[8]	RW	VO_LAYER0_ADDR_BASE1_MASK	0 : hw is NOT allowed to read base1,not start or repeat base0 1 :hw is allowed to read base1	0x0
[7:5]	-	-	-	Reserved
[4]	RW	VO_LAYER0_ADDR_BASE0_MASK	0 : hw is NOT allowed to read base0,not start or repeat base1 1 :hw is allowed to read base0	0x0
[3:1]	-	-	-	Reserved
[0]	RW	VO_LAYER0_ADDR_SEL_MODE	0: video OSD0 read buffer base0 and base1, Hw switch the pingpang address automatically. 1: video OSD0 read buffer base0 and base1, Hw switch the pingpang address according to register VO_LAYER0_ADDR_BASE0_MASK and VO_LAYER0_ADDR_BASE1_MASK	0x0

VO_OSD_RGB2YUV_CTL Offset Address: 0x340 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:29]	-	-	-	Reserved

bits	Access	Name	Descriptions	Reset
[28]	RW	VO_OSD7_RGB2YUV_EN	Set to '1', RGB -> YUV444 in OSD 0 layer is enabled. Set to '0', RGB -> YUV444 in OSD 0 layer is disabled.	0x1
[27:25]	-	-	-	Reserved
[24]	RW	VO_OSD6_RGB2YUV_EN	Set to '1', RGB -> YUV444 in OSD 0 layer is enabled. Set to '0', RGB -> YUV444 in OSD 0 layer is disabled.	0x1
[23:21]	-	-	-	Reserved
[20]	RW	VO_OSD5_RGB2YUV_EN	Set to '1', RGB -> YUV444 in OSD 0 layer is enabled. Set to '0', RGB -> YUV444 in OSD 0 layer is disabled.	0x1
[19:17]	-	-	-	Reserved
[16]	RW	VO_OSD4_RGB2YUV_EN	Set to '1', RGB -> YUV444 in OSD 0 layer is enabled. Set to '0', RGB -> YUV444 in OSD 0 layer is disabled.	0x1
[15:13]	-	-	-	Reserved
[12]	RW	VO_OSD3_RGB2YUV_EN	Set to '1', RGB -> YUV444 in OSD 0 layer is enabled. Set to '0', RGB -> YUV444 in OSD 0 layer is disabled.	0x1
[11:9]	-	-	-	Reserved
[8]	RW	VO_OSD2_RGB2YUV_EN	Set to '1', RGB -> YUV444 in OSD 0 layer is enabled. Set to '0', RGB -> YUV444 in OSD 0 layer is disabled.	0x1
[7:5]	-	-	-	Reserved
[4]	RW	VO_OSD1_RGB2YUV_EN	Set to '1', RGB -> YUV444 in OSD 1 layer is enabled. Set to '0', RGB -> YUV444 in OSD 1 layer is disabled.	0x1
[3:1]	-	-	-	Reserved
[0]	RW	VO_OSD0_RGB2YUV_EN	Set to '1', RGB -> YUV444 in OSD 0 layer is enabled. Set to '0', RGB -> YUV444 in OSD 0 layer is disabled.	0x1

VO_OSD_RGB2YUV_COEFF00 Offset Address: 0x344 Total Reset Value: 0x132

bits	Access	Name	Descriptions	Reset
[31:12]	-	-	-	Reserved
[11:0]	RW	VO_OSD_RGB2YUV_COEFF00	RGB -> YUV444 coefficient.	0x132

VO_OSD_RGB2YUV_COEFF01 Offset Address: 0x348 Total Reset Value: 0x132

bits	Access	Name	Descriptions	Reset
[31:12]	-	-	-	Reserved
[11:0]	RW	VO_OSD_RGB2YUV_COEFF01	RGB -> YUV444 coefficient.	0x259

VO_OSD_RGB2YUV_COEFF02 Offset Address: 0x34c Total Reset Value: 0x132

bits	Access	Name	Descriptions	Reset
[31:12]	-	-	-	Reserved
[11:0]	RW	VO_OSD_RGB2YUV_COEFF02	RGB -> YUV444 coefficient.	0x75

VO_OSD_RGB2YUV_COEFF03 Offset Address: 0x350 Total Reset Value: 0x132

bits	Access	Name	Descriptions	Reset
[31:12]	-	-	-	Reserved
[11:0]	RW	VO_OSD_RGB2YUV_COEFF03	RGB -> YUV444 coefficient.	0x0

VO_OSD_RGB2YUV_COEFF10 Offset Address: 0x354 Total Reset Value: 0x132

bits	Access	Name	Descriptions	Reset
[31:12]	-	-	-	Reserved
[11:0]	RW	VO_OSD_RGB2YUV_COEFF10	RGB -> YUV444 coefficient.	0xf50

VO_OSD_RGB2YUV_COEFF11 Offset Address: 0x358 Total Reset Value: 0x132

bits	Access	Name	Descriptions	Reset
[31:12]	-	-	-	Reserved
[11:0]	RW	VO_OSD_RGB2YUV_COEFF11	RGB -> YUV444 coefficient.	0xea5

VO_OSD_RGB2YUV_COEFF12 Offset Address: 0x35c Total Reset Value: 0x132

bits	Access	Name	Descriptions	Reset
[31:12]	-	-	-	Reserved
[11:0]	RW	VO_OSD_RGB2YUV_COEFF12	RGB -> YUV444 coefficient.	0x20b

VO_OSD_RGB2YUV_COEFF13 Offset Address: 0x360 Total Reset Value: 0x132

bits	Access	Name	Descriptions	Reset
[31:12]	-	-	-	Reserved

bits	Access	Name	Descriptions	Reset
[11:0]	RW	VO_OSD_RGB2YUV_COEFF13	RGB -> YUV444 coefficient.	0x80

VO_OSD_RGB2YUV_COEFF20 Offset Address: 0x364 Total Reset Value: 0x132

bits	Access	Name	Descriptions	Reset
[31:12]	-	-	-	Reserved
[11:0]	RW	VO_OSD_RGB2YUV_COEFF20	RGB -> YUV444 coefficient.	0x20b

VO_OSD_RGB2YUV_COEFF21 Offset Address: 0x368 Total Reset Value: 0x132

bits	Access	Name	Descriptions	Reset
[31:12]	-	-	-	Reserved
[11:0]	RW	VO_OSD_RGB2YUV_COEFF21	RGB -> YUV444 coefficient.	0xe4a

VO_OSD_RGB2YUV_COEFF22 Offset Address: 0x36c Total Reset Value: 0x132

bits	Access	Name	Descriptions	Reset
[31:12]	-	-	-	Reserved
[11:0]	RW	VO_OSD_RGB2YUV_COEFF22	RGB -> YUV444 coefficient.	0xfab

VO_OSD_RGB2YUV_COEFF23 Offset Address: 0x370 Total Reset Value: 0x132

bits	Access	Name	Descriptions	Reset
[31:12]	-	-	-	Reserved
[11:0]	RW	VO_OSD_RGB2YUV_COEFF23	RGB -> YUV444 coefficient.	0x80

VO_DISP_YUV2RGB_CTL Offset Address: 0x380 Total Reset Value: 0x132

bits	Access	Name	Descriptions	Reset
[31:1]	-	-	-	Reserved
[0]	RW	VO_DISP_YUV2RGB_EN	When '1', YUV444 -> RGB after display mixer is enabled. When '0', YUV444 -> RGB after display mixer is disabled.	0x1

VO_OSD_YUV2RGB_COEFF00 Offset Address: 0x384 Total Reset Value: 0x132

bits	Access	Name	Descriptions	Reset
[31:12]	-	-	-	Reserved
[11:0]	RW	VO_OSD_YUV2RGB_COEFF00	RGB -> YUV444 coefficient.	0x400

VO_OSD_YUV2RGB_COEFF01 Offset Address: 0x388 Total Reset Value: 0x132

bits	Access	Name	Descriptions	Reset

bits	Access	Name	Descriptions	Reset
[31:12]	-	-	-	Reserved
[11:0]	RW	VO_OSD_YUV2RGB_COEFF01	RGB -> YUV444 coefficient.	0x0

VO_OSD_YUV2RGB_COEFF02 Offset Address: 0x38c Total Reset Value: 0x132

bits	Access	Name	Descriptions	Reset
[31:12]	-	-	-	Reserved
[11:0]	RW	VO_OSD_YUV2RGB_COEFF02	RGB -> YUV444 coefficient.	0x5a1

VO_OSD_YUV2RGB_COEFF03 Offset Address: 0x390 Total Reset Value: 0x132

bits	Access	Name	Descriptions	Reset
[31:12]	-	-	-	Reserved
[11:0]	RW	VO_OSD_YUV2RGB_COEFF03	RGB -> YUV444 coefficient.	0xf4c

VO_OSD_YUV2RGB_COEFF10 Offset Address: 0x394 Total Reset Value: 0x132

bits	Access	Name	Descriptions	Reset
[31:12]	-	-	-	Reserved
[11:0]	RW	VO_OSD_YUV2RGB_COEFF10	RGB -> YUV444 coefficient.	0x400

VO_OSD_YUV2RGB_COEFF11 Offset Address: 0x398 Total Reset Value: 0x132

bits	Access	Name	Descriptions	Reset
[31:12]	-	-	-	Reserved
[11:0]	RW	VO_OSD_YUV2RGB_COEFF11	RGB -> YUV444 coefficient.	0xe9e

VO_OSD_YUV2RGB_COEFF12 Offset Address: 0x39c Total Reset Value: 0x132

bits	Access	Name	Descriptions	Reset
[31:12]	-	-	-	Reserved
[11:0]	RW	VO_OSD_YUV2RGB_COEFF12	RGB -> YUV444 coefficient.	0xd22

VO_OSD_YUV2RGB_COEFF13 Offset Address: 0x3a0 Total Reset Value: 0x132

bits	Access	Name	Descriptions	Reset
[31:12]	-	-	-	Reserved
[11:0]	RW	VO_OSD_YUV2RGB_COEFF13	RGB -> YUV444 coefficient.	0x88

VO_OSD_YUV2RGB_COEFF20 Offset Address: 0x3a4 Total Reset Value: 0x132

bits	Access	Name	Descriptions	Reset
[31:12]	-	-	-	Reserved

bits	Access	Name	Descriptions	Reset
[11:0]	RW	VO_OSD_YUV2RGB_COEFF20	RGB -> YUV444 coefficient.	0x400

VO_OSD_YUV2RGB_COEFF21 Offset Address: 0x3a8 Total Reset Value: 0x132

bits	Access	Name	Descriptions	Reset
[31:12]	-	-	-	Reserved
[11:0]	RW	VO_OSD_YUV2RGB_COEFF21	RGB -> YUV444 coefficient.	0x71e

VO_OSD_YUV2RGB_COEFF22 Offset Address: 0x3ac Total Reset Value: 0x132

bits	Access	Name	Descriptions	Reset
[31:12]	-	-	-	Reserved
[11:0]	RW	VO_OSD_YUV2RGB_COEFF22	RGB -> YUV444 coefficient.	0x0

VO_OSD_YUV2RGB_COEFF23 Offset Address: 0x3b0 Total Reset Value: 0x132

bits	Access	Name	Descriptions	Reset
[31:12]	-	-	-	Reserved
[11:0]	RW	VO_OSD_YUV2RGB_COEFF23	RGB -o> YUV444 coefficient.	0xf1c

VO_DISP_MIX_LAYER_GLB_ALPHA0 Offset Address: 0x3c4 Total Reset Value: 0xffffffff

bits	Access	Name	Descriptions	Reset
[31:24]	RW	VO_DISP_LAYER3_GLB_ALPHA	layer 3 global alpha value	0xff
[23:16]	RW	VO_DISP_LAYER2_GLB_ALPHA	layer 2 global alpha value	0xff
[15:8]	RW	VO_DISP_LAYER1_GLB_ALPHA	layer 1 global alpha value	0xff
[7:0]	RW	VO_DISP_LAYER0_GLB_ALPHA	layer 0 global alpha value	0xff

VO_DISP_MIX_LAYER_GLB_ALPHA1 Offset Address: 0x3c8 Total Reset Value: 0xffffffff

bits	Access	Name	Descriptions	Reset
[31:24]	RW	VO_DISP_OSD3_GLB_ALPHA	OSD 3 global alpha value	0xff
[23:16]	RW	VO_DISP_OSD2_GLB_ALPHA	OSD 2 global alpha value	0xff
[15:8]	RW	VO_DISP_OSD1_GLB_ALPHA	OSD 1 global alpha value	0xff
[7:0]	RW	VO_DISP_OSD0_GLB_ALPHA	OSD 0 global alpha value	0xff

VO_DISP_MIX_LAYER_GLB_ALPHA2 Offset Address: 0x940 Total Reset Value: 0xffffffff

bits	Access	Name	Descriptions	Reset
[31:24]	RW	VO_DISP_OSD7_GLB_ALPHA	OSD 7 global alpha value	0xff
[23:16]	RW	VO_DISP_OSD6_GLB_ALPHA	OSD 6 global alpha value	0xff
[15:8]	RW	VO_DISP_OSD5_GLB_ALPHA	OSD 5 global alpha value	0xff

bits	Access	Name	Descriptions	Reset
[7:0]	RW	VO_DISP_OSD4_GLB_ALPHA	OSD 4 global alpha value	0xff

VO_DISP_MIX_SEL0 Offset Address: 0x3cc Total Reset Value: 0x3210

bits	Access	Name	Descriptions	Reset
[31:16]	-	-	-	Reserved
[15:12]	RW	VO_DISP_LAYER3_SEL	Mixer layer 3 source selection: 4'h0: layer 0 4'h1: layer 1 4'h2: layer 2 4'h3: layer 3 4'h4: OSD0 4'h5: OSD1 4'h6: OSD2 4'h7: OSD3 4'h8: OSD4 4'h9: OSD5 4'h10: OSD6 4'h11: OSD7	0x3
[11:8]	RW	VO_DISP_LAYER2_SEL	Mixer layer 2 source selection: 4'h0: layer 0 4'h1: layer 1 4'h2: layer 2 4'h3: layer 3 4'h4: OSD0 4'h5: OSD1 4'h6: OSD2 4'h7: OSD3 4'h8: OSD4 4'h9: OSD5 4'h10: OSD6 4'h11: OSD7	0x2
[7:4]	RW	VO_DISP_LAYER1_SEL	Mixer layer 1 source selection: 4'h0: layer 0 4'h1: layer 1 4'h2: layer 2 4'h3: layer 3 4'h4: OSD0 4'h5: OSD1 4'h6: OSD2 4'h7: OSD3 4'h8: OSD4 4'h9: OSD5 4'h10: OSD6 4'h11: OSD7	0x1

bits	Access	Name	Descriptions	Reset
[3:0]	RW	VO_DISP_LAYER0_SEL	Mixer layer 0 source selection: 4'h0: layer 0 4'h1: layer 1 4'h2: layer 2 4'h3: layer 3 4'h4: OSD0 4'h5: OSD1 4'h6: OSD2 4'h7: OSD3 4'h8: OSD4 4'h9: OSD5 4'h10: OSD6 4'h11: OSD7	0x0

VO_DISP_MIX_SEL1 Offset Address: 0x950 Total Reset Value: 0x3210

bits	Access	Name	Descriptions	Reset
[831:28]	RW	VO_DISP_OSD7_SEL	Mixer OSD 7 source selection: 4'h0: layer 0 4'h1: layer 1 4'h2: layer 2 4'h3: layer 3 4'h4: OSD0 4'h5: OSD1 4'h6: OSD2 4'h7: OSD3 4'h8: OSD4 4'h9: OSD5 4'h10: OSD6 4'h11: OSD7	0x7
[27:24]	RW	VO_DISP_OSD6_SEL	Mixer OSD 6 source selection: 4'h0: layer 0 4'h1: layer 1 4'h2: layer 2 4'h3: layer 3 4'h4: OSD0 4'h5: OSD1 4'h6: OSD2 4'h7: OSD3 4'h8: OSD4 4'h9: OSD5 4'h10: OSD6 4'h11: OSD7	0x7

bits	Access	Name	Descriptions	Reset
[23:20]	RW	VO_DISP_OSD5_SEL	Mixer OSD 5 source selection: 4'h0: layer 0 4'h1: layer 1 4'h2: layer 2 4'h3: layer 3 4'h4: OSD0 4'h5: OSD1 4'h6: OSD2 4'h7: OSD3 4'h8: OSD4 4'h9: OSD5 4'h10: OSD6 4'h11: OSD7	0x9
[19:16]	RW	VO_DISP_OSD4_SEL	Mixer OSD 4 source selection: 4'h0: layer 0 4'h1: layer 1 4'h2: layer 2 4'h3: layer 3 4'h4: OSD0 4'h5: OSD1 4'h6: OSD2 4'h7: OSD3 4'h8: OSD4 4'h9: OSD5 4'h10: OSD6 4'h11: OSD7	0x8
[15:12]	RW	VO_DISP_OSD3_SEL	Mixer OSD 3 source selection: 4'h0: layer 0 4'h1: layer 1 4'h2: layer 2 4'h3: layer 3 4'h4: OSD0 4'h5: OSD1 4'h6: OSD2 4'h7: OSD3 4'h8: OSD4 4'h9: OSD5 4'h10: OSD6 4'h11: OSD7	0x7

bits	Access	Name	Descriptions	Reset
[11:8]	RW	VO_DISP_OSD2_SEL	Mixer OSD 2 source selection: 4'h0: layer 0 4'h1: layer 1 4'h2: layer 2 4'h3: layer 3 4'h4: OSD0 4'h5: OSD1 4'h6: OSD2 4'h7: OSD3 4'h8: OSD4 4'h9: OSD5 4'h10: OSD6 4'h11: OSD7	0x6
[7:4]	RW	VO_DISP_OSD1_SEL	Mixer OSD 1 source selection: 4'h0: layer 0 4'h1: layer 1 4'h2: layer 2 4'h3: layer 3 4'h4: OSD0 4'h5: OSD1 4'h6: OSD2 4'h7: OSD3 4'h8: OSD4 4'h9: OSD5 4'h10: OSD6 4'h11: OSD7	0x5
[3:0]	RW	VO_DISP_OSD0_SEL	Mixer OSD 0 source selection: 4'h0: layer 0 4'h1: layer 1 4'h2: layer 2 4'h3: layer 3 4'h4: OSD0 4'h5: OSD1 4'h6: OSD2 4'h7: OSD3 4'h8: OSD4 4'h9: OSD5 4'h10: OSD6 4'h11: OSD7	0x4

VO_DISP_BACKGROUND Offset Address: 0x3d0 Total Reset Value: 0xff

bits	Access	Name	Descriptions	Reset
[31:24]	-	-	-	Reserved
[23:16]	RW	VO_DISP_BACKGROUND_V	B component color of background	0x0
[15:8]	RW	VO_DISP_BACKGROUND_U	G component color of background	0x0

bits	Access	Name	Descriptions	Reset
[7:0]	RW	VO_DISP_BACKGROUND_Y	R component color of background	0xff

VO_DISP_DITH_CTL Offset Address: 0x3d4 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:1]	-	-	-	Reserved
[0]	RW	VO_DISP_DITH_EN	when '1', 10 to 8-bit dither after display mixer is enabled. when '0', 10 to 8-bit dither after display mixer is disabled.	0x0

VO_DISP_CLUT_CTL Offset Address: 0x3d8 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:1]	-	-	-	Reserved
[0]	RW	VO_DISP_CLUT_EN	when '1', gamma correction is enabled. when '0', gamma correction is disabled.	0x0

VO_DISP_IRQ0_CTL Offset Address: 0x3e0 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:21]	-	-	-	Reserved
[20]	RW	VO_DISP_IRQ0_EN	when '1', irq0 (Vsync IRQ) is enabled. when '0', irq0 (Vsync IRQ) is disabled.	0x0
[19:0]	-	-	-	Reserved

VO_DISP_IRQ1_CTL Offset Address: 0x3e4 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:21]	-	-	-	Reserved
[20]	RW	VO_DISP_IRQ1_EN	when '1', irq1 (Line number trigger IRQ) is enabled. when '0', irq1 (Line number trigger IRQ) is disabled.	0x0
[19:0]	-	-	-	Reserved
[12:0]	RW	VO_DISP_IRQ1_VPOS	Multiple line numbers of 2 to the power of n (n =VO_DISP_IRQ1_VPOS) to trigger irq2. And, irq2 is high active, level.	0x0

VO_DISP_IRQ2_CTL Offset Address: 0x3e8 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:21]	-	-	-	Reserved
[20]	RW	VO_DISP_IRQ2_EN	when '1', irq2 (DPI buffer underrun IRQ) is enabled. when '0', irq2 (DPI buffer underrun IRQ) is disabled.	0x0
[19:0]	-	-	-	Reserved

VO_DISP_IRQ Offset Address: 0x3ec Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:29]	-	-	-	Reserved
[28]	W	VO_ADDR_FLAG	Current frame use address set 0 or 1	0x0
[27:24]	-	-	-	Reserved
[23:16]	W	VO_UNDERRUN_CNT	DPI buffer under flow counter in 64 frames	0x0
[14]	W	OSD7_underflow_clr	Write '1' to clear OSD7_underflow, write only.	0x0
[14]	R	OSD7_underflow	Vo OSD7 underflow status, read only.	0x0
[13]	W	OSD6_underflow_clr	Write '1' to clear OSD6_underflow, write only.	0x0
[13]	R	OSD6_underflow	Vo OSD6 underflow status, read only.	0x0
[12]	W	OSD5_underflow_clr	Write '1' to clear OSD5_underflow, write only.	0x0
[12]	R	OSD5_underflow	Vo OSD5 underflow status, read only.	0x0
[11]	W	OSD4_underflow_clr	Write '1' to clear OSD4_underflow, write only.	0x0
[11]	R	OSD4_underflow	Vo OSD4 underflow status, read only.	0x0
[10]	W	OSD3_underflow_clr	Write '1' to clear OSD3_underflow, write only.	0x0
[10]	R	OSD3_underflow	Vo OSD3 underflow status, read only.	0x0
[9]	W	OSD2_underflow_clr	Write '1' to clear OSD2_underflow, write only.	0x0
[9]	R	OSD2_underflow	Vo OSD2 underflow status, read only.	0x0
[8]	W	OSD1_underflow_clr	Write '1' to clear OSD1_underflow, write only.	0x0
[8]	R	OSD1_underflow	Vo OSD1 underflow status, read only.	0x0
[7]	W	OSD0_underflow_clr	Write '1' to clear OSD0_underflow, write only.	0x0
[7]	R	OSD0_underflow	Vo OSD0 underflow status, read only.	0x0
[6]	W	Layer3_underflow_clr	Write '1' to clear Layer3_underflow, write only.	0x0
[6]	R	Layer3_underflow	Vo layer3 underflow status, read only.	0x0
[5]	W	Layer2_underflow_clr	Write '1' to clear Layer2_underflow, write only.	0x0
[5]	R	Layer2_underflow	Vo layer2 underflow status, read only.	0x0
[4]	W	Layer1_underflow_clr	Write '1' to clear Layer1_underflow, write only.	0x0
[4]	R	Layer1_underflow	Vo layer1 underflow status, read only.	0x0
[3]	W	Layer0_underflow_clr	Write '1' to clear Layer0_underflow, write only.	0x0
[3]	R	Layer0_underflow	Vo layer0 underflow status, read only.	0x0
[2]	W	VO_DISP_IRQ2_CLR	Write '1' to clear VO_DISP_IRQ2,write only.	0x0
[2]	R	VO_DISP_IRQ2	VO_DISP_IRQ2 status, read only.	0x0
[1]	W	VO_DISP_IRQ1_CLR	Write '1' to clear VO_DISP_IRQ1,write only.	0x0
[1]	R	VO_DISP_IRQ1	VO_DISP_IRQ1 status, read only.	0x0
[0]	W	VO_DISP_IRQ0_CLR	Write '1' to clear VO_DISP_IRQ0,write only.	0x0
[0]	R	VO_DISP_IRQ0	VO_DISP_IRQ0 status, read only.	0x0

VO_WB_DMA_CH_WR_EN Offset Address: 0x400 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:1]	-	-	-	Reserved
[0]	RW	VO_WB_DMA_CH_WR_EN	VO_WB_DMA_WR_EN 1'b0: disable ; 1'b1: enable	0x0

VO_WB_MODE Offset Address: 0x404 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:3]	-	-	-	Reserved
[2]	RW	IN_CH_MODE	Input channel mode, 1x(2 ccyle/pixel)/2x(one cycle/pixel) 0: BT656; 1:BT1120/mipi	
only valid for DVP Interface	0x0			
[1:0]	-	-	-	Reserved

VO_WB_PIXEL_FORMAT_OUT Offset Address: 0x408 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:8]	-	-	-	Reserved
[7:5]	RW		RGB/RAW pixel width 0:8bit; 1:10bit; 2:12bit; 3:14bit; 4:16bit; 5:18bit; 6:20bit; 7:24bit;	0x0
[4:3]	RW		YUV422 pixel order 0:YUYV 1:VYU 2:UYVY 3:VYUY	0x0
[2]	RW		YUV/RGB out put format 0:YUV422;1:YUV420(YUV)	0x0
[1]	RW		YUV input format 0:YUV422;1:YUV420	0x0
[0]	RW		YUV/RGB(RAW) format, 0:RGB(RAW);1:YUV	0x0

VO_WB_FRM_BASE_OUT_ADDR0_Y Offset Address: 0x40c Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset

bits	Access	Name	Descriptions	Reset
[31:0]	RW	VO_WB_FRM_BASE_OUT_ADDR0_Y	Write Back base address(Y0)	0x0

VO_WB_FRM_BASE_OUT_ADDR1_Y Offset Address: 0x410 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	VO_WB_FRM_BASE_OUT_ADDR1_Y	Write Back base address(Y1)	0x0

VO_WB_FRM_BASE_OUT_ADDR0_UV Offset Address: 0x414 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	VO_WB_FRM_BASE_OUT_ADDR0_UV	Write Back base address(UV0)	0x0

VO_WB_FRM_BASE_OUT_ADDR1_UV Offset Address: 0x418 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:0]	RW	VO_WB_FRM_BASE_OUT_ADDR1_UV	Write Back base address(UV1)	0x0

VO_WB_HSTRIDE_OUT_Y Offset Address: 0x41c Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:13]	-	-	-	Reserved
[12:0]	RW	VO_WB_HSTRIDE_OUT_Y	Write Back line stride (Y)	0x0

VO_WB_HSTRIDE_OUT_UV VO_OSD_RGB2YUV_COEFF21 Offset Address: 0x420 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:13]	-	-	-	Reserved
[12:0]	RW	VO_WB_HSTRIDE_OUT_UV	Write Back line stride (UV)	0x0

VO_WB_FRM_BUF_SIZE_OUT Offset Address: 0x424 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:16]	-	-	-	Reserved
[15:0]	RW	VO_WB_FRM_BUF_SIZE_OUT	Write Back frame buffer size, 64KByte unit, 64K*N/Stride=Int (N=1,2,4,8)	0x0

VO_WB_V_SIZE_ACTIVE_OUT Offset Address: 0x428 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:13]	-	-	-	Reserved
[12:0]	RW	VO_WB_V_SIZE	Write Back Active vertical size,(actual -1),	0x0

VO_WB_H_SIZE_ACTIVE_OUT Offset Address: 0x42c Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset

bits	Access	Name	Descriptions	Reset
[31:13]	-	-	-	Reserved
[12:0]	RW	VO_WB_H_SIZE	Write Back Active horizontal size,(actual -1),	0x0

VO_WB_CH0_DMA_MODE Offset Address: 0x440 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:29]	-	-	-	Reserved
[28]	RW	VO_WB_DMA_WR_ERR_MODE_EN	dma write channel error detection enable	0x0
[27:16]	-	-	-	Reserved
[15]	RW	VO_WB_DMA_WR_OUT_SWAP	dma Y/UV output swap enable 1'b0: 64bit mode {P0,P1,P2,P3,P4,P5,P6,P7 } 128bit mode {P0,P1,P2,P3,P4,P5,P6,P7, P8,P9,P10,P11,P12,P13,P14,P15} 1'b1: 64bit mode {P7,P6,P5,P4,P3,P2,P1,P0 } 128bit mode {P15,P14,P13,P12,P11,P10,P9,P8 P7,P6,P5,P4,P3,P2,P1,P0	0x0
[14]	RW	VO_WB_DMA_WR_DATA_MODE	dma Y/UV 0: continuous input mode enable 1: non-continuous input mode enable	0x0
[13]	-	-	-	Reserved
[12]	RW	VO_WB_DMA_WR_BLEN	Write channel burst length 0: burst length =8; 0: burst length =16 mode enable	0x0
[11:4]	-	-	-	Reserved
[3:0]	RW	VO_WB_DMA_WR_OT_NUM	[3:0]: Write channel outstanding number, burst length=16: max 4; burst length =8: max 8;	0x0

VO_WB_CH0_ERROR_UNIT Offset Address: 0x444 Total Reset Value: 0x1

bits	Access	Name	Descriptions	Reset
[31:8]	-	-	-	Reserved
[7:0]	RW	VO_WB_ERROR_CHK_UNIT_WR	[7:0] dma write channel error interrupt generate threshold,frame level min=1; max =255.interrupt will generate when error counter equivalent to threshold.	0x1

VO_WB_CH0_ERROR_TH Offset Address: 0x448 Total Reset Value: 0x1

bits	Access	Name	Descriptions	Reset
[31:8]	-	-	-	Reserved
[7:0]	RW	VO_WB_DMA_CH0_ERROR_TH	[7:0] dma write channel error interrupt generate threshold,frame level min=1; max =255.interrupt will generate when error counter equivalent to threshold.	0x1

VO_WB_CH0_INFO_CLR Offset Address: 0x44c Total Reset Value: 0x1

bits	Access	Name	Descriptions	Reset
[31:4]	-	-	-	Reserved
[3:0]	RW	VO_WB_CH0_INFO_CLR	dma_y write channel [0]: frame counter clear, high active [1]: fifo error counter clear, high active [2]: max bandwidth information clear, high active [3]: total bandwidth information clear, high active	0x1

VO_WB_CH0_RST_REQ Offset Address: 0x44c Total Reset Value: 0x1

bits	Access	Name	Descriptions	Reset
[31:1]	-	-	-	Reserved
[0]	RW	VO_WB_CH0_RST_REQ	write reset request, high active	0x0

VO_WB_CH1_DMA_MODE Offset Address: 0x460 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:29]	-	-	-	Reserved
[28]	RW	VO_WB_DMA_WR_ERR_MODE_EN	dma write channel error detection enable	0x0
[27:16]	-	-	-	Reserved
[15]	RW	VO_WB_DMA_WR_OUT_SWAP	dma Y/UV output swap enable 1'b0: 64bit mode {P0,P1,P2,P3,P4,P5,P6,P7 } 128bit mode {P0,P1,P2,P3,P4,P5,P6,P7, P8,P9,P10,P11,P12,P13,P14,P15} 1'b1: 64bit mode {P7,P6,P5,P4,P3,P2,P1,P0 } 128bit mode {P15,P14,P13,P12,P11,P10,P9,P8 P7,P6,P5,P4,P3,P2,P1,P0	0x0
[14]	RW	VO_WB_DMA_WR_DATA_MODE	dma Y/UV 0: continuous input mode enable 1: non-continuous input mode enable	0x0

bits	Access	Name	Descriptions	Reset
[13]	-	-	-	Reserved
[12]	RW	VO_WB_DMA_WR_BLEN	Write channel burst length 0: burst length =8; 0: burst length =16 mode enable	0x0
[11:4]	-	-	-	Reserved
[3:0]	RW	VO_WB_DMA_WR_OT_NUM	[3:0]: Write channel outstanding number, burst length=16: max 4; burst length =8: max 8;	0x0

VO_WB_CH1_ERROR_UNIT Offset Address: 0x464 Total Reset Value: 0x1

bits	Access	Name	Descriptions	Reset
[31:8]	-	-	-	Reserved
[7:0]	RW	VO_WB_ERROR_CHK_UNIT_WR	[7:0] dma write channel error interrupt generate threshold,frame level min=1; max =255.interrupt will generate when error counter equivalent to threshold.	0x1

VO_WB_CH1_ERROR_TH Offset Address: 0x468 Total Reset Value: 0x1

bits	Access	Name	Descriptions	Reset
[31:8]	-	-	-	Reserved
[7:0]	RW	VO_WB_DMA_CH1_ERROR_TH	[7:0] dma write channel error interrupt generate threshold,frame level min=1; max =255.interrupt will generate when error counter equivalent to threshold.	0x1

VO_WB_CH1_INFO_CLR Offset Address: 0x46c Total Reset Value: 0x1

bits	Access	Name	Descriptions	Reset
[31:4]	-	-	-	Reserved
[3:0]	RW	VO_WB_CH1_INFO_CLR	dma_y write channel [0]: frame counter clear, high active [1]: fifo error counter clear, high active [2]: max bandwidth information clear, high active [3]: total bandwidth information clear, high active	0x1

VO_WB_CH1_RST_REQ Offset Address: 0x470 Total Reset Value: 0x1

bits	Access	Name	Descriptions	Reset
[31:1]	-	-	-	Reserved
[0]	RW	VO_WB_CH1_RST_REQ	write reset request, high active	0x0

VO_WB_Y_WR_OUT_IDLE Offset Address: 0x480 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:1]	-	-	-	Reserved
[0]	RW	VO_WB_Y_WR_OUT_IDLE	Write back DMA channel Idle(Y) 1'b0: busy; 1'b1: idle	0x0

VO_WB_UV_WR_OUT_IDLE Offset Address: 0x484 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:1]	-	-	-	Reserved
[0]	RW	VO_WB_UV_WR_OUT_IDLE	Write back DMA channel Idle (UV) 1'b0: busy; 1'b1: idle	0x0

VO_WB_Y_WR_OUT_INT Offset Address: 0x488 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:1]	-	-	-	Reserved
[0]	RW	VO_WB_Y_WR_OUT_INT_ST	[0]: Write back DMA channel Int State (Y)High active	0x0

VO_WB_UV_WR_OUT_INT Offset Address: 0x48c Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:1]	-	-	-	Reserved
[0]	RW	VO_WB_UV_WR_OUT_INT_ST	[0]: Write back DMA channel Int State (UV)High active	0x0

VO_WB_Y_WR_OUT_FUNC_0 Offset Address: 0x490 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:16]	RW	VO_WB_Y_WR_OUT_FUNC_0_ERR_CNT	Write Back DMA channel (UV) Buffer error counter, frame unit.No more than one error will generate per one frame Frame counter	0x0
[15:0]	RW	VO_WB_Y_WR_OUT_FUNC_0_FRM_CNT	Write Back DMA channel (Y) Frame counter	0x0

VO_WB_Y_WR_OUT_FUNC_1 Offset Address: 0x494 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31]	RW	VO_WB_Y_WR_OUT_FUNC_0_DMA_IDLE	DMA_IDLE:state machine dma controller. 1'b0:dma idle , can be reset by software; 1'b1 dma busy	0x0
[30]	RW	VO_WB_Y_WR_OUT_FUNC_0_BASE_SEL	frame base selection: 1'b0: frame base0 ; 1'b1: frame base1	0x0

bits	Access	Name	Descriptions	Reset
[29:26]	-	-	-	Reserved
[25:8]	RW	VO_WB_Y_WR_OUT_FUNC_0_BD_TOTAL	bw_total:total bandwidth form last clear, Mbyte	
unit	0x0			
[7:0]	RW	VO_WB_Y_WR_OUT_FUNC_1_BD_MAX	BW_max: max bandwidth per frame , 128KByte unit	0x0

VO_WB_UV_WR_OUT_FUNC_0 Offset Address: 0x490 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:16]	RW	VO_WB_UV_WR_OUT_FUNC_0_ERR_CNT	Write Back DMA channel (UV) Buffer error counter, frame unit.No more than one error will generate per one frame Frame counter	0x0
[15:0]	RW	VO_WB_UV_WR_OUT_FUNC_0_FRM_CNT	Write Back DMA channel (UV) Frame counter	0x0

VO_WB_UV_WR_OUT_FUNC_1 Offset Address: 0x494 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31]	RW	VO_WB_UV_WR_OUT_FUNC_0_DMA_IDLE	DMA_IDLE:state machine dma controller. 1'b0:dma idle , can be reset by software; 1'b1 dma busy	0x0
[30]	RW	VO_WB_UV_WR_OUT_FUNC_0_BASE_SEL	frame base selection: 1'b0: frame base0 ; 1'b1: frame base1	0x0
[29:26]	-	-	-	Reserved
[25:8]	RW	VO_WB_UV_WR_OUT_FUNC_0_BD_TOTAL	bw_total:total bandwidth form last clear, Mbyte	
unit	0x0			
[7:0]	RW	VO_WB_UV_WR_OUT_FUNC_1_BD_MAX	BW_max: max bandwidth per frame , 128KByte unit	0x0

ctrl_fill_info_area_0_0 Offset Address: 0x600

rectangle and line draw control information(area_0-area_31):2x32=64,(0x0000-0x00FF) 64bit register per control information. ch[N]_area_N_0: 0x0600+0x0000+(N42+0x0000) ch[N]_area_N_1: 0x0600+0x0000+(N42+0x0004)

bits	Access	Name	Descriptions	Reset
[31]	-	-	-	Reserved
[30]	RW	h_line_u_en	vertical line in up boundry draw enable	0x0

bits	Access	Name	Descriptions	Reset
[29]	RW	h_line_b_en	horizontal line in bottom boundry draw enable	0x0
[28:16]	RW	line_start_pos_x	horizontal direction position start,pixel unit	0x0
[15]	RW	v_line_l_en	vertical line in left boundry draw enable	0x0
[14]	RW	v_line_r_en	vertical line in right boundry of full image draw enable	0x0
[13]	RW	line_draw_en	rectangle area/line draw enable	0x0
[12:0]	RW	line_end_pos_x	horizontal direction position end, pixel unit	0x0

ctrl_fill_info_area_0_1 Offset Address: 0x4 Total Reset Value: 0x0

bits	Access	Name	Descriptions	Reset
[31:29]	RW	line_width_h	bit[5:3] of line width , pixel unit	0x0
[28:16]	RW	line_start_pos_y	vertical direction position start, pixel unit	0x0
[15:13]	RW	line_width_l	bit[2:0] of line width , pixel unit	0x0
[12:0]	RW	line_end_pos_y	vertical direction position end, pixel unit	0x0

ctrl_fill_data_area_0_0 Offset Address: 0x100

rectangle and line draw data information(data_0-data_31):1x32=32,(0x0100-0x017F) 32bit register per control information. ch[N]_data_N_0: 0x0600+0x0100+(N*4+0x0000)

bits	Access	Name	Descriptions	Reset
[31:24]	RW	Fill_alpha	Alpha value of mixer between background and lines pixel_remap=pixel_src*alpha+('d256-alpha)*fill_data	0x0
[23:16]	RW	fill_value_y	Value of Y component	0x0
[15:13]	RW	fill_value_cb	Value of Cb component	0x0
[7:0]	RW	fill_value_cr	Value of Cr component	0x0

ctrl_fill_cmd Offset Address: 0x100

bits	Access	Name	Descriptions	Reset
[31]	RW	fill_en_yuv	YUV(BT1120) channel fill enable	0x0
[30:29]	-	-	-	Reserved
[28:16]	RW	image_active_v	Full Image vertical height (actual -1)	0x0
[15]	RW	fill_en_rgb	RGB channel fill enable	0x0
[14:13]	-	-	-	Reserved
[12:0]	RW	image_active_h	Full Image horzonal width(actual -1)	0x0

8.3 MIPI RX

8.3.1 Overview

MIPI RX implements the physical layer for the MIPI D-PHY interface, stacked in a two data lanes and one clock lane configuration. Data Lanes are unidirectional for HS support and ULPS. Clocklane is unidirectional supporting HS and ULPS.

The PHY is used for CSI-2 host applications.

The PHY is targeted for high-speed digital data reception between a host processor and camera interfaces supporting a maximum effective bit rate of 2.5 Gbps (per lane). The assembled two data lanes system enables up to 5 Gbps aggregate communication throughputs, delivering the bandwidth needed for high throughput data reception.

8.3.2 Function description

8.3.2.1 Features

The PHY has the following features:

- configuration clock-27MHz
- Lane operation ranging from 80 Mbps to 2.5 Gbps in forward direction
- Aggregate throughput up to 5 Gbps with two data lanes
- Maximum LP data rate supported of 10Mbps
- PHY-Protocol Interface (PPI) for clock and data lanes
- Low-power Escape modes and Ultra Low Power state
- HS RX Automatic deskew calibration
- HS RX Equalizer up to 3dB
- Optional usage of external reference resistor (200 Ω)
- HS RX automatic offset cancellation
- Intra-PHY Testability
- Inter-PHY Testability
- PHY-2-PHY HS Checker
- PHY-2-PHY Low power loopback
- PHY-2-PHY HS loopback
- Analog Test Bus -Internal ADC for digital voltage sensing
- Power Collapsing
- 1.8 V +10%,-7% analog high voltage supply operation (vph)
- 0.8 V +10%,-7% analog low voltage supply operation (vp)
- 0.8 V +10%,-7% digital supply operation (vdd)

8.3.2.2 Control Mode

Control mode is the default operating mode. After the initialization is completed, the PHY remains in this default mode until a request is made either by the protocol layer, or directly through the sequence of lowpower signals in the lanes in case of RX. While in control mode, the transmitter side sets the LP-11 state in the lines - this is also called the Stop state. The receiver side remains in control mode while receiving LP-11 in the lines. Any request must start from and end in Stop state. Following a request, a lane can leave control mode for either high-speed data transfer mode, Escape mode, or Ultra Low Power state.

8.3.2.3 High-Speed Data Reception

Once the initialization sequence is completed, the PHY remains in control mode, which is the default operating mode, until some request appears. High-speed is one of the possible requests at this point. Highspeed data transfer occurs in bursts. Only during these bursts the lane is in high-speed mode. A high-speed burst must start from and return to a Stop state (control mode). A high-speed burst allows for the transmission of payload data by the data lanes.

Inherent to such data transmission is the existence of a valid DDR clock in the clock lane.

High speed data bursts should start synchronously in all datalanes (one UI difference in HS-SYNC is tolerated), but may end independently of the state of the remaining data lanes due to an unequal amount of received bytes per datalane.

For clock and data lanes the HS-RX receivers are enabled with the reception of the low power transition (LP11>L01) followed by the enable of the termination when the LP01>LP00 transition is received in the respective lines.

To neglect transition effects on clock and data, after these transitions RX waits (using counters counter_for_des_en and hsrxtssettle) for TCLK-SETTLE and THS-SETTLE to expire (while in TCLK-zero and THS-zero state).

Before the payload data there's a leader sequence (HS-SYNC value 1'h1D) that allow for the receiver synchronization. The transmission of such sequence requires the existence of a valid high-speed clock signal in the clock lane

The clock lane will enter the high-speed mode before a high-speed data transmission (TCLK-PRE) begins and it must not leave this state before all the lanes finish their respective high-speed. The operation sequence when leaving the high-speed mode is slightly different for data and clock lanes. For a clock lane, the high-speed transfer always ends with a HS-0 state, this absence of clock transitions will be detected by RX (TCLK-MISS) followed by the disabling of the HS-RX of the clocklane.

As for a data lane, transmission ends with the differential state opposite to the last bit transmitted and then Stop state is observed in the lines (LP11) and the PHY returns to Control Mode

8.3.2.4 Control and Test Modes

The PHY contains a set of control registers that are primarily used for the PHY configuration of its normal operating modes but that are also used for testing either under the scope of normal silicon characterization or debug activities and production testing in the ATE environment. These control registers are accessible through the PHY test and control interface signals (testdin[7:0], testdout[7:0], testen, testclk and testclr). For correct configuration and testability of the PHY, ensure that this test and control interface is accessible through the link controller and/or by any other means (multiplex/demultiplex layer to chip I/Os, JTAG controller, and so on).

This section details the procedure for configuring a test control register in the PHY. The standard procedure is two-folded; first the necessary control register address is programmed and then the related test data is set to the tester's inputs. All PHY related configurations that impact its functional operation need to be done with the PHY in shutdown mode. The correct operation of the PHY may be compromised if the test and control interface is used during active modes to change the PHY operation configuration. It is allowed to use the test and control interface during active modes for debugging purposes and signals observability only. At the first power-up, it is recommended to apply a tester reset pulse (testclr = 1) before any test control register is set.

Each control register address is composed by a 4-bit word (testcode MSBs) and an 8-bit word (testcode LSBs). With an 8-bit test and control interface, to access each control register, it is necessary to perform two write operations to respectively set the control register address 12-bit testcode MSB + testcode LSB.

The test code programming is done using the following steps:

1. For writing the 4-bit testcode MSBs:
 - a. Ensure that testclk and testen is set to low
 - b. Set testen to high
 - c. Set testclk to high
 - d. Place 0x00 in testdin

- e. Set testclk to low (with the falling edge on testclk, the testdin signal content is latched internally)
- f. Set testen to low
- g. Place the 8-bit word corresponding to the testcode MSBs in testdin
- h. Set testclk to high

- 2. For writing the 8-bit testcode LSBs:
 - a. Set testclk to low
 - b. Set testen to high
 - c. Set testclk to high
 - d. Place the 8-bit word test data in testdin
 - e. Set testclk to low (with the falling edge on testclk, the testdin signal content is latched internally)
 - f. Set testen to low

- 3. For writing the data:
 - a. Place the 8-bit word corresponding to the page offset in testdin
 - b. Set testclk to high (test data is programmed internally)

It is only required to write the 4-bit testcode MSBs if they change for the next testcode to be written. If not, they do not need to be written again. This represents executing steps 2 and 3 above.

The steps above for writing the data can be repeated for different test data to be written to the same test code.

Repeat the above procedure to program subsequent test codes. Additionally, a test reset procedure (testclr = 1) is only needed if you wish to reset the PHY configuration to its default values and override any changes made meanwhile.

After a test code is effectively programmed, testdout[7:0] asynchronously outputs relevant data for that specific control register, whether it is pure read-back data or other meaningful signals

8.3.3 Register summary

8.3.4 Register description

Name=dphy2rxtester_DIG_testcode_x_reg,offset=0x0

Bits	Access Mode	Name	Description
7:4	R/W	reserved	"Description: Unused Value After Reset: 0x0"
3:0	R/W	testcode_x	"Description: Extends the testcode's MSB Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_SYS_0,offset=0x1

Bits	Access Mode	Name	Description

Bits	Access Mode	Name	Description
7:6	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
5	R/W	hsfreqrange_ovr_en_rw	"Description: HS frequency range of operation selection override Value After Reset: 0x0"
4:1	R/W	target_state_rw[3:0]	"Description: Target FSM state to which FSM will move when force_state = 1'b1 'b0000: POWERDWN: Power Down 4'b0001: BGPON: Common block enable 4'b0010: CAL_TYPE: Termination Calibration Type 4'b0011: BURNIN_CAL: Termination Calibration without rext 4'b0100: TERMICAL: Termination Calibration with rext 4'b0110: OFFSET_LANE Data Lane Offset Calibration 4'b0111: IDLE: system boot completed 4'b1000: ULP: system in global ULP 4'b1001: DDLTUNNING: DDL tuning Calibration 4'b1010: SKEW_BACKWARD: Backwards Compatibility Skew Compensation all others: reserved Value After Reset: 0x0"
0	R/W	force_state_rw	Description: Force FSM to move to target state Value After Reset: 0x0

Name=dphy2rxtester_DIG_RDWR_RX_SYS_1,offset=0x2

Bits	Access Mode	Name	Description
7	R/W	timebase_ovr_en_rw	"Description: 4us timebase override enable. Value After Reset: 0x0"
6:0	R/W	hsfreqrange_ovr_rw[6:0]	"Description: HS frequency range of operation selection override Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_SYS_2,offset=0x3

Bits	Access Mode	Name	Description
7:0	R/W	timebase_ovr_rw[7:0]	Description: 4us timebase override. Value After Reset: 0x0

Name=dphy2rxtester_DIG_RDWR_RX_SYS_3,offset=0x4

Bits	Access Mode	Name	Description
7:6	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
5	R/W	cmp_polarity_rw	"Description: Comparator polarity Value After Reset: 0x0"

Bits	Access Mode	Name	Description
4	R/W	noext_burnin_res_cal_rw	"Description: Selection between type of calibration (if 1, no REXT will be done) Value After Reset: 0x0"
3	R/W	hsdirect_rw	"Description: Allows to set PHY in HS mode without going through LP states Value After Reset: 0x0"
2	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
1	R/W	thsexit_min_ovr_en_rw	"Description: Ths-exit minimum time override enable Value after Reset: 0x0"
0	R/W	thsexit_min_ovr_rw	Description: Ths-exit minimum time override enable value Value after Reset: 0x0

Name=dphy2rxtester_DIG_RDWR_RX_SYS_4,offset=0x5

Bits	Access Mode	Name	Description
7	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
6	R/W	hsrxtclkmisspd_ovr_en_rw	"Description: Tclk-miss detection override enable. Value After Reset: 0x0"
5	R/W	hsrxtclkmisspd_ovr_rw	"Description: Tclk-miss detection override. Value After Reset: 0x0"
4:3	R/W	tclk_miss_opt_rw[1:0]	"Description: Allows optimization of Tclk-miss detection structure response. ould only be used for lower cfg_clk frequencies and according to databook table. Value After Reset: 0x3"
2:0	R/W	number_of_ready_count_rw[2:0]	"Description: Latency of measurement of analog data for data lane offset al after changing setting Value After Reset: 0x1"

Name=dphy2rxtester_DIG_RDWR_RX_SYS_5,offset=0x6

Bits	Access Mode	Name	Description
7:0	R/W	offset_comp_threshold_rw[7:0]	"Description: 1s 0s threshold to increase/decrease offset voltage setting Value After Reset: 0x3f"

Name=dphy2rxtester_DIG_RDWR_RX_SYS_6,offset=0x7

Bits	Access Mode	Name	Description

Bits	Access Mode	Name	Description
7:0	R/W	tolerance_rw[7:0]	"Description: Tolerance of offset comparator threshold (register 0x76) Value After Reset: 0x2"

Name=dphy2rxtester_DIG_RDWR_RX_SYS_7,offset=0x8

Bits	Access Mode	Name	Description
7	R/W	tclk_miss_div2_ovr_en_rw	"Description: Use divider by 2 on Tclk-miss detection circuit override. Value After Reset: 0x0"
6	R/W	tclk_miss_div2_ovr_rw	"Description: Use divider by 2 on Tclk-miss detection circuit override. Value After Reset: 0x0"
5	R/W	deskew_polarity_rw	"Description: De-skew Calibration pattern control (only used in backwards compatibility) Value After Reset: 0x0"
4:3	R/W	deskew_latency_rw[1:0]	"Description: number of cycles 2'b00: 3 cycles 2'b01: 4 cycles 2'b10: 5 cycles 2'b11: 6 cycles Value After Reset: 0x3"
2	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
1	R/W	skew_mux_sel_rw	"Description: Selects between auto mode and manual selection of de-skew cal mechanism Value After Reset: 0x0"
0	R/W	skew_mux_run_rw	"Description: Selects type of de-skew calibration mode to run (internal or burst based). For 1.5gbps or 2.0gbps. 0 is for backwards compatibility (requires skew_mux_sel = 1'b1) Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_SYS_8,offset=0x9

Bits	Access Mode	Name	Description
7	R/W	tafailsafez_rw	"Description: Enable error contention detection in all LP modes Value After Reset: 0x0"
6:5	R/W	aligner_dk_cnf_rw[1:0]	"Description: aligner config Value After Reset: 0x0"

Bits	Access Mode	Name	Description
4	R/W	noalign_error_rw	"Description: No alignment error bit. Selects how errsotsynchs_behaves (if 1 only flag own lane's alignment, if 0, all errsotsynchs_ behave as a single link) Value After Reset: 0x0"
3:2	R/W	deskew_jump2steps_rw[1:0]	"Description: De-skew Calibration steps control. Jumps 2 steps at a time in the de-skew calibration machine (reduces calibration time but results in higher sampling error) Value After Reset: 0x0"
1	R/W	deskew_overflow_rw	"Description: Feature that forces de-skew algorithm convergence by setting 2nd edge pointer to step 31 (for situations where eventual DDL variation over PVT could cause potential failures). Value After Reset: 0x0"
0	R/W	deskew_numedges_update_rw	"Description: Procedure Set deskew_nemedge_update to 1'b0; Change deskew_numedges to desired value; Set deskew_nemedge_update to 1'b1; Circuitry ensures synchronous update of internal counters Value After Reset: 0x1"

Name=dphy2rxtester_DIG_RDWR_RX_SYS_9,offset=0xa

Bits	Access Mode	Name	Description
7:0	R/W	deskew_numedges_rw[7:0]	"Description: [7:4] De-skew accumulator size (sinusoildal jitter). [3:0] controls the F jump, meaning the number of phases which are jumped after the algorithm finds the first edge (edge1pointer) Value After Reset: 0x40"

Name=dphy2rxtester_DIG_RD_RX_SYS_0,offset=0x1e

Bits	Access Mode	Name	Description
7:6	R	reserved	"Description: Reserved field Value After Reset: 0x0"
5	R	hsdirect_if	"Description: Direct HS indication Value After Reset: 0x0"
4	R	cb_det_supply	"Description: Supply presence detection Value After Reset: 0x0"
3:0	R	state_main[3:0]	"Description: Main FSM state Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_SYS_1,offset=0x1f

Bits	Access Mode	Name	Description
7	R	reserved	"Description: Reserved field Value After Reset: 0x0"
6:0	R	hsfreqrange[6:0]	"Description: HS frequency range selected Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_SYS_2,offset=0x20

Bits	Access Mode	Name	Description
7:0	R	timebase[7:0]	"Description: 4us timebase Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_SYS_3,offset=0x21

Bits	Access Mode	Name	Description
7	R	ulpsactivenot_global	"Description: Global ULPS entry signal (from all lanes) Value After Reset: 0x0"
6	R	txulpsexit_global	"Description: Global ULPS exit signal (from all lanes) Value After Reset: 0x0"
5	R	cb_cal_en_ifpowerctrl	"Description: Output of main FSM signaling termination calibration enable Value After Reset: 0x0"
4	R	lptxenpulldwn_ifpowerctrl	"Description: Main FSM's LP-TX pull down enable Value After Reset: 0x0"
3	R	lptxdin_ifpowerctrl	"Description: Main FSM's LP-TX data in Value After Reset: 0x0"
2	R	lptxena_ifpowerctrl	"Description: Main FSM's LP-TX driver enable Value After Reset: 0x0"
1	R	lptxpon_ifpowerctrl	"Description: Main FSM's LP-TX driver power on Value After Reset: 0x0"
0	R	bintpon_ifpowerctrl	"Description: Main FSM's level shifter power on Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_CONT_DATA_0,offset=0x23

Bits	Access Mode	Name	Description
7	R	reserved	"Description: Reserved field Value After Reset: 0x0"
6:0	R	cont_data[6:0]	"Description: top level cont_data observability Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_SYS_4,offset=0x25

Bits	Access Mode	Name	Description

Bits	Access Mode	Name	Description
7:2	R	reserved	"Description: Reserved field Value After Reset: 0x0"
1:0	R	rx_rxhs_prog[1:0]	"Description: Analog programmability Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_SYS_DDL_0,offset=0x37

Bits	Access Mode	Name	Description
7:0	R/W	rx_ddl_offset_error_rw[7:0]	"Description: bits [7:0]: RX DDL offset error Value After Reset: 0x32"

Name=dphy2rxtester_DIG_RDWR_RX_SYS_DDL_1,offset=0x38

Bits	Access Mode	Name	Description
7:4	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
3:0	R/W	rx_ddl_offset_error_rw[11:8]	"Description: bits [7:0]: RX DDL offset error Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_SYS_DDL_2,offset=0x39

Bits	Access Mode	Name	Description
7:2	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
1:0	R/W	rx_rxhs_ddl_clk_out_div_rw[1:0]	"Description: RX HS DDL output clock divider bits Value After Reset: 0x2"

Name=dphy2rxtester_DIG_RDWR_RX_SYSTIMERS_0,offset=0x4e

Bits	Access Mode	Name	Description
7:0	R/W	timeinitbg_rw[7:0]	"Description: Timer is loaded with value 10'bXXXXXXXXX00 Value After Reset: 0x10"

Name=dphy2rxtester_DIG_RDWR_RX_SYSTIMERS_1,offset=0x4f

Bits	Access Mode	Name	Description
7:2	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
1:0	R/W	timeinitbg_rw[9:8]	"Description: Timer is loaded with value 10'bXXXXXXXXX00 Value After Reset: 0x1"

Name=dphy2rxtester_DIG_RDWR_RX_SYSTIMERS_2,offset=0x50

Bits	Access Mode	Name	Description

Bits	Access Mode	Name	Description
7:6	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
5:0	R/W	ttago_rw[5:0]	"Description: Number of cycles that the TX drives the bridge state during a turnaround procedure Value After Reset: 0x3"

Name=dphy2rxtester_DIG_RDWR_RX_SYSTIMERS_3,offset=0x51

Bits	Access Mode	Name	Description
7:6	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
5:0	R/W	ttasure_rw[5:0]	"Description: Number of cycles that the RX waits after a bridge state has been detected during a turnaround procedure Value After Reset: 0x1"

Name=dphy2rxtester_DIG_RDWR_RX_SYSTIMERS_4,offset=0x52

Bits	Access Mode	Name	Description
7:6	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
5:0	R/W	ttawait_rw[5:0]	"Description: Number of cycles that the PHY waits for until a turnaround request is processed Value After Reset: 0x32"

Name=dphy2rxtester_DIG_RDWR_RX_SYSTIMERS_5,offset=0x53

Bits	Access Mode	Name	Description
7:0	R/W	twakeupulpm_rw[7:0]	"Description: Timer is loaded with value 10'bXXXXXXXXX11 Value After Reset: 0x44"

Name=dphy2rxtester_DIG_RDWR_RX_SYSTIMERS_6,offset=0x54

Bits	Access Mode	Name	Description
7:2	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
1:0	R/W	twakeupulpm_rw[9:8]	"Description: Timer is loaded with value 10'bXXXXXXXXX11 Value After Reset: 0x1"

Name=dphy2rxtester_DIG_RDWR_RX_SYSTIMERS_7,offset=0x55

Bits	Access Mode	Name	Description
7:0	R/W	txdindlytime_rw[7:0]	"Description: Timer is loaded with value 10'bXXXXXXXXX00 Value After Reset: 0x1c"

Name=dphy2rxtester_DIG_RDWR_RX_SYSTIMERS_8,offset=0x56

Bits	Access Mode	Name	Description
7:2	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
1:0	R/W	txdindlytime_rw[9:8]	"Description: Timer is loaded with value 10'bXXXXXXXXX00 Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_SYSTIMERS_9,offset=0x57

Bits	Access Mode	Name	Description
7:0	R/W	txstarttime_rw[7:0]	"Description: Timer is loaded with value 10'bXXXXXXXXX00 Value After Reset: 0x10"

Name=dphy2rxtester_DIG_RDWR_RX_SYSTIMERS_10,offset=0x58

Bits	Access Mode	Name	Description
7:2	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
1:0	R/W	txstarttime_rw[9:8]	"Description: Timer is loaded with value 10'bXXXXXXXXX00 Value After Reset: 0x1"

Name=dphy2rxtester_DIG_RDWR_RX_SYSTIMERS_11,offset=0x59

Bits	Access Mode	Name	Description
7:0	R/W	rxpontime_rw[7:0]	"Description: Timer used in offset calibration for state machine power up. Value After Reset: 0x5"

Name=dphy2rxtester_DIG_RD_RX_SYSTIMERS_0,offset=0x7a

Bits	Access Mode	Name	Description
7	R	reserved	"Description: Reserved field Value After Reset: 0x0"
6:0	R	hsrxthssettle_datalanes[6:0]	"Description: HS RX THSSettle Timing Counter Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_RX_STARTUP_OBS_0,offset=0xc7

Bits	Access Mode	Name	Description
7	R	hsrxpon_clklane_ifpowerctrl	"Description: Output of main FSM enabling clock lane HS receiver analog circuitry Value After Reset: 0x0"

Bits	Access Mode	Name	Description
6	R	hsrxencal_lane_offset_ifpowerctrl	"Description: Output of main FSM enabling digital data lane offset calibration machine Value After Reset: 0x0"
5	R	clk_en_lanes	"Description: Clock enable lanes Value After Reset: 0x0"
4	R	"des_en_for_data lane_offset_cal_ifpowerctrl"	"Description: Output of main FSM enabling feed of clock lane receivers with internal oscillator signal Value After Reset: 0x0"
3	R	rxlane_feed_int_clk_ifpowerctrl	"Description: Output of main FSM enabling feed of data lane receivers with internal oscillator signal Value After Reset: 0x0"
2	R	rxclk_rxhs_feed_int_clk_if	"Description: Observes the selection between internal clock source or external (receiver) override mux output Value After Reset: 0x0"
1	R	rxclk_feed_int_clk_ifpowerctrl	"Description: Output of main FSM enabling feed of clock lane receivers with internal oscillator signal Value After Reset: 0x0"
0	R	burnin_cal_en_if	"Description: Burnin oscillator enable signal Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_RX_STARTUP_OBS_1,offset=0xc8

Bits	Access Mode	Name	Description
7	R	skew_back_comp_en	"Description: de-skew backwards compatibility observation Value After Reset: 0x0"
6	R	skew_back_comp_en_ifpowerctrl	"Description: Output of main FSM signalling backwards compatibility de-skew compensation enable Value After Reset: 0x0"
5	R	perform_skew	"Description: Internal signaling to distinguish de-skew calibration through internal oscillator or external burst training sequence (1 if compatibility mode) Value After Reset: 0x0"
4	R	rx_rxhs_compatibility_mode	"Description: compatibility mode observation Value After Reset: 0x0"

Bits	Access Mode	Name	Description
3	R	global_ddltunning_done	"Description: Flag indicating that the DDL calibrations for all the active lanes are finished Value After Reset: 0x0"
2	R	ddltunning_en_ifpowerctrl	"Description: Output of main FSM enabling DDL calibration Value After Reset: 0x0"
1	R	global_offset_cal_done	"Description: Flag indicating that the offset calibration for all the active lanes are finished Value After Reset: 0x0"
0	R	hsrxpon_datalane_ifpowerctrl	"Description: Output of main FSM enabling data lane HS receiver analog circuitry Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_RX_STARTUP_OBS_2,offset=0xc9

Bits	Access Mode	Name	Description
7:6	R	reserved	"Description: Reserved field Value After Reset: 0x0"
5	R	deskew_legacyz	"Description: De-skew algorithm currently enabled (1'b0 for backwards compatibility) Value After Reset: 0x0"
4	R	enable_lanes	"Description: Lanes enable observation Value After Reset: 0x0"
3:2	R	rx_rxhs_gmode[1:0]	"Description: RX HS gmode observation Value After Reset: 0x0"
1	R	align_error	"Description: No alignment error bit. Selects how errsotsynchs_behaves (if 1 only flag own lane's alignment, if 0, all errsotsynchs_ behave as a single link) Value After Reset: 0x0"
0	R	global_skew_cal_done	"Description: Flag indicating that the de-skew calibrations for all the active lanes are finished Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_RX_STARTUP_OBS_3,offset=0xca

Bits	Access Mode	Name	Description
7:4	R	reserved	"Description: Reserved field Value After Reset: 0x0"

Bits	Access Mode	Name	Description
3:0	R	counter_for_des_en_if[3:0]	"Description: Observation of the counter needed to meet GTP (min and max clocklane settle time) Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_RX_STARTUP_OBS_4,offset=0xcb

Bits	Access Mode	Name	Description
7	R	reserved	"Description: Reserved field Value After Reset: 0x0"
6:0	R	hsfrequent_top[6:0]	"Description: hsfrequent observation after mux override Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_RX_STARTUP_OVR_0,offset=0xe0

Bits	Access Mode	Name	Description
7	R/W	"rx_rxhs_compatibility_mode_ovr_en_rw"	"Description: compatibility mode override enable Value After Reset: 0x0"
6	R/W	bypass_skew_machine_rw	"Description: Bypass of de-skew machine Value After Reset: 0x0"
5	R/W	skew_back_comp_en_ovr_rw	"Description: de-skew backwards compatibility override enable value Value After Reset: 0x0"
4	R/W	skew_back_comp_en_ovr_en_rw	"Description: de-skew back comparator override enable control Value After Reset: 0x0"
3	R/W	bypass_ddltuning_machine_rw	"Description: Bypass DDL tuning machine Value After Reset: 0x0"
2	R/W	bypass_offset_machine_rw	"Description: Bypass offset machine bit Value After Reset: 0x0"
1	R/W	clk_en_lanes_tester_rw	"Description: Clock enable lanes for tester Value After Reset: 0x0"
0	R/W	clk_en_lanes_bypass_rw	"Description: Clock enable lanes bypass Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_RX_STARTUP_OVR_1,offset=0xe1

Bits	Access Mode	Name	Description
7:6	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"

Bits	Access Mode	Name	Description
5	R/W	deskew_legacyz_ovr_en_rw	"Description: De-skew algorithm override enable. Value After Reset: 0x0"
4	R/W	deskew_legacyz_ovr_rw	"Description: De-skew algorithm override (requires deskew_legacyz_ovr_en = 1'b1). Value After Reset: 0x0"
3	R/W	enable_lanes_ovr_rw	"Description: Lanes enable override (requires enable_lanes_ovr_e = 1'b1) Value After Reset: 0x0"
2	R/W	enable_lanes_ovr_en_rw	"Description: Lanes enable override enable Value After Reset: 0x0"
1	R/W	"rxclk_rxhs_ddr_clk_en_bypass_r w"	"Description: HS RX DDR clock enable bypass control Value After Reset: 0x0"
0	R/W	"rx_rxhs_compatibility_mode_ovr_r w"	"Description: compatibility mode override value (needs rx_rxhs_compatibility_mode_ovr_en = 1'b1) Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_RX_STARTUP_OVR_2, Offset = 0xe2

Bits	Access Mode	Name	Description
7:0	R/W	ddl_osc_freq_target_ovr_rw[7:0]	"Description: DDL oscillation frequency override value (Main) Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_RX_STARTUP_OVR_3,offset=0xe3

Bits	Access Mode	Name	Description
7:4	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
3:0	R/W	ddl_osc_freq_target_ovr_rw[11:8]	"Description: DDL oscillation frequency override value (Main) Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_RX_STARTUP_OVR_4,offset=0xe4

Bits	Access Mode	Name	Description
7:4	R/W	"counter_for_des_en_config_if_rw[3:0]"	"Description: Override value of the counter used to meet the min and max clocklane settle time Value After Reset: 0x0"

Bits	Access Mode	Name	Description
3:2	R/W	rx_rxhs_gmode_if_ovr_rw[1:0]	"Description: RX HS gmode override value Value After Reset: 0x0"
1	R/W	rx_rxhs_gmode_if_ovr_en_rw	"Description: RX HS gmode override enable control Value After Reset: 0x0"
0	R/W	ddl_osc_freq_target_ovr_en_rw	"Description: DDL oscillation frequency override enable Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_RX_STARTUP_OVR_5,offset=0xe5

Bits	Access Mode	Name	Description
7:1	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
0	R/W	counter_for_des_en_bypass_rw	"Description: Override enable of the counter used to meet the min and max clocklane settle time Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_RX_STARTUP_OVR_17,offset=0xf1

Bits	Access Mode	Name	Description
7	R/W	"hsrxthssettle_ovr_en_datalanes_rw"	"Description: HS_SETTLE timer override enable for datalanes Value After Reset: 0x0"
6:0	R/W	"hsrxthssettle_datalanes_ovr_rw[6:0]"	"Description: HS_SETTLE timer override value for datalanes Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_BIST_0,offset=0x107

Bits	Access Mode	Name	Description
7:4	R/W	pg_mode_r_rw[3:0]	"Description: Pattern Generator modes selection Value After Reset: 0x7"
3:0	R/W	Reserved	"Description: Reserved Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_BIST_1,offset=0x108

Bits	Access Mode	Name	Description
7:0	R/W	pg_userst_pat_rw[7:0]	"Description: Pattern Generator User Defined Data Control Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_BIST_2,offset=0x109

Bits	Access Mode	Name	Description
7:0	R/W	Reserved	"Description: Reserved Value After Reset: 0xaa"

Name=dphy2rxtester_DIG_RDWR_RX_BIST_OVR_3,offset=0x113

Bits	Access Mode	Name	Description
7	R/W	rx_rxhs_prog_ovr_en_rw	"Description: AFE internal clock routing attachment to receiver override enable. Value After Reset: 0x0"
6:5	R/W	rx_rxhs_prog_ovr_rw[1:0]	"Description: bit [1] is unused and should be considered reserved. Value After Reset: 0x0"
4	R/W	rxclk_lb_sel_ovr_en_rw	"Description: Analog loopback multiplexer override enable. Value After Reset: 0x0"
3	R/W	rxclk_lb_sel_ovr_rw	"Description: Analog loopback multiplexer override. Used to force AFE to recover data from internal loopback driver. Value After Reset: 0x0"
2	R/W	Reserved	"Description: Reserved Value After Reset: 0x0"
1	R/W	Reserved	"Description: Reserved Value After Reset: 0x0"
0	R/W	del_datalane_ovr_en_rw	"Description: Serializer timer override enable. Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_BIST_PIPE_TIMER,offset=0x114

Bits	Access Mode	Name	Description
7:0	R/W	pipe_clean_time_rw[7:0]	"Description: Timer set in pulses of cfg_clk. Value After Reset: 0x80"

Name=dphy2rxtester_DIG_RD_RX_BIST_0,offset=0x11e

Bits	Access Mode	Name	Description
7:5	R	reserved	"Description: Reserved field Value After Reset: 0x0"
4	R	rx1_error_latch_if	"Description: Lane 1 error indication Value After Reset: 0x0"
3	R	rx0_error_latch_if	"Description: Lane 0 error indication Value After Reset: 0x0"
2	R	pg_sel	"Description: Indication that Pattern Checker is enabled Value After Reset: 0x0"
1	R	Reserved	"Description: Reserved Value After Reset: 0x0"

Bits	Access Mode	Name	Description
0	R	Reserved	"Description: Reserved Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_RX_DUAL_PHY_0,offset=0x133

Bits	Access Mode	Name	Description
7:1	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
0	R/W	mastermacro_rw	"Description: PHY which has its own clock lane enabled should set this field to 1'b1 (master PHY). The other should set it to 1'b0 (slave PHY). Value After Reset: 0x1"

Name=dphy2rxtester_DIG_RD_RX_DUAL_PHY_0,offset=0x148

Bits	Access Mode	Name	Description
7	R	rxclk_rxhs_rx_en_in	"Description: Input rxclk_rxhs_rx_en_in of slave PHY should be connected to output rxclk_rxhs_rx_en_out of master PHY. Input rxclk_rxhs_rx_en_in of master PHY must be tied to 1'b0. Value After Reset: 0x0"
6	R	rxclk_rxhs_rx_en_out	"Description: Output rxclk_rxhs_rx_en_out of master PHY should be connected to input rxclk_rxhs_rx_en_out of master PHY. Output rxclk_rxhs_rx_en_out of slave PHY should be left floating. Value After Reset: 0x0"
5	R	rxclk_rxhs_rx_pon_in	"Description: Input rxclk_rxhs_rx_pon_in of slave PHY should be connected to output rxclk_rxhs_rx_pon_out of master PHY. Input rxclk_rxhs_rx_pon_in of master PHY must be tied to 1'b0. Value After Reset: 0x0"
4	R	rxclk_rxhs_rx_pon_out	"Description: Output rxclk_rxhs_rx_pon_out of master PHY should be connected to input rxclk_rxhs_rx_pon_out of master PHY. Output rxclk_rxhs_rx_pon_out of slave PHY should be left floating. Value After Reset: 0x0"
3	R	rxclk_rxhs_rx_enlogic_in	"Description: Input rxclk_rxhs_rx_enlogic_in of slave PHY should be connected to output rxclk_rxhs_rx_enlogic_out of master PHY. Input rxclk_rxhs_rx_enlogic_in of master PHY must be tied to 1'b0. Value After Reset: 0x0"

Bits	Access Mode	Name	Description
2	R	rxclk_rxhs_rx_enlogic_out	"Description: Output rxclk_rxhs_rx_enlogic_out of master PHY should be connected to input rxclk_rxhs_rx_enlogic_out of master PHY. Output rxclk_rxhs_rx_enlogic_out of slave PHY should be left floating. Value After Reset: 0x0"
1	R	rxclk_rxhs_clk_en_in	"Description: Input rxclk_rxhs_clk_en_in of slave PHY should be connected to output rxclk_rxhs_clk_en_out of master PHY. Input rxclk_rxhs_clk_en_in of master PHY must be tied to 1'b0. Value After Reset: 0x0"
0	R	rxclk_rxhs_clk_en_out	"Description: Output rxclk_rxhs_clk_en_out of master PHY should be connected to input rxclk_rxhs_clk_en_out of master PHY. Output rxclk_rxhs_clk_en_out of slave PHY should be left floating. Value After Reset: 0x0"
Bits	Access Mode	Name	Description

Name=dphy2rxtester_DIG_RD_RX_DUAL_PHY_1,offset=0x149

Bits	Access Mode	Name	Description
7	R	reserved	"Description: Reserved field Value After Reset: 0x0"
6	R	rxclk_rxhs_feed_int_clk_in	"Description: Input rxclk_rxhs_feed_int_clk_in of slave PHY should be connected to output rxclk_rxhs_feed_int_clk_out of master PHY. Input rxclk_rxhs_feed_int_clk_in of master PHY must be tied to 1'b0. Value After Reset: 0x0"
5	R	rxclk_rxhs_feed_int_clk_out	"Description: Output rxclk_rxhs_feed_int_clk_out of master PHY should be connected to input rxclk_rxhs_feed_int_clk_out of master PHY. Output rxclk_rxhs_feed_int_clk_out of slave PHY should be left floating. Value After Reset: 0x0"
4:3	R	rx_rxhs_prog_in[1:0]	"Description: Input rx_rxhs_prog_in of slave PHY should be connected to output rx_rxhs_prog_out of master PHY. Input rx_rxhs_prog_in of master PHY must be tied to 1'b0. Value After Reset: 0x0"
2:1	R	rx_rxhs_prog_out[1:0]	"Description: Output rx_rxhs_prog_out of master PHY should be connected to input rx_rxhs_prog_out of master PHY. Output rx_rxhs_prog_out of slave PHY should be left floating. Value After Reset: 0x0"

Bits	Access Mode	Name	Description
0	R	ulpsactivenotclk_in	"Description: Input ulpsactivenotclk_in of slave PHY should be connected to output ulpsactivenotclk of master PHY. Input ulpsactivenotclk_in of master PHY must be tied to 1'b0. Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_DUAL_PHY_2,offset=0x14a

Bits	Access Mode	Name	Description
7:2	R	reserved	"Description: Reserved field Value After Reset: 0x0"
1	R	launch_des_en_in	"Description: Input launch_des_en_in of slave PHY should be connected to output launch_des_en_out of master PHY. Input launch_des_en_in of master PHY must be tied to 1'b0. Value After Reset: 0x0"
0	R	launch_des_en_out	"Description: Output launch_des_en_out of master PHY should be connected to input launch_des_en_out of master PHY. Output launch_des_en_out of slave PHY should be left floating. Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_CB_0,offset=0x1aa

Bits	Access Mode	Name	Description
7	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
6:5	R/W	cb_sel_vrefcd_lprx_rw[1:0]	"Description: LPTX voltage reference selection: 0 : 1100mV 1 : 1200mV Value After Reset: 0x1"
4:2	R/W	cb_sel_v400_prog_rw[2:0]	"Description: Select value of cb_v400 1XX : 400mV 011 : 450mV 010 : 350mV 001 : 300mV 000 : 200mV Value After Reset: 0x4"
1	R/W	cb_sel_chop_clk_rw	"Description: Select bandgap clock source: 0 : Internal oscillator (internal clk source) 1 : External clock signal (cb_chop_clk) Value After Reset: 0x1"

Bits	Access Mode	Name	Description
0	R/W	cb_chop_clk_en_rw	"Description: Bandgap chop clock enable 0 : Chop clock gated 1 : Chop clock switching Value After Reset: 0x1"

Name=dphy2rxtester_DIG_RDWR_RX_CB_1,offset=0x1ab

Bits	Access Mode	Name	Description
7	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
6	R/W	cb_atb_vbe_sel_rw	"Description: VBe selection to Analog Test Bus Value After Reset: 0x0"
5:3	R/W	txlp_prog_rw[2:0]	"Description: LP-TX programmability Value After Reset: 0x0"
2	R/W	cb_sel_vreflptx_prog_rw	"Description: LPRX contention detector voltage reference selection: 00 : 275mV 01 : 300mV 10 : 325mV 11: 350mV Value After Reset: 0x1"
1:0	R/W	cb_sel_vref_lprx_rw[1:0]	"Description: LPRX voltage reference selection: 00 : 600mV 01 : 625mV 10 : 650mV 11 : 675mV Value After Reset: 0x1"

Name=dphy2rxtester_DIG_RDWR_RX_CB_2,offset=0x1ac

Bits	Access Mode	Name	Description
7	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
6	R/W	rx_rxlp_bias_prog_rw	"Description: LP-RX biasing control Value After Reset: 0x0"
5:4	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
3	R/W	rx_cb_pre_iref_on_rw	"Description: common block reference current control Value After Reset: 0x1"
2	R/W	cb_atb_ext_con_rw	"Description: allows direct connecting ATB line into ATB pin: 0 : disable 1 : enable Value After Reset: 0x0"

Bits	Access Mode	Name	Description
1:0	R/W	cb_atb_sel_rw[1:0]	"Description: Analog Test Bus selection: cb_atb_sel[1]: 0 : ATB LINE (atb_line) 1 : External resistors cb_atb_sel[0]: 0 : DAC 1 : TX/RX replica Value After Reset: 0x3"

Name=dphy2rxtester_DIG_RDWR_RX_CB_3,offset=0x1ad

Bits	Access Mode	Name	Description
7	R/W	cb_lpibias_en_ovr_en_rw	"Description: Override Enable voltage control to LPTX and currents for LPTX, LPRX Value After Reset: 0x0"
6	R/W	cb_bg_en_ovr_rw	"Description: Bandgap enable override Value After Reset: 0x0"
5	R/W	cb_bg_en_ovr_en_rw	"Description: Bandgap enable override Value After Reset: 0x0"
4	R/W	cb_atb_comp_clk_ovr_rw	"Description: Clock used in analog comparator override Value After Reset: 0x0"
3	R/W	cb_atb_comp_clk_ovr_en_rw	"Description: Clock used in analog comparator override Value After Reset: 0x0"
2:0	R/W	Reserved	"Description: Reserved Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_CB_4,offset=0x1ae

Bits	Access Mode	Name	Description
7	R/W	cb_atb_on_ovr_en_rw	"Description: Analog Test Bus enable override Value After Reset: 0x0"
6:3	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
2	R/W	cb_on_ovr_rw	"Description: Common Block power on override control Value After Reset: 0x0"
1	R/W	cb_on_ovr_en_rw	"Description: Common Block power on override control Value After Reset: 0x0"
0	R/W	cb_lpibias_en_ovr_rw	"Description: Override Enable voltage control to LPTX and currents for LPTX, LPRX Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_CB_5,offset=0x1af

Bits	Access Mode	Name	Description

Bits	Access Mode	Name	Description
7:1	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
0	R/W	cb_atb_on_ovr_rw	"Description: Analog Test Bus enable override Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_CB_6,offset=0x1b0

Bits	Access Mode	Name	Description
7:5	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
4	R/W	rx_cb_lb_en_rext_if_rw	"Description: Static termination driver test, REXT enable Value After Reset: 0x0"
3:1	R/W	rx_cb_lb_setrx_if_rw[2:0]	"Description: comparator threshold (resistance mismatch to REXT) : 3'b111: -20% : 3'b011: -15% : 3'b101: -10% : 3'b001: -5% : 3'b110: 5% : 3'b010: 10% : 3'b100: 15% : 3'b000: 20% Value After Reset: 0x0"
0	R/W	rx_cb_ibias_en_if_rw	"Description: Enable currents for LP-TX, LP-RX and HS-RX Value After Reset: 0x1"

Name=dphy2rxtester_DIG_RD_RX_ANA_CB_0,offset=0x1c4

Bits	Access Mode	Name	Description
7:6	R	reserved	"Description: Reserved field Value After Reset: 0x0"
5	R	cb_on	"Description: Common block level shifter enable Value After Reset: 0x0"
4	R	cb_lpi bias_en	"Description: Enable voltage currents for LP-TX and LP-RX Value After Reset: 0x0"
3	R	cb_ical_x4	"Description: Programmable current for external resistor calibration Value After Reset: 0x0"
2	R	cb_bg_en	"Description: Bandgap enable Value After Reset: 0x0"
1	R	cb_atb_on	"Description: Analog test bus enable Value After Reset: 0x0"

Bits	Access Mode	Name	Description
0	R	cb_atb_comp_clk	"Description: Analog test bus comparator clock Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_DAC_0,offset=0x1da

Bits	Access Mode	Name	Description
7:2	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
1	R/W	adc_done_clear_rw	"Description: Clears last ADC procedure status Value After Reset: 0x0"
0	R/W	adc_en_rw	"Description: ADC enable, triggers ADC conversion Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_DAC_1,offset=0x1db

Bits	Access Mode	Name	Description
7:0	R/W	cb_dac_prog_ovr_rw[7:0]	"Description: Digital word to be translated to analog voltage override Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_DAC_2,offset=0x1dc

Bits	Access Mode	Name	Description
7:2	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
1:0	R/W	cb_dac_prog_ovr_rw[9:8]	"Description: Digital word to be translated to analog voltage override Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_DAC_3,offset=0x1dd

Bits	Access Mode	Name	Description
7:1	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
0	R/W	cb_dac_prog_ovr_en_rw	"Description: Digital word to be translated to analog voltage override enable Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_DAC_0,offset=0x1f2

Bits	Access Mode	Name	Description
7:1	R	reserved	"Description: Reserved field Value After Reset: 0x0"
0	R	adc_done	"Description: Done flag for observation in ADC machine Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_DAC_1,offset=0x1f3

Bits	Access Mode	Name	Description
7:0	R	cb_dac_prog[7:0]	"Description: ADC digital word representing analog voltage SAR algorithm result Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_DAC_2,offset=0x1f4

Bits	Access Mode	Name	Description
7:2	R	reserved	"Description: Reserved field Value After Reset: 0x0"
1:0	R	cb_dac_prog[9:8]	"Description: ADC digital word representing analog voltage SAR algorithm result Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_TERM_CAL_0,offset=0x209

Bits	Access Mode	Name	Description
7:4	R/W	cb_cal_repl_ovr_rw[3:0]	"Description: Calibration word for termination lower section override Value After Reset: 0x0"
3:2	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
1	R/W	cb_cal_en_ovr_rw	"Description: Allow to override the lower part termination control Value After Reset: 0x0"
0	R/W	cb_cal_en_ovr_en_rw	"Description: Allow to override the lower part termination control Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_TERM_CAL_1,offset=0x20a

Bits	Access Mode	Name	Description
7	R/W	cb_en_45_ohm_rw	"Description: Selection for 45ohm termination impedance Value After Reset: 0x0"
6:1	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
0	R/W	cb_cal_repl_ovr_en_rw	"Description: Calibration word for termination lower section replica override enable Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_TERM_CAL_2,offset=0x20b

Bits	Access Mode	Name	Description

Bits	Access Mode	Name	Description
7:2	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
1	R/W	cb_ical_x4_ovr_rw	"Description: Programming current for termination calibration algorithm override Value After Reset: 0x0"
0	R/W	cb_ical_x4_ovr_en_rw	"Description: Programming current for termination calibration algorithm override Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_TERM_CAL_0,offset=0x220

Bits	Access Mode	Name	Description
7:6	R	reserved	"Description: Reserved field Value After Reset: 0x0"
5:2	R	cb_cal_repl[3:0]	"Description: Termination lower section replica setting Value After Reset: 0x0"
1	R	reserved	"Description: Reserved field Value After Reset: 0x0"
0	R	cb_cal_en	"Description: Termination lower section analog circuitry enable Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_TERM_CAL_1,offset=0x221

Bits	Access Mode	Name	Description
7	R	rescal_done	"Description: Lower section termination calibration algorithm done Value After Reset: 0x0"
6	R	reserved	"Description: Reserved field Value After Reset: 0x0"
5	R	rescal_en	"Description: Lower section termination calibration algorithm enable Value After Reset: 0x0"
4	R	cb_comp_out	"Description: Analog comparator output Value After Reset: 0x0"
3:0	R	reserved	"Description: Reserved field Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_TERM_CAL_2,offset=0x222

Bits	Access Mode	Name	Description
7:1	R	reserved	"Description: Reserved field Value After Reset: 0x0"

Bits	Access Mode	Name	Description
0	R	rescal_error	"Description: Termination lower section calibration algorithm error This bit corresponds to a multi-toggle flag which is not considered an error Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_NOEXTTUNE_0,offset=0x237

Bits	Access Mode	Name	Description
7:0	R/W	count_ref_rw[7:0]	"Description: Termination calibration without REXT decision threshold (to increase/decrease resistance) Value After Reset: 0x96"

Name=dphy2rxtester_DIG_RDWR_RX_NOEXTTUNE_1,offset=0x238

Bits	Access Mode	Name	Description
7:2	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
1:0	R/W	count_ref_rw[9:8]	"Description: Termination calibration without REXT decision threshold (to increase/decrease resistance) Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_NOEXTTUNE_2,offset=0x239

Bits	Access Mode	Name	Description
7:4	R/W	setr_min_rw[3:0]	"Description: Termination calibration without REXT minimum oscillation threshold Value After Reset: 0x4"
3:0	R/W	setr_max_rw[3:0]	"Description: Termination calibration without REXT maximum oscillation threshold Value After Reset: 0xb"

Name=dphy2rxtester_DIG_RDWR_RX_NOEXTTUNE_3,offset=0x23a

Bits	Access Mode	Name	Description
7:4	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
3:0	R/W	setr_typ_rw[3:0]	"Description: Termination calibration without REXT typical oscillation threshold Value After Reset: 0x7"

Name=dphy2rxtester_DIG_RDWR_RX_NOEXTTUNE_4,offset=0x23b

Bits	Access Mode	Name	Description

Bits	Access Mode	Name	Description
7:0	R/W	stability_time_rw[7:0]	"Description: Additional stability time for internal oscillator used for termination calibration without REXT Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_NOEXTTUNE_5,offset=0x23c

Bits	Access Mode	Name	Description
7:5	R/W	rx_rxhs_setr_ovr_rw[2:0]	"Description: Termination calibration word override (needs rx_rxhs_setr_ovr_en = 1'b1) Value After Reset: 0x0"
4	R/W	rx_rxhs_setr_ovr_en_rw	"Description: Termination calibration word override enable Value After Reset: 0x0"
3	R/W	rxclk_rxhs_osc_res_en_ovr_rw	"Description: Oscillator mode (poly/MOS) to analog circuitry override Value After Reset: 0x0"
2	R/W	"rxclk_rxhs_osc_res_en_ovr_en_rw"	"Description: Oscillator mode (poly/MOS) to analog circuitry override enable Value After Reset: 0x0"
1	R/W	rxclk_rxhs_osc_pon_ovr_rw	"Description: Oscillator power-on override Value After Reset: 0x0"
0	R/W	rxclk_rxhs_osc_pon_ovr_en_rw	"Description: Oscillator power-on override enable Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_NOEXTTUNE_6,offset=0x23d

Bits	Access Mode	Name	Description
7:4	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
3	R/W	burnin_cal_en_ovr_rw	"Description: Burnin oscillator setting override value (needs burnin_cal_en_ovr_en = 1'b1) Value After Reset: 0x0"
2	R/W	burnin_cal_en_ovr_en_rw	"Description: Burnin oscillator setting override value (needs burnin_cal_en_ovr_en = 1'b1) Value After Reset: 0x0"
1	R/W	norexttune_done_ovr_rw	"Description: Termination calibration (without external reference resistor) override control Value After Reset: 0x0"

Bits	Access Mode	Name	Description
0	R/W	norexttune_done_ovr_en_rw	"Description: Termination calibration (without external reference resistor) override control Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_NOEXTTUNE_0,offset=0x252

Bits	Access Mode	Name	Description
7:0	R	count_osc0[7:0]	"Description: Counter Oscillator Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_NOEXTTUNE_1,offset=0x253

Bits	Access Mode	Name	Description
7:2	R	reserved	"Description: Reserved field Value After Reset: 0x0"
1:0	R	count_osc0[9:8]	"Description: Counter Oscillator Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_NOEXTTUNE_2,offset=0x254

Bits	Access Mode	Name	Description
7:0	R	count_osc1[7:0]	"Description: Counter oscillator Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_NOEXTTUNE_3,offset=0x255

Bits	Access Mode	Name	Description
7:6	R	reserved	"Description: Reserved field Value After Reset: 0x0"
5:2	R	setr_burnin[3:0]	"Description: NOEXT calibration result word Value After Reset: 0x0"
1:0	R	count_osc1[9:8]	"Description: Counter oscillator Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_NOEXTTUNE_4,offset=0x256

Bits	Access Mode	Name	Description
7	R	reserved	"Description: Reserved field Value After Reset: 0x0"
6	R	noext_burn_error	"Description: Termination calibration without external resistor error Value After Reset: 0x0"

Bits	Access Mode	Name	Description
5	R	norexttune_done	"Description: Termination calibration without external resistor done Value After Reset: 0x0"
4:2	R	rx_rxhs_setr[2:0]	"Description: Termination setting Value After Reset: 0x0"
1	R	rxclk_rxhs_osc_res_en	"Description: Burn-in oscillator mode enable Value After Reset: 0x0"
0	R	rxclk_rxhs_osc_pon	"Description: Burn-in oscillator power on Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_CLKLANE_LANE_0,offset=0x301

Bits	Access Mode	Name	Description
7	R/W	hsrx_en_bypass_clklane_rw	"Description: Clock lane HS receiver override enable Value After Reset: 0x0"
6	R/W	bintpon_clklane_rw	"Description: Bias block power-on; can be overridden if bypass signal is 1'b1 Value After Reset: 0x0"
5	R/W	bintpon_bypass_clklane_rw	"Description: Bias block power-on bypass override Value After Reset: 0x0"
4:1	R/W	atb_sel_clklane_rw[3:0]	"Description: Analog Test Bus signals selection: 000: disconnected 001: vp 010: vpclk 011: gd 100: Voltage steering termination switches Value After Reset: 0x0"
0	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_CLKLANE_LANE_1,offset=0x302

Bits	Access Mode	Name	Description
7:5	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
4	R/W	hsrxpon_if_clklane_rw	"Description: Power on of RX HS Value After Reset: 0x0"
3	R/W	hsrxpon_bypass_clklane_rw	"Description: Power on of RX HS bypass bit Value After Reset: 0x0"
2	R/W	hsrxenterm_if_clklane_rw	"Description: RX HS enable term bit Value After Reset: 0x0"
1	R/W	hsrxenterm_bypass_clklane_rw	"Description: RX HS enable term bypass bit Value After Reset: 0x0"

Bits	Access Mode	Name	Description
0	R/W	hsrx_en_if_clklane_tester_rw	"Description: HS RX clock lane enable Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_CLKLANE_LANE_3,offset=0x304

Bits	Access Mode	Name	Description
7	R/W	lprxponcd_bypass_clklane_rw	"Description: Power on the low power contention detector receiver bypass control Value After Reset: 0x0"
6:0	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_CLKLANE_LANE_4,offset=0x305

Bits	Access Mode	Name	Description
7:5	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
4	R/W	lprxponulp_clklane_rw	"Description: Power on of ULP mode Value After Reset: 0x0"
3	R/W	lprxponulp_bypass_clklane_rw	"Description: Power on of ULP mode of bypass control Value After Reset: 0x0"
2	R/W	lprxponlp_clklane_rw	"Description: Power on of the low power receiver Value After Reset: 0x0"
1	R/W	lprxponlp_bypass_clklane_rw	"Description: Power on of the low power receiver bypass control Value After Reset: 0x0"
0	R/W	lprxponcd_clklane_rw	"Description: Power on the low power contention detector receiver Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_CLKLANE_LANE_6,offset=0x307

Bits	Access Mode	Name	Description
7	R/W	"rxclk_rxhs_pull_long_channel_if_rw"	"Description: RX HS pull long channel Value After Reset: 0x0"
6	R/W	rxclk_rxhs_int_clk_sel_rw	"Description: RX HS internal clock selection Value After Reset: 0x0"
5	R/W	"rxclk_rxhs_feed_int_clk_ovr_en_rw"	"Description: RX HS source data override enable control Value After Reset: 0x0"
4	R/W	rxclk_rxhs_feed_int_clk_ovr_rw	"Description: RX HS source data override Value After Reset: 0x0"

Bits	Access Mode	Name	Description
3	R/W	rxclk_rxhs_ddr_clk_en_if_rw	"Description: RX HS DDR clock enable bit Value After Reset: 0x0"
2	R/W	"rxclk_rxhs_clk_to_long_channel_if_rw"	"Description: RX HS clock to long channel bits Value After Reset: 0x0"
1	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
0	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_CLKLANE_LANE_7,offset=0x308

Bits	Access Mode	Name	Description
7:5	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
4	R/W	rxclk_rxhs_rx_en200mode_if_rw	"Description: Clock lane's 200 ohm differential termination enable control Value After Reset: 0x0"
3	R/W	rxclk_rxhs_ddr_clk_sel_if_rw	"Description: DDR clock lane source selection control (internal or external through long channel) Value After Reset: 0x0"
2:1	R/W	"rxclk_rxhs_ddr_clk_div_ovr_rw[1:0]"	"Description: Clock lane's DDR clock division factor override Value After Reset: 0x0"
0	R/W	"rxclk_rxhs_ddr_clk_div_ovr_en_rw"	"Description: Clock lane's DDR clock division factor override enable Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_CLKLANE_LANE_8,offset=0x309

Bits	Access Mode	Name	Description
7	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
6:3	R/W	rxclk_lb_prog_if_rw[3:0]	"Description: clock lane receiver biasing currents Value After Reset: 0xf"
2	R/W	rxclk_lb_res_cm_en_if_rw	"Description: clock lane static termination driver middle point pull down enable Value After Reset: 0x0"

Bits	Access Mode	Name	Description
1	R/W	rxclk_lb_res_dn_en_if_rw	"Description: clock lane static termination driver test dn line enable Value After Reset: 0x0"
0	R/W	rxclk_lb_res_dp_en_if_rw	"Description: clock lane static termination driver test dp line enable Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_CLKLANE_LANE_0,offset=0x32b

Bits	Access Mode	Name	Description
7	R	reserved	"Description: Reserved field Value After Reset: 0x0"
6:5	R	lprxdoutlp_clklane_if[1:0]	"Description: LP-RX receiver outputs Value After Reset: 0x0"
4:3	R	lprxdoutcd_clklane_if[1:0]	"Description: LP-CD receiver outputs Value After Reset: 0x0"
2	R	hsrx_en_if_clklane	"Description: HS RX clock lane enable Value After Reset: 0x0"
1	R	enable_clklane	"Description: Enable signal for clock lane Value After Reset: 0x0"
0	R	bintpon_if_clklane	"Description: Bias block power-on override enable Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_CLKLANE_LANE_1,offset=0x32c

Bits	Access Mode	Name	Description
7:2	R	reserved	"Description: Reserved field Value After Reset: 0x0"
1:0	R	lprxdoutulp_clklane_if[1:0]	"Description: Low power RX ULP output data Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_CLKLANE_LANE_2,offset=0x32d

Bits	Access Mode	Name	Description
7	R	rxclk_rxhs_rx_enterm_if	"Description: Clock lane HS termination enable Value After Reset: 0x0"
6	R	rxclk_rxhs_rx_enlogic_if	"Description: Clock lane level shifter enable Value After Reset: 0x0"
5	R	rxclk_rxhs_rx_en_if	"Description: Clock lane HS receiver enable Value After Reset: 0x0"
4	R	rxclk_rxhs_osc_out_if	"Description: Internal burnin oscillator clock Value After Reset: 0x0"

Bits	Access Mode	Name	Description
3	R	rxclk_rxhs_ddr_clk_if	"Description: Clock lane's divided DDR clock Value After Reset: 0x0"
2:1	R	rxclk_rxhs_ddr_clk_div[1:0]	"Description: DDR clock division ratio Value After Reset: 0x0"
0	R	lsrxclk_clklane	"Description: Low power clock lane clock Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_CLKLANE_LANE_3,offset=0x32e

Bits	Access Mode	Name	Description
7:4	R	reserved	"Description: Reserved field Value After Reset: 0x0"
3	R	rxclk_rxlp_ponulp_if	"Description: Clock lane ULP receiver Value After Reset: 0x0"
2	R	rxclk_rxlp_ponlp_if	"Description: Clock lane LP receiver Value After Reset: 0x0"
1	R	rxclk_rxlp_poncd_if	"Description: Clock lane LP contention detector Value After Reset: 0x0"
0	R	rxclk_rxhs_rx_pon_if	"Description: Clock lane HS receiver power on Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_CLKLANE_OFFSET_CAL_0,offset=0x37a

Bits	Access Mode	Name	Description
7:2	R/W	"rxclk_rxhs_clk_offset_cal_ovr_rw[5:0]"	"Description: RX HS clock offset calibration override control Value After Reset: 0x0"
1	R/W	rxclk_caldone_ovr_rw	"Description: Clock lane offset calibration done override Value After Reset: 0x0"
0	R/W	rxclk_caldone_ovr_en_rw	"Description: Clock lane offset calibration done override enable Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_CLKLANE_OFFSET_CAL_1,offset=0x37b

Bits	Access Mode	Name	Description
7:5	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"

Bits	Access Mode	Name	Description
4	R/W	rxclk_start_calibration_ovr_rw	"Description: Clock lane digital offset start calibration override enable control Value After Reset: 0x0"
3	R/W	rxclk_start_calibration_ovr_en_rw	"Description: Clock lane digital offset start calibration override enable control Value After Reset: 0x0"
2	R/W	"rxclk_rxhs_start_calibration_ovr_rw"	"Description: Clock lane analog offset start calibration override enable control Value After Reset: 0x0"
1	R/W	"rxclk_rxhs_start_calibration_ovr_en_rw"	"Description: Clock lane analog offset start calibration override enable control Value After Reset: 0x0"
0	R/W	"rxclk_rxhs_clk_offset_cal_ovr_en_rw"	"Description: RX HS clock offset calibration override enable control Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_CLKLANE_OFFSET_CAL_0,offset=0x39d

Bits	Access Mode	Name	Description
7:6	R	reserved	"Description: Reserved field Value After Reset: 0x0"
5	R	rxclk_rxhs_clk_m_cal	"Description: Clock lane offset calibration m-line comparator output Value After Reset: 0x0"
4:1	R	rxclk_errcal[3:0]	"Description: Clock lane offset calibration error rxclk_errcal[3:0]=0x1 This value corresponds to a multi-toggle flag which is not considered an error Value After Reset: 0x0"
0	R	rxclk_caldone	"Description: Clock lane offset calibration done Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_CLKLANE_OFFSET_CAL_1,offset=0x39e

Bits	Access Mode	Name	Description
7	R	rxclk_rxhs_start_calibration	"Description: Clock lane analog offset start calibration Value After Reset: 0x0"

Bits	Access Mode	Name	Description
6	R	rxclk_rxhs_clk_p_cal	"Description: Clock lane offset calibration p-line comparator output Value After Reset: 0x0"
5:0	R	rxclk_rxhs_clk_offset_cal[5:0]	"Description: Clock lane offset calibration setting Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_CLKLANE_OFFSET_CAL_2,offset=0x39f

Bits	Access Mode	Name	Description
7:1	R	reserved	"Description: Reserved field Value After Reset: 0x0"
0	R	rxclk_start_calibration	"Description: Clock lane digital offset start calibration Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_LANE0_LANE_0,offset=0x501

Bits	Access Mode	Name	Description
7	R/W	hsrx_en_bypass_lane0_rw	"Description: HS RX enable bypass Value After Reset: 0x0"
6	R/W	bintpon_lane0_rw	"Description: Bias block power-on; can be overridden if bypass signal is 1'b1 Value After Reset: 0x0"
5	R/W	bintpon_bypass_lane0_rw	"Description: Bias block power-on bypass override Value After Reset: 0x0"
4:1	R/W	atb_sel_lane0_rw[3:0]	"Description: Analog Test Bus signals selection: Value After Reset: 0x0"
0	R/W	atb_lptx1200_on_lane0_rw	"Description: Selection to to probe in analog test bus LP transmitter voltage reference Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_LANE0_LANE_1,offset=0x502

Bits	Access Mode	Name	Description
7:5	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
4	R/W	hsrxpon_if_lane0_rw	"Description: Power on of the RX HS Value After Reset: 0x0"
3	R/W	hsrxpon_bypass_lane0_rw	"Description: Power on of the RX HS bypass control Value After Reset: 0x0"
2	R/W	hsrxenterm_if_lane0_rw	"Description: RX HS enable term Value After Reset: 0x0"

Bits	Access Mode	Name	Description
1	R/W	hsrxenterm_bypass_lane0_rw	"Description: Bypass control of RX high speed enable term Value After Reset: 0x0"
0	R/W	hsrx_en_if_lane0_tester_rw	"Description: HS RX enable for tester Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_LANE0_LANE_3,offset=0x504

Bits	Access Mode	Name	Description
7	R/W	lprxponcd_bypass_lane0_rw	"Description: LP RX Contention Detector power-on bypass override Value After Reset: 0x0"
6:0	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_LANE0_LANE_4,offset=0x505

Bits	Access Mode	Name	Description
7:6	R/W	lptxdin_lane0_rw[1:0]	"Description: LP TX Driver Data input, can be overridden if bypass signal is 1'b1 Value After Reset: 0x0"
5	R/W	lptxdin_bypass_lane0_rw	"Description: LP TX Driver Data input bypass override Value After Reset: 0x0"
4	R/W	lprxponulp_lane0_rw	"Description: LP RX ULP power-on; can be overridden if bypass signal is 1'b1 Value After Reset: 0x0"
3	R/W	lprxponulp_bypass_lane0_rw	"Description: LP RX ULP power-on bypass override Value After Reset: 0x0"
2	R/W	lprxponlp_lane0_rw	"Description: LP RX LP power-on; can be overridden if bypass signal is 1'b1 Value After Reset: 0x0"
1	R/W	lprxponlp_bypass_lane0_rw	"Description: LP RX LP power-on bypass override Value After Reset: 0x0"
0	R/W	lprxponcd_lane0_rw	"Description: LP RX Contention Detector power-on; can be overridden if bypass signal is 1'b1 Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_LANE0_LANE_5,offset=0x506

Bits	Access Mode	Name	Description
7	R/W	lptxpon_lane0_rw	"Description: LP TX power-on Value After Reset: 0x0"

Bits	Access Mode	Name	Description
6	R/W	lptxpon_bypass_lane0_rw	"Description: LP TX power-on bypass bit Value After Reset: 0x0"
5	R/W	lptxenulp_lane0_rw	"Description: LP TX enable ULP Value After Reset: 0x0"
4	R/W	lptxenulp_bypass_lane0_rw	"Description: LP TX enable ULP bypass bit Value After Reset: 0x1"
3	R/W	lptxenpulldwn_lane0_rw	"Description: LP TX enable pull-down Value After Reset: 0x0"
2	R/W	lptxenpulldwn_bypass_lane0_rw	"Description: LP TX enable pull-down bypass bit Value After Reset: 0x0"
1	R/W	lptxena_lane0_rw	"Description: LP TX enable Value After Reset: 0x0"
0	R/W	lptxena_bypass_lane0_rw	"Description: LP TX enable bypass bit Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_LANE0_LANE_6,offset=0x507

Bits	Access Mode	Name	Description
7:6	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
5	R/W	rx0_rxhs_feed_int_clk_ovr_en_rw	"Description: HS-RX source clock override enable control for lane 0. Selects between internal clock source or external(receiver) Value After Reset: 0x0"
4	R/W	rx0_rxhs_feed_int_clk_ovr_rw	"Description: HS-RX source clock override control for lane 0. Selects between internal clock source or external (receiver) Value After Reset: 0x0"
3:1	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
0	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_LANE0_LANE_7,offset=0x508

Bits	Access Mode	Name	Description
7:6	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"

Bits	Access Mode	Name	Description
5	R/W	"rx0_rxhs_clk_from_long_channel_if_rw"	"Description: RX HS clock from long channel Value After Reset: 0x0"
4	R/W	recal_lane0_ovr_rw	"Description: Override Enable voltage control for recal Value After Reset: 0x0"
3	R/W	recal_lane0_ovr_en_rw	"Description: Override Enable voltage control for recal Value After Reset: 0x0"
2:0	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_LANE0_LANE_8,offset=0x509

Bits	Access Mode	Name	Description
7	R/W	rx_des_en_lane0_rw	"Description: RX deserializer enable Value After Reset: 0x0"
6	R/W	rx_des_en_bypass_lane0_rw	"Description: RX deserializer enable bypass control Value After Reset: 0x0"
5	R/W	rx0_txlp_sr_bypass_if_rw	"Description: LP TX slew-rate bypass Value After Reset: 0x0"
4:3	R/W	rx0_txlp_eniocont_if_rw[1:0]	"Description: RX IO continuity enable Value After Reset: 0x0"
2:0	R/W	rx0_rxhs_data_eq_if_rw[2:0]	"Description: RX HS data equalizer Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_LANE0_LANE_9,offset=0x50a

Bits	Access Mode	Name	Description
7	R/W	"rx0_pm_sample_error_counter_rw"	"Description: strobe to read the error counter Value After Reset: 0x0"
6	R/W	rx0_pm_start_control_ovr_en_rw	"Description: Pattern matcher start override enable. Value After Reset: 0x0"
5	R/W	rx0_pm_start_control_ovr_rw	"Description: Pattern matcher start override. Value After Reset: 0x0"
4:1	R/W	rx0_pm_mode_r_rw[3:0]	"Description: Configuration mode for the pattern matcher Value After Reset: 0x7"
0	R/W	"rx0_data_retiming_out_swap_sel_rw"	"Description: retiming mux control Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_LANE0_LANE_12,offset=0x50d

Bits	Access Mode	Name	Description
7	R/W	rx0_pm_man_error_latch_rw	"Description: Allows manual pattern matcher error counter latch. Latching is controlled by rx0_pm_sample_error_counter Value After Reset: 0x0"
6:3	R/W	rx0_lb_prog_if_rw[3:0]	"Description: lane 0 receiver biasing currents Value After Reset: 0xf"
2	R/W	rx0_lb_res_cm_en_if_rw	"Description: lane 0 static termination driver middle point pull down enable Value After Reset: 0x0"
1	R/W	rx0_lb_res_dn_en_if_rw	"Description: lane 0 static termination driver test dn line enable Value After Reset: 0x0"
0	R/W	rx0_lb_res_dp_en_if_rw	"Description: lane 0 static termination driver test dp line enable Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_LANE0_LANE_0,offset=0x52b

Bits	Access Mode	Name	Description
7:4	R	reserved	"Description: Reserved field Value After Reset: 0x0"
3	R	rx0_rxhs_deserial_data_en_if	"Description: HS-RX deserializer reset observation for lane 0 Value After Reset: 0x0"
2	R	hsrx_en_if_lane0	"Description: HS-RX enable of lane 0 Value After Reset: 0x0"
1	R	enable_lane0	"Description: Signal that passes AFE control from soft macro to lane 0 Value After Reset: 0x0"
0	R	bintpon_if_lane0	"Description: Level shifter enable of lane 0 Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_LANE0_LANE_1,offset=0x52c

Bits	Access Mode	Name	Description
7	R	lptxpon_if_lane0	"Description: LP-TX power on Value After Reset: 0x0"

Bits	Access Mode	Name	Description
6	R	lptxenulp_if_lane0	"Description: LP-TX pull up enable (for stopstate) Value After Reset: 0x0"
5	R	lptxenpulldn_if_lane0	"Description: LP-TX pull down enable (for ULP) Value After Reset: 0x0"
4	R	lptxena_if_lane0	"Description: LP-TX driver enable Value After Reset: 0x0"
3:2	R	lptxdin_if_lane0[1:0]	"Description: LP-TX data in Value After Reset: 0x0"
1:0	R	reserved	"Description: Reserved field Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_LANE0_LANE_2,offset=0x52d

Bits	Access Mode	Name	Description
7	R	rx0_rxhs_rx_enterm_if	"Description: RX HS enable term Value After Reset: 0x0"
6	R	rx0_rxhs_rx_enlogic_if	"Description: RX HS enlogic Value After Reset: 0x0"
5	R	rx0_rxhs_rx_en_if	"Description: RX HS enable Value After Reset: 0x0"
4:0	R	reserved	"Description: Reserved field Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_LANE0_LANE_3,offset=0x52e

Bits	Access Mode	Name	Description
7:6	R	lprxdoutlp_lane0[1:0]	"Description: LP-RX receiver outputs Value After Reset: 0x0"
5:4	R	lprxdoutcd_lane0[1:0]	"Description: LP-CD receiver outputs Value After Reset: 0x0"
3:1	R	reserved	"Description: Reserved field Value After Reset: 0x0"
0	R	rx0_rxhs_rx_pon_if	"Description: RX HS power on Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_LANE0_LANE_4,offset=0x52f

Bits	Access Mode	Name	Description
7:5	R	reserved	"Description: Reserved field Value After Reset: 0x0"
4	R	rx0_deskewcalfailed_if	"Description: Deskew calibration failed signal Value After Reset: 0x0"

Bits	Access Mode	Name	Description
3	R	rx0_deskewcaldone_if	"Description: Deskew calibration don Value After Reset: 0x0"
2	R	lsrxclk_lane0	"Description: Low speed clock (XOR'ed from data lines) Value After Reset: 0x0"
1:0	R	lprxdoutulp_lane0[1:0]	"Description: LP-RX ULP receiver outputs Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_LANE0_LANE_5,offset=0x530

Bits	Access Mode	Name	Description
7:0	R	rx0_rxhs_data_if[7:0]	"Description: RX HS data Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_LANE0_LANE_6,offset=0x531

Bits	Access Mode	Name	Description
7:6	R	reserved	"Description: Reserved field Value After Reset: 0x0"
5	R	rx0_pm_error_pm	"Description: Observation of the error flag generated by the master Value After Reset: 0x0"
4:3	R	rx0_txlp_iocont_if[1:0]	"Description: IO continuity receiver output Value After Reset: 0x0"
2	R	rx0_rxhs_word_clk_if	"Description: RX HS word clock Value After Reset: 0x0"
1	R	rx0_rxhs_feed_int_clk_if	"Description: RX HS internal clock feedback Value After Reset: 0x0"
0	R	rx0_rxhs_des_en_if	"Description: RX HS deserializer enable Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_LANE0_LANE_7,offset=0x532

Bits	Access Mode	Name	Description
7:0	R	rx0_pm_error_counter[7:0]	"Description: Error counter value Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_LANE0_LANE_8,offset=0x533

Bits	Access Mode	Name	Description
7:0	R	rx0_pm_error_counter[15:8]	"Description: Error counter value Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_LANE0_LANE_9,offset=0x534

Bits	Access Mode	Name	Description

Bits	Access Mode	Name	Description
7:1	R	reserved	"Description: Reserved field Value After Reset: 0x0"
0	R	rx0_pm_start	"Description: Pattern matcher start flag observation Value After Reset: 0x1"

Name=dphy2rxtester_DIG_RD_RX_LANE0_AFE_OBS_2,offset=0x556

Bits	Access Mode	Name	Description
7	R	tx0_txlp_enulp	"Description: LP-TX ULP enable Value After Reset: 0x0"
6	R	tx0_txlp_ena	"Description: LP-TX driver enable Value After Reset: 0x0"
5	R	tx0_txlp_enlogic	"Description: LP-TX level shifter enable Value After Reset: 0x0"
4	R	tx0_txlp_pon	"Description: LP-TX driver power on Value After Reset: 0x0"
3:0	R	reserved	"Description: Reserved field Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_TX_LANE0_AFE_OBS_3,offset=0x557

Bits	Access Mode	Name	Description
7	R	reserved	"Description: Reserved field Value After Reset: 0x0"
6	R	tx0_rxlp_ponulp	"Description: LP-RX ULP receiver power on Value After Reset: 0x0"
5	R	tx0_rxlp_ponlp	"Description: LP-RX LP receiver power on Value After Reset: 0x0"
4	R	tx0_rxlp_poncd	"Description: LP-RX contention detector receiver power on Value After Reset: 0x0"
3	R	tx0_txlp_sr_bypass	"Description: HS-TX slew rate circuitry bypass Value After Reset: 0x0"
2	R	tx0_txlp_enpulldwn	"Description: LP-TX pull down enable Value After Reset: 0x0"
1:0	R	tx0_txlp_din[1:0]	"Description: LP-TX driver data in Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_LANE0_OFFSET_CAL_0,offset=0x57b

Bits	Access Mode	Name	Description

Bits	Access Mode	Name	Description
7	R/W	"rx0_rxhs_data_offset_cal_en_ovr_en_rw"	"Description: Data offset calibration override enable control Value After Reset: 0x0"
6	R/W	rx0_cal_done_ovr_rw	"Description: Calibration done override enable control Value After Reset: 0x0"
5	R/W	rx0_cal_done_ovr_en_rw	"Description: Calibration done override enable control Value After Reset: 0x0"
4	R/W	rx0_start_calibration_ovr_rw	"Description: RX start calibration Value After Reset: 0x0"
3	R/W	rx0_start_calibration_ovr_en_rw	"Description: RX start calibration Value After Reset: 0x0"
2:0	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_LANE0_OFFSET_CAL_1,offset=0x57c

Bits	Access Mode	Name	Description
7	R/W	"rx0_rxhs_data_offset_cal_ovr_en_rw"	"Description: Offset calibration setting override Value After Reset: 0x0"
6:1	R/W	"rx0_rxhs_data_offset_cal_ovr_rw[5:0]"	"Description: Offset calibration setting override enable Value After Reset: 0x0"
0	R/W	"rx0_rxhs_data_offset_cal_en_ovr_rw"	"Description: Data offset calibration override enable control Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_LANE0_OFFSET_CAL_2,offset=0x57d

Bits	Access Mode	Name	Description
7:0	R/W	rx0_rxhs_data_ovr_rw[7:0]	"Description: RX HS data override Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_LANE0_OFFSET_CAL_3,offset=0x57e

Bits	Access Mode	Name	Description
7:2	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
1	R/W	rx0_rxhs_repeatcal_rw	"Description: RX HS repeat calibration Value After Reset: 0x0"

Bits	Access Mode	Name	Description
0	R/W	rx0_rxhs_data_ovr_en_rw	"Description: RX HS data override enable Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_LANE0_OFFSET_CAL_0,offset=0x59f

Bits	Access Mode	Name	Description
7:3	R	reserved	"Description: Reserved field Value After Reset: 0x0"
2	R	rx0_cal_error	"Description: Offset calibration algorithm error Value After Reset: 0x0"
1	R	rx0_cal_done	"Description: Offset calibration algorithm done Value After Reset: 0x0"
0	R	rx0_start_calibration	"Description: Offset calibration algorithm start Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_LANE0_OFFSET_CAL_1,offset=0x5a0

Bits	Access Mode	Name	Description
7:0	R	rx0_rxhs_data_int[7:0]	"Description: HS-RX data received (AFE output) Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_LANE0_OFFSET_CAL_2,offset=0x5a1

Bits	Access Mode	Name	Description
7	R	reserved	"Description: Reserved field Value After Reset: 0x0"
6	R	rx0_rxhs_data_offset_cal_en	"Description: Offset calibration algorithm enable Value After Reset: 0x0"
5:0	R	rx0_rxhs_data_offset_cal[5:0]	"Description: Offset calibration algorithm setting chosen Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_LANE0_OFFSET_CAL_3,offset=0x5a2

Bits	Access Mode	Name	Description
7:0	R	rx0_total_ones[7:0]	"Description: Offset calibration algorithm counter of 1s Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_LANE0_DDL_0,offset=0x5e0

Bits	Access Mode	Name	Description
7:4	R	reserved	"Description: Reserved field Value After Reset: 0x0"
3	R	rx0_ddl_start_calibration_if	"Description: DDL tune digital algorithm start Value After Reset: 0x0"

Bits	Access Mode	Name	Description
2	R	rx0_ddl_finished_if	"Description: DDL tune done indication Value After Reset: 0x0"
1	R	rx0_ddl_error_flag_if	"Description: DDL tune error Value After Reset: 0x0"
0	R	ddl_tune_en_lane0_if	"Description: DDL tune analog circuitry enable indication Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_LANE0_DDL_1,offset=0x5e1

Bits	Access Mode	Name	Description
7:0	R	rx0_freq_meas_out_if[7:0]	"Description: RX DDL tuning oscillation measurement Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_LANE0_DDL_2,offset=0x5e2

Bits	Access Mode	Name	Description
7:4	R	reserved	"Description: Reserved field Value After Reset: 0x0"
3:0	R	rx0_freq_meas_out_if[11:8]	"Description: RX DDL tuning oscillation measurement Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_LANE0_DDL_3,offset=0x5e3

Bits	Access Mode	Name	Description
7:0	R	rx0_osc_freq_target_if[7:0]	"Description: RX oscillator frequency target obs Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_LANE0_DDL_4,offset=0x5e4

Bits	Access Mode	Name	Description
7:4	R	reserved	"Description: Reserved field Value After Reset: 0x0"
3:0	R	rx0_osc_freq_target_if[11:8]	"Description: RX oscillator frequency target obs Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_LANE0_DDL_5,offset=0x5e5

Bits	Access Mode	Name	Description
7:6	R	reserved	"Description: Reserved field Value After Reset: 0x0"
5	R	rx0_rxhs_ddl_phase_change_p_if	"Description: RX HS DDL phase change Value After Reset: 0x0"
4:0	R	rx0_rxhs_ddl_tune_if[4:0]	"Description: RX HS DDL tuning setting Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_LANE0_DDL_6,offset=0x5e6

Bits	Access Mode	Name	Description
7:5	R	reserved	"Description: Reserved field Value After Reset: 0x0"
4:0	R	rx0_rxhs_ddl_phase_clk_if[4:0]	"Description: Clock DDL setting chosen Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_LANE0_DDL_7,offset=0x5e7

Bits	Access Mode	Name	Description
7:4	R	reserved	"Description: Reserved field Value After Reset: 0x0"
3:0	R	rx0_rxhs_ddl_phase_data_if[3:0]	"Description: Data DDL setting chosen Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_LANE0_DDL_0,offset=0x606

Bits	Access Mode	Name	Description
7:6	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
5	R/W	rx0_rxhs_ddl_tune_en_if_rw	"Description: RX HS DDL tune enable Value After Reset: 0x0"
4	R/W	rx0_rxhs_ddl_tune_en_bypass_rw	"Description: RX HS DDL tune enable bypass Value After Reset: 0x0"
3	R/W	rx0_ddl_start_calibration_ovr_rw	"Description: RX DDL tuning calibration start override Value After Reset: 0x0"
2	R/W	"rx0_ddl_start_calibration_ovr_en_ rw"	"Description: RX DDL tuning calibration start override enable Value After Reset: 0x0"
1	R/W	rx0_ddl_finished_ovr_rw	"Description: RX DDL tuning done override Value After Reset: 0x0"
0	R/W	rx0_ddl_finished_ovr_en_rw	"Description: RX DDL tuning done override enable Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_LANE0_DDL_1,offset=0x607

Bits	Access Mode	Name	Description
7	R/W	"rx0_rxhs_ddl_phase_change_p_o vr_rw"	"Description: RX HS DDL phase changed override control Value After Reset: 0x0"

Bits	Access Mode	Name	Description
6	R/W	"rx0_rxhs_ddl_phase_change_p_ovr_en_rw"	"Description: RX HS DDL phase changed override control Value After Reset: 0x0"
5	R/W	rx0_rxhs_ddl_tune_ovr_en_rw	"Description: RX HS DDL tune override enable control Value After Reset: 0x0"
4:0	R/W	rx0_rxhs_ddl_tune_ovr_rw[4:0]	"Description: RX HS DDL tune override for lane 0 Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_LANE0_DDL_2,offset=0x608

Bits	Access Mode	Name	Description
7:6	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
5	R/W	"rx0_rxhs_ddl_phase_clk_ovr_en_rw"	"Description: RX HS DDL phase clock override control Value After Reset: 0x0"
4:0	R/W	"rx0_rxhs_ddl_phase_clk_ovr_rw[4:0]"	"Description: RX HS DDL phase clock override Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_LANE0_DDL_3,offset=0x609

Bits	Access Mode	Name	Description
7:5	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
4	R/W	"rx0_rxhs_ddl_phase_data_ovr_en_rw"	"Description: Override enable for ddl phase data Value After Reset: 0x0"
3:0	R/W	"rx0_rxhs_ddl_phase_data_ovr_rw[3:0]"	"Description: RX HS DDL phase data override Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_LANE0_DDL_4,offset=0x60a

Bits	Access Mode	Name	Description
7:0	R/W	"rx0_osc_freq_target_if_ovr_rw[7:0]"	"Description: Oscillating frequency override Target Value After Reset: 0x77"

Name=dphy2rxtester_DIG_RDWR_RX_LANE0_DDL_5,offset=0x60b

Bits	Access Mode	Name	Description
7:4	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
3:0	R/W	"rx0_osc_freq_target_if_ovr_rw[11:8]"	"Description: Oscillating frequency override Target Value After Reset: 0x1"

Name=dphy2rxtester_DIG_RDWR_RX_LANE0_DDL_6,offset=0x60c

Bits	Access Mode	Name	Description
7:1	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
0	R/W	rx0_osc_freq_target_if_ovr_en_rw	"Description: Override for oscillation frequency target Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_LANE1_LANE_0,offset=0x701

Bits	Access Mode	Name	Description
7	R/W	hsrx_en_bypass_lane1_rw	"Description: HS RX enable bypass Value After Reset: 0x0"
6	R/W	bintpon_lane1_rw	"Description: Bias block power-on; can be overridden if bypass signal is 1'b1 Value After Reset: 0x0"
5	R/W	bintpon_bypass_lane1_rw	"Description: Bias block power-on bypass override Value After Reset: 0x0"
4:1	R/W	atb_sel_lane1_rw[3:0]	"Description: Analog Test Bus signals selection: Value After Reset: 0x0"
0	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_LANE1_LANE_1,offset=0x702

Bits	Access Mode	Name	Description
7:5	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
4	R/W	hsrxpon_if_lane1_rw	"Description: Power on of the RX HS Value After Reset: 0x0"
3	R/W	hsrxpon_bypass_lane1_rw	"Description: Power on of the RX HS bypass control Value After Reset: 0x0"
2	R/W	hsrxenterm_if_lane1_rw	"Description: RX HS enable term Value After Reset: 0x0"
1	R/W	hsrxenterm_bypass_lane1_rw	"Description: Bypass control of RX high speed enable term Value After Reset: 0x0"
0	R/W	hsrx_en_if_lane1_tester_rw	"Description: HS RX enable for tester Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_LANE1_LANE_3,offset=0x704

Bits	Access Mode	Name	Description

Bits	Access Mode	Name	Description
7	R/W	lprxponcd_bypass_lane1_rw	"Description: LP RX Contention Detector power-on bypass override Value After Reset: 0x0"
6:0	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_LANE1_LANE_4,offset=0x705

Bits	Access Mode	Name	Description
7:5	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
4	R/W	lprxponulp_lane1_rw	"Description: LP RX ULP power-on; can be overridden if bypass signal is 1'b1 Value After Reset: 0x0"
3	R/W	lprxponulp_bypass_lane1_rw	"Description: LP RX ULP power-on bypass override Value After Reset: 0x0"
2	R/W	lprxponlp_lane1_rw	"Description: LP RX LP power-on; can be overridden if bypass signal is 1'b1 Value After Reset: 0x0"
1	R/W	lprxponlp_bypass_lane1_rw	"Description: LP RX LP power-on bypass override Value After Reset: 0x0"
0	R/W	lprxponcd_lane1_rw	"Description: LP RX Contention Detector power-on; can be overridden if bypass signal is 1'b1 Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_LANE1_LANE_6,offset=0x707

Bits	Access Mode	Name	Description
7:6	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
5	R/W	rx1_rxhs_feed_int_clk_ovr_en_rw	"Description: HS-RX source clock override enable control for lane 1. Selects between internal clock source or external(receiver) Value After Reset: 0x0"
4	R/W	rx1_rxhs_feed_int_clk_ovr_rw	"Description: HS-RX source clock override control for lane 1. Selects between internal clock source or external (receiver) Value After Reset: 0x0"
3:1	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"

Bits	Access Mode	Name	Description
0	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_LANE1_LANE_7,offset=0x708

Bits	Access Mode	Name	Description
7:6	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
5	R/W	"rx1_rxhs_clk_from_long_channel_if_rw"	"Description: RX HS clock from long channel Value After Reset: 0x0"
4	R/W	recal_lane1_ovr_rw	"Description: Override Enable voltage control for recal Value After Reset: 0x0"
3	R/W	recal_lane1_ovr_en_rw	"Description: Override Enable voltage control for recal Value After Reset: 0x0"
2:0	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_LANE1_LANE_8,offset=0x709

Bits	Access Mode	Name	Description
7	R/W	rx_des_en_lane1_rw	"Description: RX deserializer enable Value After Reset: 0x0"
6	R/W	rx_des_en_bypass_lane1_rw	"Description: RX deserializer enable bypass control Value After Reset: 0x0"
5:3	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
2:0	R/W	rx1_rxhs_data_eq_if_rw[2:0]	"Description: RX HS data equalizer Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_LANE1_LANE_9,offset=0x70a

Bits	Access Mode	Name	Description
7	R/W	"rx1_pm_sample_error_counter_rw"	"Description: strobe to read the error counter Value After Reset: 0x0"
6	R/W	rx1_pm_start_control_ovr_en_rw	"Description: Pattern matcher start override enable. Value After Reset: 0x0"
5	R/W	rx1_pm_start_control_ovr_rw	"Description: Pattern matcher start override. Value After Reset: 0x0"

Bits	Access Mode	Name	Description
4:1	R/W	rx1_pm_mode_r_rw[3:0]	"Description: Configuration mode for the pattern matcher Value After Reset: 0x7"
0	R/W	"rx1_data_retiming_out_swap_sel_rw"	"Description: retiming mux control Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_LANE1_LANE_10,offset=0x70b

Bits	Access Mode	Name	Description
7	R/W	"rx1_rxhs_ddl_phase_clk_ovr_en_rw"	"Description: RX HS DDL phase data override control Value After Reset: 0x0"
6:2	R/W	"rx1_rxhs_ddl_phase_clk_ovr_rw[4:0]"	"Description: RX HS DDL phase clock override Value After Reset: 0x0"
1	R/W	"rx1_rxhs_ddl_phase_change_p_ovr_en_rw"	"Description: RX HS DDL phase changed override control Value After Reset: 0x0"
0	R/W	"rx1_rxhs_ddl_phase_change_p_ovr_rw"	"Description: RX HS DDL phase changed override control Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_LANE1_LANE_11,offset=0x70c

Bits	Access Mode	Name	Description
7:5	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
4	R/W	"rx1_rxhs_ddl_phase_data_ovr_en_rw"	"Description: Override enable for ddl phase data Value After Reset: 0x0"
3:0	R/W	"rx1_rxhs_ddl_phase_data_ovr_rw[3:0]"	"Description: RX HS DDL phase clock override Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_LANE1_LANE_12,offset=0x70d

Bits	Access Mode	Name	Description
7	R/W	rx1_pm_man_error_latch_rw	"Description: Allows manual pattern matcher error counter latch. Latching is controlled by rx1_pm_sample_error_counter Value After Reset: 0x0"
6:3	R/W	rx1_lb_prog_if_rw[3:0]	"Description: lane 1 receiver biasing currents Value After Reset: 0xf"

Bits	Access Mode	Name	Description
2	R/W	rx1_lb_res_cm_en_if_rw	"Description: lane 1 static termination driver middle point pull down enable Value After Reset: 0x0"
1	R/W	rx1_lb_res_dn_en_if_rw	"Description: lane 1 static termination driver test dn line enable Value After Reset: 0x0"
0	R/W	rx1_lb_res_dp_en_if_rw	"Description: lane 1 static termination driver test dp line enable Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_LANE1_LANE_0,offset=0x72b

Bits	Access Mode	Name	Description
7:4	R	reserved	"Description: Reserved field Value After Reset: 0x0"
3	R	rx1_rxhs_deserial_data_en_if	"Description: HS-RX deserializer reset observation for lane 1 Value After Reset: 0x0"
2	R	hsrx_en_if_lane1	"Description: HS-RX enable of lane 1 Value After Reset: 0x0"
1	R	enable_lane1	"Description: Signal that passes AFE control from soft macro to lane 1 Value After Reset: 0x0"
0	R	bintpon_if_lane1	"Description: Level shifter enable of lane 1 Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_LANE1_LANE_2,offset=0x72d

Bits	Access Mode	Name	Description
7	R	rx1_rxhs_rx_enterm_if	"Description: RX HS enable term Value After Reset: 0x0"
6	R	rx1_rxhs_rx_enlogic_if	"Description: RX HS enlogic Value After Reset: 0x0"
5	R	rx1_rxhs_rx_en_if	"Description: RX HS enable Value After Reset: 0x0"
4:0	R	reserved	"Description: Reserved field Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_LANE1_LANE_3,offset=0x72e

Bits	Access Mode	Name	Description
7:6	R	lprxdoutlp_lane1[1:0]	"Description: LP-RX receiver outputs Value After Reset: 0x0"
5:4	R	lprxdoutcd_lane1[1:0]	"Description: LP-CD receiver outputs Value After Reset: 0x0"
3	R	rx1_rxlp_ponulp_if	"Description: LP-RX ULP receiver power on Value After Reset: 0x0"
2	R	rx1_rxlp_ponlp_if	"Description: LP-RX LP receiver power on Value After Reset: 0x0"
1	R	rx1_rxlp_poncd_if	"Description: LP-RX contention detector power on Value After Reset: 0x0"
0	R	rx1_rxhs_rx_pon_if	"Description: RX HS power on Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_LANE1_LANE_4,offset=0x72f

Bits	Access Mode	Name	Description
7:5	R	reserved	"Description: Reserved field Value After Reset: 0x0"
4	R	rx1_deskewcalfailed_if	"Description: Deskew calibration failed signal Value After Reset: 0x0"
3	R	rx1_deskewcaldone_if	"Description: Deskew calibration don Value After Reset: 0x0"
2	R	lsrxclk_lane1	"Description: Low speed clock (XOR'ed from data lines) Value After Reset: 0x0"
1:00	R	lprxdoutulp_lane1[1:0]	"Description: LP-RX ULP receiver outputs Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_LANE1_LANE_5,offset=0x730

Bits	Access Mode	Name	Description
7:0	R	rx1_rxhs_data_if[7:0]	"Description: RX HS data Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_LANE1_LANE_6,offset=0x731

Bits	Access Mode	Name	Description
7:6	R	reserved	"Description: Reserved field Value After Reset: 0x0"
5	R	rx1_pm_error_pm	"Description: Observation of the error flag generated by the master Value After Reset: 0x0"
4:3	R	reserved	"Description: Reserved field Value After Reset: 0x0"

Bits	Access Mode	Name	Description
2	R	rx1_rxhs_word_clk_if	"Description: RX HS word clock Value After Reset: 0x0"
1	R	rx1_rxhs_feed_int_clk_if	"Description: RX HS internal clock feedback Value After Reset: 0x0"
0	R	rx1_rxhs_des_en_if	"Description: RX HS deserializer enable Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_LANE1_LANE_7,offset=0x732

Bits	Access Mode	Name	Description
7:0	R	rx1_pm_error_counter[7:0]	"Description: Error counter value Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_LANE1_LANE_8,offset=0x733

Bits	Access Mode	Name	Description
7:0	R	rx1_pm_error_counter[15:8]	"Description: Error counter value Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_LANE1_LANE_9,offset=0x734

Bits	Access Mode	Name	Description
7	R	reserved	"Description: Reserved field Value After Reset: 0x0"
6:2	R	rx1_rxhs_ddl_phase_clk_if[4:0]	"Description: Clock DDL setting chosen Value After Reset: 0x0"
1	R	rx1_rxhs_ddl_phase_change_p_if	"Description: DDL setting change latch pulse Value After Reset: 0x0"
0	R	rx1_pm_start	"Description: Pattern matcher start flag observation Value After Reset: 0x1"

Name=dphy2rxtester_DIG_RD_RX_LANE1_LANE_10,offset=0x735

Bits	Access Mode	Name	Description
7:4	R	reserved	"Description: Reserved field Value After Reset: 0x0"
3:0	R	rx1_rxhs_ddl_phase_data_if[3:0]	"Description: Data DDL setting chosen Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_LANE1_OFFSET_CAL_0,offset=0x77b

Bits	Access Mode	Name	Description

Bits	Access Mode	Name	Description
7	R/W	"rx1_rxhs_data_offset_cal_en_ovr_en_rw"	"Description: Data offset calibration override enable control Value After Reset: 0x0"
6	R/W	rx1_cal_done_ovr_rw	"Description: Calibration done override enable control Value After Reset: 0x0"
5	R/W	rx1_cal_done_ovr_en_rw	"Description: Calibration done override enable control Value After Reset: 0x0"
4	R/W	rx1_start_calibration_ovr_rw	"Description: RX start calibration Value After Reset: 0x0"
3	R/W	rx1_start_calibration_ovr_en_rw	"Description: RX start calibration Value After Reset: 0x0"
2:0	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_LANE1_OFFSET_CAL_1,offset=0x77c

Bits	Access Mode	Name	Description
7	R/W	"rx1_rxhs_data_offset_cal_ovr_en_rw"	"Description: Offset calibration setting override Value After Reset: 0x0"
6:1	R/W	"rx1_rxhs_data_offset_cal_ovr_rw[5:0]"	"Description: Offset calibration setting override enable Value After Reset: 0x0"
0	R/W	"rx1_rxhs_data_offset_cal_en_ovr_rw"	"Description: Data offset calibration override enable control Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_LANE1_OFFSET_CAL_2,offset=0x77d

Bits	Access Mode	Name	Description
7:0	R/W	rx1_rxhs_data_ovr_rw[7:0]	"Description: RX HS data override Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_LANE1_OFFSET_CAL_3,offset=0x77e

Bits	Access Mode	Name	Description
7:2	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
1	R/W	rx1_rxhs_repeatcal_rw	"Description: RX HS repeat calibration Value After Reset: 0x0"

Bits	Access Mode	Name	Description
0	R/W	rx1_rxhs_data_ovr_en_rw	"Description: RX HS data override enable Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_LANE1_OFFSET_CAL_0,offset=0x79f

Bits	Access Mode	Name	Description
7:3	R	reserved	"Description: Reserved field Value After Reset: 0x0"
2	R	rx1_cal_error	"Description: Offset calibration algorithm error Value After Reset: 0x0"
1	R	rx1_cal_done	"Description: Offset calibration algorithm done Value After Reset: 0x0"
0	R	rx1_start_calibration	"Description: Offset calibration algorithm start Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_LANE1_OFFSET_CAL_1,offset=0x7a0

Bits	Access Mode	Name	Description
7:0	R	rx1_rxhs_data_int[7:0]	"Description: HS-RX data received (AFE output) Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_LANE1_OFFSET_CAL_2,offset=0x7a1

Bits	Access Mode	Name	Description
7	R	reserved	"Description: Reserved field Value After Reset: 0x0"
6	R	rx1_rxhs_data_offset_cal_en	"Description: Offset calibration algorithm enable Value After Reset: 0x0"
5:0	R	rx1_rxhs_data_offset_cal[5:0]	"Description: Offset calibration algorithm setting chosen Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_LANE1_OFFSET_CAL_3,offset=0x7a2

Bits	Access Mode	Name	Description
7:0	R	rx1_total_ones[7:0]	"Description: Offset calibration algorithm counter of 1s Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_LANE1_DDL_0,offset=0x7e0

Bits	Access Mode	Name	Description
7:4	R	reserved	"Description: Reserved field Value After Reset: 0x0"
3	R	rx1_ddl_start_calibration_if	"Description: DDL tune digital algorithm start Value After Reset: 0x0"

Bits	Access Mode	Name	Description
2	R	rx1_ddl_finished_if	"Description: DDL tune done indication Value After Reset: 0x0"
1	R	rx1_ddl_error_flag_if	"Description: DDL tune error Value After Reset: 0x0"
0	R	ddl_tune_en_lane1_if	"Description: DDL tune analog circuitry enable indication Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_LANE1_DDL_1,offset=0x7e1

Bits	Access Mode	Name	Description
7:0	R	rx1_freq_meas_out_if[7:0]	"Description: RX DDL tuning oscillation measurement Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_LANE1_DDL_2,offset=0x7e2

Bits	Access Mode	Name	Description
7:4	R	reserved	"Description: Reserved field Value After Reset: 0x0"
3:0	R	rx1_freq_meas_out_if[11:8]	"Description: RX DDL tuning oscillation measurement Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_LANE1_DDL_3,offset=0x7e3

Bits	Access Mode	Name	Description
7:0	R	rx1_osc_freq_target_if[7:0]	"Description: RX oscillator frequency target obs Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_LANE1_DDL_4,offset=0x7e4

Bits	Access Mode	Name	Description
7:4	R	reserved	"Description: Reserved field Value After Reset: 0x0"
3:0	R	rx1_osc_freq_target_if[11:8]	"Description: RX oscillator frequency target obs Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RD_RX_LANE1_DDL_5,offset=0x7e5

Bits	Access Mode	Name	Description
7:5	R	reserved	"Description: Reserved field Value After Reset: 0x0"
4:0	R	rx1_rxhs_ddl_tune_if[4:0]	"Description: RX HS DDL tuning setting Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_LANE1_DDL_0,offset=0x806

Bits	Access Mode	Name	Description
7:6	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
5	R/W	rx1_rxhs_ddl_tune_en_if_rw	"Description: RX HS DDL tune enable Value After Reset: 0x0"
4	R/W	rx1_rxhs_ddl_tune_en_bypass_rw	"Description: RX HS DDL tune enable bypass Value After Reset: 0x0"
3	R/W	rx1_ddl_start_calibration_ovr_rw	"Description: RX DDL tuning calibration start override Value After Reset: 0x0"
2	R/W	"rx1_ddl_start_calibration_ovr_en_ rw"	"Description: RX DDL tuning calibration start override enable Value After Reset: 0x0"
1	R/W	rx1_ddl_finished_ovr_rw	"Description: RX DDL tuning done override Value After Reset: 0x0"
0	R/W	rx1_ddl_finished_ovr_en_rw	"Description: RX DDL tuning done override enable Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_LANE1_DDL_1,offset=0x807

Bits	Access Mode	Name	Description
7:6	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
5	R/W	rx1_rxhs_ddl_tune_ovr_en_rw	"Description: RX HS DDL tune override enable control Value After Reset: 0x0"
4:0	R/W	rx1_rxhs_ddl_tune_ovr_rw[4:0]	"Description: RX HS DDL tune override for lane 1 Value After Reset: 0x0"

Name=dphy2rxtester_DIG_RDWR_RX_LANE1_DDL_4,offset=0x80a

Bits	Access Mode	Name	Description
7:0	R/W	"rx1_osc_freq_target_if_ovr_rw[7:0]"	"Description: Oscillating frequency override Target Value After Reset: 0x77"

Name=dphy2rxtester_DIG_RDWR_RX_LANE1_DDL_5,offset=0x80b

Bits	Access Mode	Name	Description
7:4	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
3:0	R/W	"rx1_osc_freq_target_if_ovr_rw[11: 8]"	"Description: Oscillating frequency override Target Value After Reset: 0x1"

Name=dphy2rxtester_DIG_RDWR_RX_LANE1_DDL_6,offset=0x80c

Bits	Access Mode	Name	Description
7:1	R/W	reserved	"Description: Reserved field Value After Reset: 0x0"
0	R/W	rx1_osc_freq_target_if_ovr_en_rw	"Description: Override for oscillation frequency target Value After Reset: 0x0"

8.3.5 Application description

8.3.5.1 Start-Up sequence for 2lane

Before D-PHY power-up, the D-PHY needs to be properly configured for the desired operation. This needs to be done with the PHY in shutdown mode and it is assumed that such configurations are static or at least are stable prior to leaving the Shutdown mode.

The following configurations need to be applied for the desired operation.

The D-PHY operating frequency, input reference clock frequency and DDL target oscillation frequency need to be configured through hsfreqrange[6:0], cfgclkfreqrange[5:0] and osc_freq_target[11:0].

- IP operating frequency: hsfreqrange[6:0]
- Input reference clock frequency: cfgclkfreqrange[5:0]
- DDL target oscillation frequency: osc_freq_target[11:0]

Regarding input system clock cfg_clk frequency, cfgclkfreqrange[5:0] needs to be configured according to the following formula: $\text{cfgclkfreqrange}[5:0] = \text{round}[(\text{Fcfg\_clk}(\text{MHz}) - 17) * 4]$

Depending on the obtained cfgclkfreqrange the register counter_for_des_en_config_if_rw need also to configured according to Table 8-1.

Table 8-1 counter_for_des_en configuration

cfgclkfreqrange[5:0] (decimal)	counter_for_des_en_config_if[3:0] (decimal)
$0 \leq \text{cfgclkfreqrange}[5:0] \leq 12$	0
$13 \leq \text{cfgclkfreqrange}[5:0] \leq 38$	1
$39 \leq \text{cfgclkfreqrange}[5:0] \leq 40$	2

To configure the bit rate operation you should directly configure hsfreqrange for the desired frequency range or set the register hsfreqrange, "Test Codes: 0x1 and 0x2", with the proper code. You can only configure hsfreqrange before the D-PHY power-up. You can not change hsfreqrange setting during active operation.

The desired operating frequency configuration is required to meet the D-PHY specification global timing parameters and intervals. The D-PHY specification has several global timing parameters and intervals which have to be followed to ensure proper operation.

The DDL target oscillation frequency needs to be configured for the desired operating frequency according to "Frequency Ranges and Defaults". It can be configured through the following test codes:

- global control over all lanes through 0xe2, 0xe3, 0xe4
- independent lane 0 control through 0x60a, 0x60b, 0x60c
- independent lane 1 control through 0x80a, 0x80b, 0x80c

Table 8-2 Frequency Ranges and Defaults

Range (Mbps)	Default Bit Rate (Mbps)	hsfrequentrange[6:0]	osc_freq_target[11:0] (decimal)
80 - 97.125	80	0000000	460
80 - 107.625	90	0010000	460
83.125 - 118.125	100	0100000	460
92.625 - 128.625	110	0110000	460
102.125 - 139.125	120	0000001	460
111.625 - 149.625	130	0010001	460
121.125 - 160.125	140	0100001	460
130.625 - 170.625	150	0110001	460
140.125 - 181.125	160	0000010	460
149.625 - 191.625	170	0010010	460
159.125 - 202.125	180	0100010	460
168.625 - 212.625	190	0110010	460
182.875 - 228.375	205	0000011	460
197.125 - 244.125	220	0010011	460
211.375 - 259.875	235	0100011	460
225.625 - 275.625	250	0110011	460
249.375 - 301.875	275	0000100	460
273.125 - 328.125	300	0010100	460
296.875 - 354.375	325	0100101	460
320.625 - 380.625	350	0110101	460
368.125 - 433.125	400	0000101	460
415.625 - 485.625	450	0010110	460
463.125 - 538.125	500	0100110	460
510.625 - 590.625	550	0110111	460
558.125 - 643.125	600	0000111	460
605.625 - 695.625	650	0011000	460
653.125 - 748.125	700	0101000	460
700.625 - 800.625	750	0111001	460
748.125 - 853.125	800	0001001	460
795.625 - 905.625	850	0011001	460
843.125 - 958.125	900	0101001	460
890.625 - 1010.625	950	0111010	460
938.125 - 1063.125	1000	0001010	460
985.625 - 1115.625	1050	0011010	460

Range (Mbps)	Default Bit Rate (Mbps)	hsfrequentrange[6:0]	osc_freq_target[11:0] (decimal)
1033.125 - 1168.125	1100	0101010	460
1080.625 - 1220.625	1150	0111011	460
1128.125 - 1273.125	1200	0001011	460
1175.625 - 1325.625	1250	0011011	460
1223.125 - 1378.125	1300	0101011	460
1270.625 - 1430.625	1350	0111100	460
1318.125 - 1483.125	1400	0001100	460
1365.625 - 1535.625	1450	0011100	460
1413.125 - 1588.125	1500	0101100	460
1460.625 - 1640.625	1550	0111101	285
1508.125 - 1693.125	1600	0001101	295
1555.625 - 1745.625	1650	0011101	304
1603.125 - 1798.125	1700	0101110	313
1650.625 - 1850.625	1750	0111110	322
1698.125 - 1903.125	1800	0001110	331
1745.625 - 1955.625	1850	0011110	341
1793.125 - 2008.125	1900	0101111	350
1840.625 - 2060.625	1950	0111111	359
1888.125 - 2113.125	2000	0001111	368
1935.625 - 2165.625	2050	1000000	377
1983.125 - 2218.125	2100	1000001	387
2030.625 - 2270.625	2150	1000010	396
2078.125 - 2323.125	2200	1000011	405
2125.625 - 2375.625	2250	1000100	414
2173.125 - 2428.125	2300	1000101	423
2220.625 - 2480.625	2350	1000110	432
2268.125 - 2500	2400	1000111	442
2315.625 - 2500	2450	1001000	451
2363.125 - 2500	2500	1001001	460

The PHY consists of two data lanes, but applications can use two or lesser number data lanes. In such cases, you are granted access to individual enabling signals, enable_n, that control which lanes should be used and evolve through all the necessary initialization steps. The enable signals must not be changed during operation. They should be set in power-down.

If only a single data lane is to be enabled, it must be data lane 0

After the reset signals—rstz and shutdownz—are released, the PHY begins an initialization sequence that allows its

correct operation. shutdownz should precede rstz; it is also assumed that the cfg_clk signal is available and stable by that time. The initialization sequence determines that the bandgap and biasing blocks are enabled first. After the current references get settled, a second step is triggered where the internal calibrations are performed. This includes:

1. Termination resistors calibration
2. Data and clock lane offset compensation
3. Digital Delay Line (DDL) tuning calibration
4. For operating frequencies < 1.5Gbps (DPHY1.1 backwards compatibility mode), an internal deskew compensation algorithm is executed with internal stimuli provided to data and clock lanes.

The deskew algorithm selection is done automatically based on hsfreqrange[6:0] configuration but can be overridden through test control registers, as well as all other power-up calibration algorithms. When this second step is complete, the control is passed to the lanes, which handle the power management for LP/HS requests, enabling or disabling the corresponding receivers. All initialization steps are complete once the stopstatedata_n and stopstatclk outputs get asserted (if D-PHY Transmitter is driving LP11). For operating frequencies > 1.5Gbps, a deskew compensation calibration algorithm can be executed (enabled by default for operating frequencies > 1.5Gbps), if the RX D-PHY is triggered through a defined pattern sent by the TX after the D-PHY RX macro is already powered-up and in STOPSTATE.

Initialization period (TINIT) is a protocol dependent parameter with a minimum 100 μ s defined by the specification. The PHY does not set any limit to the initialization period, meaning it starts decoding the LP commands after the AFE initialization in RX mode. It is up to the controller or the upper layers to ensure the proper initialization times through the correct handling of the PHY control signals. This time must conform to D-PHY specification and obey the minimum specified 100 μ s value.

Power-up configuration examples are provided as below.

By default, the D-PHY power-up sequence performs termination calibration considering the presence of an external termination resistor. For this algorithm to run without external termination resistor, signal noext_burnin_res_cal needs to be configured to enable it.

For D-PHY configuration or debug purposes test and control interface must be exercised after testclr pulse and before IP leaves shutdown mode (shutdownz and rstz de-assertion) and start D-PHY power-up sequence.

While shutdownz and rstz are de-asserted, the DPHY remains in POWERDWN state. If the PHY properly reached the end of its startup sequence, it will be in IDLE state.

Case 1 (operating rate 1 Gbps; external rx; cfg_clk=27 Mhz; D-PHY 1.1 backwards compatibility mode):

1. Set rstz = 1'b0;
2. Set shutdownz = 1'b0;
3. Set testclr = 1'b1;
4. Wait for 15 ns;
5. Set testclr = 1'b0;
6. Set hsfreqrange[6:0] = 7'b0001010;
7. Configure register 0xe5 to set counter_for_des_en_bypass (bit 0) to 1'b1;
8. Refer to Table 5-7 on page 79 and configure register 0xe4 (bits 7:4) with the appropriate value for counter_for_des_en_config_if_rw;
9. Configure register 0x1ab to set cb_sel_vref_lprx_rw (bits 1:0) to 2'b10;
10. Configure register 0x1ac to set rx_rxlp_bias_prog_rw (bit 6) to 1'b1;
11. Configure register 0x8 to set deskew_polarity_rw signal (bit 5) to 1'b1;
12. Configure register 0x307 to set rxclk_rxhs_pull_long_channel_if_rw signal (bit 7) to 1'b1;
13. Configure register 0x607 to set rx0_rxhs_ddl_tune_ovr_en_rw (bit 5) to 1'b1 and set rx0_rxhs_ddl_tune_ovr_rw[4:0] (bits 4:0) to 5'b11111;
14. Configure register 0x807 to set rx1_rxhs_ddl_tune_ovr_en_rw (bit 5) to 1'b1 and set rx1_rxhs_ddl_tune_ovr_rw[4:0] (bits 4:0) to 5'b11111;

15. Set `cfgclkfreqrange[5:0] = round[(Fcfg_clk(MHz)-17)*4] = 6'b011100;`
16. Apply `cfg_clk` signal with the appropriate frequency with 27Mhz frequency;
17. Set `basedir_0 = 1'b1;`
18. Set `forcerxmode_n = 1'b1;`
19. Set all requests inputs to zero;
20. Wait for 15 ns;
21. Set `enable_n` and `enableclk=1'b1;`
22. Wait 5ns;
23. Set `shutdownz=1'b1;`
24. Wait 5ns;
25. Set `rstz=1'b1;`
26. Wait until `stopstatedata_n` and `stopstateclk` outputs are asserted indicating PHY is receiving LP11 in enabled datalanes and clocklanes;
27. Set `forcerxmode_n = 1'b0`

8.3.5.2 Start-Up sequence for 4lane

The procedure to setup and operate both PHYs as a single entity is the following:

- Turn on reset for both macros (`shutdownz=rstz=1'b0`)
- Define master PHY and slave PHY and configure each accordingly (this choice needs to be coherent with wiring connections mentioned above):
 - Set bit [0] (`mastermacro -> set as master PHY`) of register 0x133 of the master PHY to 1'b1
 - Set bit [0] (`mastermacro -> set as slave PHY`) of register 0x133 of the slave PHY to 1'b0
- Configure master PHY clock lane to drive long channel clock:
 - Set bit [2] (`rxclk_rxhs_clk_to_long_channel_if -> enable the hs clock to long channel`) of register 0x307 of the master PHY to 1'b1
 - Set bit [2] (`rxclk_rxhs_clk_to_long_channel_if -> disable the hs clock to long channel`) of register 0x307 of the slave PHY to 1'b0
 - Set bit 7 of register 0x307 of the master PHY and slave PHY to 1'b0
- Configure both master and slave PHY data lanes to get clock from long channel:
 - Set bit [5] (`rxhs_clk_from_long_channel_if_rw -> enable the hs clock to long channel of lane0`) of register 0x508 of the master/slave PHYs to 1'b1
 - Set bit [5] (`rxhs_clk_from_long_channel_if_rw of lane1 -> enable the hs clock to long channel of lane0`) of register 0x708 of the master/slave PHYs to 1'b1
 - Set bit [5] (`rxhs_clk_from_long_channel_if_rw of lane2 -> enable the hs clock to long channel of lane0`) of register 0x908 of the master/slave PHYs to 1'b1 (lane 2 if used)
 - Set bit [5] (`rxhs_clk_from_long_channel_if_rw of lane3 -> enable the hs clock to long channel of lane0`) of register 0xB08 of the master/slave PHYs to 1'b1 (lane 3 if used)
- Configure slave PHY clock lane to bypass long channel clock to DDR clock:
 - Set bit [3] (`rxclk_rxhs_ddr_clk_sel_if -> disable the DDR clock source`) of register 0x308 of the master PHY to 1'b0 (use short channel to generate DDR)
 - Set bit [3] (`rxclk_rxhs_ddr_clk_sel_if -> enable the DDR clock source`) of register 0x308 of the slave PHY to 1'b1 (use long channel to generate DDR)
- Override slave PHY clock lane enable (`DPHYRXCLK_CLL_demux` module):
 - Set bits [1:0] (`clk_en_lanes_tester,clk_en_lanes_bypass -> enable the clock for tester and bypass the clock lane`) of register 0x0E0 of the slave PHY to 2'b11
- Override slave PHY DDR clock lane enable (`DPHYHSRX_div124` module):
 - Set bit [1] (`rxclk_rxhs_ddr_clk_en_bypass -> enable DDR clock bypass`) of register 0x0E1 of the slave PHY to 1'b1

- Set bit [3](rxclk_rxhs_ddr_clk_en → enable the HS DDR clock) of register 0x307 of the slave PHY to 1'b1 (this configuration can be done when setting slave PHY clock lane to don't drive long channel clock)
- Turn off slave PHY LP-RX clock lane:
 - Set bit [7](lprxponcd_bypass_clk lane → enable the LP-CD bypass) of register 0x304 of the slave PHY to 1'b1
 - Set bits [4:0](lprxponulp_clk lane, lprxponulp_bypass_clk lane, lprxponlp_clk lane, lprxponlp_bypass_clk lane, lprxponcd_clk lane → disable the ulp mode, LP mode and enable the ULP mode bypass control) of register 0x305 of the slave PHY to 5'b01010
- Release reset for master PHY and wait until all its lanes (including clock lane) reach STOPSTATE
- Release reset for slave PHY and wait until all its data lanes reach STOPSTATE

Aggregation Power up/initialization sequence :

1. Assert reset for both DPHYs (shutdownz=rstz=1'b0)
2. Program relevant register of both DPHYs according to DPHY “standard” initialization seq (see DPHY-DB , set enableclk = 1'b0 of Slave DPHY)
3. Program relevant register of Master and Slave according to DPHY “aggregation” initialization seq (see aggregation app notes)
4. Release shutdownz (shutdownz=1'b1) for both DPHYs
5. Release rstz (rstz =1'b1) for Master DPHY
6. Wait until Master DPHY asserts stopstatedata_n and stopstateclkout indicating DPHY is receiving LP11
7. Release rstz (rstz =1'b1) for Slave DPHY
8. Wait until Slave DPHY asserts stopstatedata_n indicating DPHY is receiving LP11
9. Both DPHYs are in LP11 state and ready to receive data/deskew bursts

8.4 MIPI TX

8.4.1 Overview

MIPI TX implements the physical layer of TX universal lanes for the MIPI D-PHY interface. The PHY is stacked in a configuration with four data lanes and one clock lane. Lane 0 is unidirectional for HS support and is bidirectional for Escape modes (LPDT, Trigger, ULPS) and Turnaround and other lanes are unidirectional supporting HS and TX Escape modes (Trigger, ULPS). Clocklane is unidirectional supporting HS and ULPS.

The PHY can be used for DSI host applications.

The PHY also includes a clock multiplier PLL for high-speed (HS) clock generation. The PHY is targeted for high-speed data digital transmission between a host processor and display drivers supporting a maximum effective bit rate of 2.5 Gbps per lane. The assembled four-data-lane system enables up to 10 Gbps aggregate communication throughput, delivering the bandwidth needed for high-throughput data transfer.

There is an additional reduced-throughput, low-power data transfer mode in each differential pair, which reduces line count and minimizes cable wires and EMI shielding requirements.

8.4.2 Function description

8.4.2.1 Features

The PHY has the following features:

- Configuration clock-27MHz
- Lane operation ranging from 80 Mbps to 2.5 Gbps in forward direction
- Aggregate throughput up to 10 Gbps with four data lanes

- Attachable PLL clock multiplication unit for High-Speed operation
- 50% DDR output clock duty-cycle
- Maximum LP data rate supported of 10 Mbps
- PHY-Protocol Interface (PPI) for clock and data lanes
- Low-power Escape modes and Ultra Low Power state
- Programmable HS Amplitude Levels
- HS Slew Rate Control
- External Reference Resistor (200 Ω) required only for calibration process
- Internal Pattern Generator
- Intra-PHY testability
 - LP/ULPS External Loopback
 - I/O Continuity Test Support
- Inter-PHY testability
 - PHY-2-PHY High Speed Checker
 - PHY-2-PHY High Speed Loopback
 - PHY-2-PHY LP/ULPS External Loopback
- Analog Test Bus
- Internal ADC for digital voltage sensing
- Power Collapsing
- 1.8 V (+10%,-7%) analog high voltage supply operation (vph)
- 0.8 V (+10%,-7%) analog low-voltage analog supply operation (vp)
- 0.8 V (+10%,-7%) digital voltage supply operation (vdd)

8.4.2.2 Control Mode

Control mode is the default operating mode. After the initialization is completed (analog calibrations and PLL locking), the PHY remains in this default mode until a request is made either by the protocol layer, or directly through the sequence of low-power signals in the lanes in case of RX (only present in data lane 0). While in control mode, the transmitter side sets the LP-11 state in the lines - this is also called the Stop state.

The receiver side remains in control mode while receiving LP-11 in the lines. Any request must start from and end in Stop state. Following a request, a lane can leave control mode for either high-speed data transfer mode, Escape mode, or Ultra Low Power state.

8.4.2.3 High-Speed Data Transmission

Once the initialization sequence is completed, the PHY remains in control mode, which is the default operating mode, until some request appears. High-speed is one of the possible requests at this point. High-speed data transmission occurs in bursts. Only during these bursts the lanes are in high-speed mode. A high-speed burst must start from and return to a Stop state (control mode). A high-speed burst allows for the transmission of payload data by the data lanes. Inherent to such data transmission is the existence of a valid DDR clock in the clock lane. High-speed data bursts are independent for each lane, which means that each data lane can end a high-speed transmission independently of the state of the remaining data lanes. A burst contains the low-power initialization sequence, the high-speed data payload, and also the end of transmission sequence.

From the transmitter side, high-speed mode is entered when the corresponding txrequesths input is set high (assuming that the PHY is in Stop state), PLL is enabled and locked. A TX HS request is not allowed if these conditions are not met. This request is processed in a slightly different way for clock and data lanes. For a clock lane, processing the high-speed starts with the transmission of a low-power sequence (LP01 and LP00). Only after generating this sequence the low-power driver is disabled, and the high-speed driver enabled. After the time necessary to settle TCLK-ZERO, the transmission of the high-speed DDR clock starts. For a data lane, the high-speed entry sequence is the same as for clocklanes (LP01(TLPX) followed by LP00 (THS-prepare) and by THS-ZERO), but in addition, before the payload data

there is a leader sequence (THS-SYNC value 8'h1D) that allow for the receiver synchronization. The transmission of such sequence requires the existence of a valid high-speed clock signal in the clock lane. When the high-speed request input is disabled, each lane leaves the high-speed data transmission mode. It is important that a clock lane must be in high-speed mode during the complete high-speed data transmission state of all the lanes. The clock lane must enter the high-speed mode before a high-speed data transmission (Tclk-pre) begins and it must not leave this state before all the lanes finish their respective high-speed (Tclk-post). The operation sequence when leaving the high-speed mode is also slightly different for data and clock lanes. For a clock lane, the high-speed transmission always ends with a HS-0 state (TCLK-trail), followed by the disabling of the high-speed driver, and enabling of low-power driver. As for a data lane, the transmission ends with the differential state opposite to the last bit transmitted (THS-trail), followed by the disabling of the high-speed driver, and enabling of the low-power driver driving Stop state in the lines (LP11) and returning to Control Mode.

Figure 8-9 High Speed data reception

Escape mode is a special mode of operation that uses the data lanes to communicate using the low-power states at low-speed. The PHY supports this mode in both directions for data lane 0 and only in forward direction for lanes 1 to 3. A Data Lane enters the Escape mode through an Escape mode entry procedure (LP-11, LP-10, LP-00, LP-01, LP-00), if an LP-11 is detected before reaching LP-00 state, the entry is aborted and the receiver returns to the Stop state. Once the sequence is correctly completed, the transmitter sends an 8-bit command to indicate a requested action.

Each symbol consists of the following two parts:

Table 8-3 Possible Escape Mode Sequences for Data Lanes

Escape Mode Action	Entry Command Pattern(First Bit to Last Bit to be Transmitted)	Command Type
Unknown - 3	8'b01011101	trigger
Unknown - 4	8'b00100001	trigger
Unknown - 5	8'b10100000	trigger

Low-Power Data Transmission (LPDT)

In LPDT mode, the data can be transmitted by the protocol at low speed in Low-Power mode—high-speed drivers or receivers are off and low-power drivers or receivers are on. During LPDT, the protocol can pause by maintaining a Space state on the lines. The last state before exiting Escape mode is Mark-1—not considered as 1 bit—because it is followed by the Stop state.

Data shall be encoded on the lines with the same Spaced-One-Hot code as used for the Entry Commands. This encoding allows data to be self-clocked and does not rely on the Clock Lane.

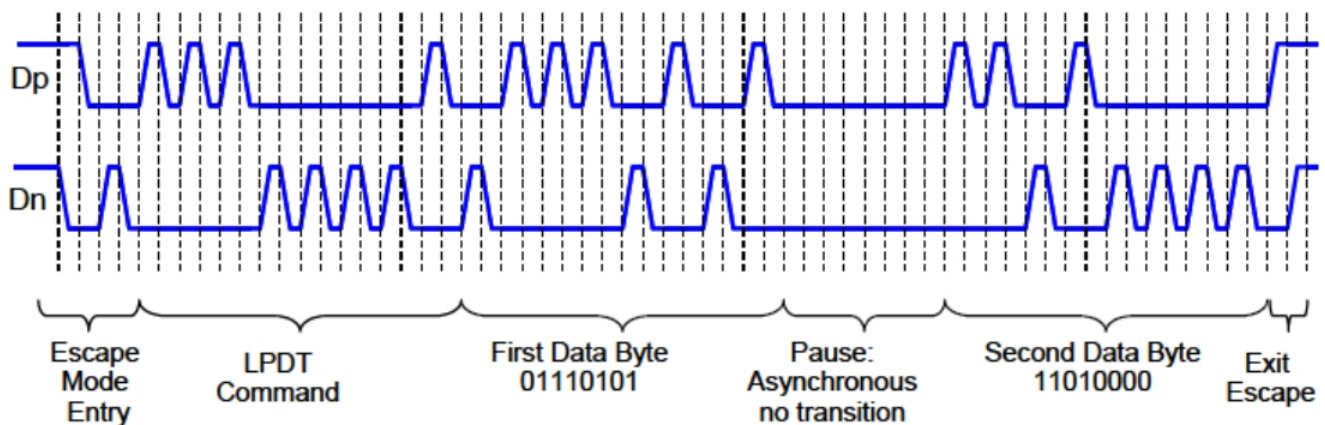


Figure 8-10 LPDT example

Remote Trigger

This mode allows the protocol to send a flag to the receiving side, on request of the transmitting side. If the Entry Command Pattern matches the Reset-Trigger Command a Trigger is flagged to the protocol at the receive side through the logical PPI. Any bit received after a Trigger Command but before the Lines go to Stop state shall be ignored. Therefore, dummy bytes can be concatenated in order to provide Clock information to the receive side.

Ultra Low Power State (ULPS)

This mode has the lowest power consumption (which can be further reduced when PHY automatically turns off PLL and part of AFE circuitry if all enabled lanes are in forward direction and enter ULPS), excluding the Shutdown mode. For data lanes, this mode is entered by sending an Ultra Lower Power state entry command, after the Escape mode entry command. During this mode, the lines are in the Space state (LP-00). Although the Clock lane does not support regular Escape mode, the Clock lane supports ULPS.

To exit ULPS, the transmitter side drives a Mark-1 followed by a Stop state. The Twakeup time must be ensured by the protocol layer, possibly using TXUlpsExit and TXRequestEsc/TXUlpsClk control signals. By default, the PLL clock multiplier and part of AFE circuitry is switched off if all enabled lanes are in ULPS. If one of the lanes leaves ULPS (using txulpsexit), the PHY will enable the PLL and turn AFE completely on. It is required that the txbyteclkhs clock connection to txbyteclksrc is active and not gated for the ULPS exit procedure to complete. All enabled lanes need to exit ULP state at the same time. The forcepll signal can be used to maintain the PLL enabled during ULPS or to disable it during ULPS. For this, forcepll needs to be asserted before the PHY enters in ULPS state.

If PLL is disabled during ULPS, and PLL configuration is being done through the PLL SoC signals you must apply updatepll pulse before exiting from ULPS.

During ULPS state, cfg_clk external clock can be gated for additional power savings as indicated in the following reference procedure:

1. Enter ULP in all enabled lanes (including clock lane) - Global ULP state.
2. Wait until ulpsactivenot_* =1'b0 in all lanes.
3. Wait 20 cfg_clk cycles.
4. Gate cfg_clk.
5. Ungate cfg_clk, at least 20 clock cycles prior to leaving ULP.
6. Leave ULP.
7. Wait until ulpsactivenot_* =1'b1 in all lanes.

Turnaround

The PHY allows the transmission direction of the data lane 0 to be swapped by means of a turnaround request. The sequence of a turnaround operation occurs in this way:

1. Transmitter side
 - a. Starts when turnrequest_0 input is set high on transmitter side, assuming the PHY is in Stop state
 - b. Transmitter sends specific low-power sequence—LP11, LP10, LP00, L10, L00
2. Receiver side
 - a. After receiving complete low-power sequence, receiver enables low-power driver
 - b. Transmits LP00, indicating change setting direction_0 output low
 - c. Overlapping LP00 transmitted by initial transmitter
 - d. Initial receiver side transmits LP10, followed by LP11, returning to Control mode

The turnaround procedure is the same for a Forward-to-Reverse direction or Reverse-to-Forward direction. However, in the Reverse-to-Forward direction, lane 0 only supports Escape Mode.

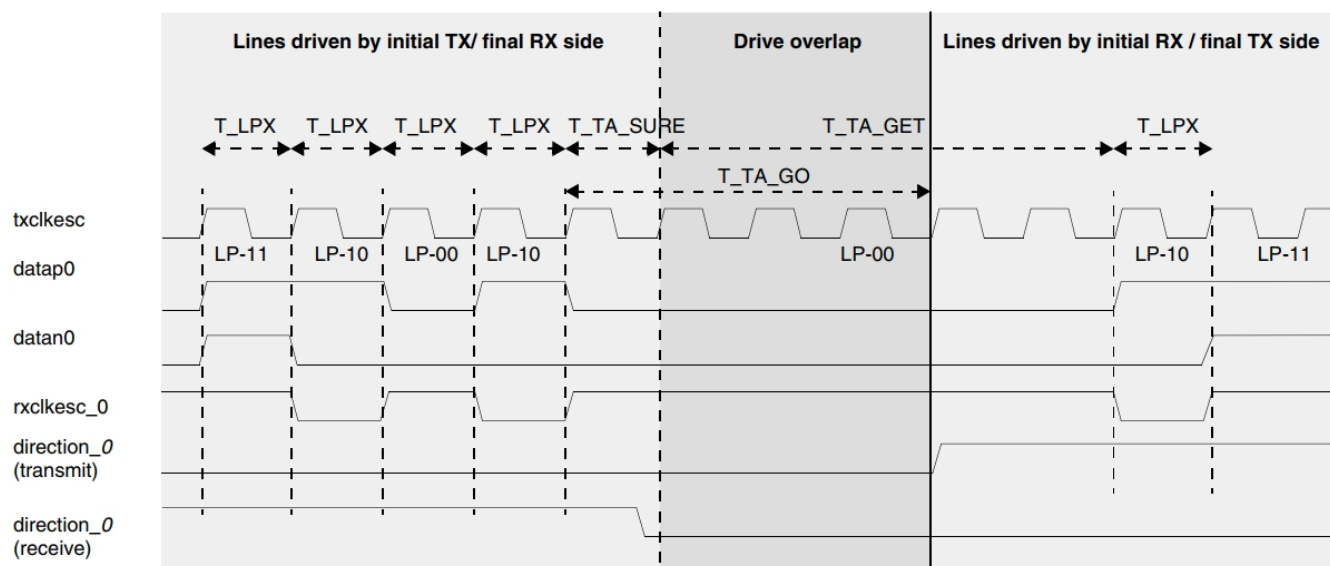


Figure 8-11 Turnaround Timing Diagram

8.4.2.5 Control and Test Modes

The PHY contains a set of control registers that are primarily used for the PHY configuration of its normal operating modes but that are also used for testing either under the scope of normal silicon characterization or debug activities and production testing in the ATE environment. These control registers are accessible through the PHY test

and control interface signals (testdin[7:0], testdout[7:0], testen, testclk and testclr). For correct configuration and testability of the PHY, ensure that this test and control interface is accessible through the link controller and/or by any other means (multiplex/demultiplex layer to chip I/Os, JTAG controller, and so on).

This section contains the timing diagrams for configuring a test code in the PHY. The standard procedure is two-folded; first the necessary test code is programmed and then the related test data words are fed to the tester's inputs.

To configure a test code, have the PHY in shutdown mode (shutdownz=0) and then reset it (rstz=0). This avoids the transient periods in the operation during re-configuration procedures. It is also recommended to apply a tester reset pulse (testclr = 1) before any test code configuration.

The test code programming is done using the following steps:

- 1. Set the desired test code
 - a. Ensure that testclk is set to high.
 - b. Place the 8-bit word corresponding to the test code in TESTDIN.
 - c. Set testen to high.
 - d. Set testclk to low.With the falling edge on testclk, the TESTDIN[7:0] signal content is latched internally as the current test code.
 - e. Set testen to low.
- 2. Enter the necessary test data
 - a. Set testclk to low, if not done already.
 - b. Place the 8-bit word corresponding to the required test data in TESTDIN.
 - c. Set testclk to high.Test data is programmed internally.
 - d. Repeat the steps to add more test data for the same test code.

Repeat the above procedure to program subsequent test codes. Additionally, a test reset procedure (testclr = 1) is only needed prior to the first programming operation or if you wish to reset the PHY's configuration to its default values and override any changes made meanwhile.

Here shows a generic timing diagram for test operation. After a test code is effectively programmed, TESTDOUT[7:0] asynchronously outputs relevant data for that specific test code, whether it is pure read-back data or other meaningful signals.

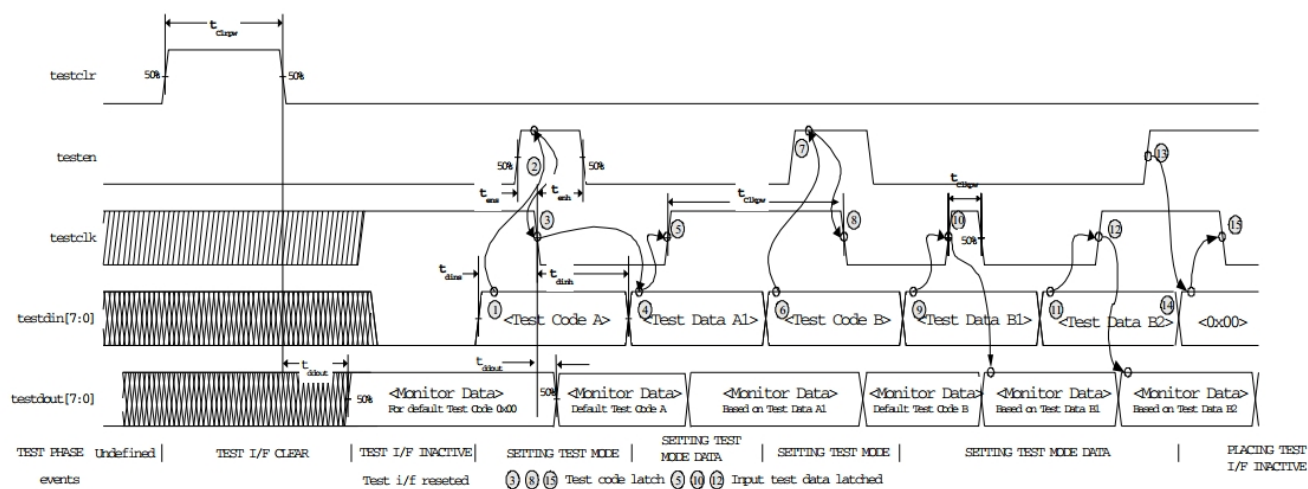


Figure 8-12 Testability Interface Timing Diagram

8.4.3 Interface description

Pins	I/O	Description

Pins	I/O	Description
rext_0	IO	200 ohm External termination resistor
rext_1	IO	200 ohm External termination resistor.
atb_0	IO	Analog test bus pin
Clkn0	IO	Negative D-PHY Differential Clock Line
Clkp0	IO	Positive D-PHY Differential Clock Line
Clkn1	IO	reserved
Clkp1	IO	reserved
datan0	IO	Negative D-PHY Differential Data Line 0
datap0	IO	Positive D-PHY Differential Data Line 0
datan1	IO	Negative D-PHY Differential Data Line 1
datap1	IO	Positive D-PHY Differential Data Line 1
datan2	IO	Negative D-PHY Differential Data Line 2
datap2	IO	Positive D-PHY Differential Data Line 2
datan3	IO	Negative D-PHY Differential Data Line 3
datap3	IO	Positive D-PHY Differential Data Line 3

8.4.4 Register summary

offset address	register	description
0x0000	TXDPHY_CFG0	configuration register 0 for Txdphy
0x0004	TXDPHY_CFG1	configuration register 1 for Txdphy
0x0008	TXDPHY_PLL_CFG	configuration register for PLL
0x000c	TXDPHY_PLL_OBS	observe register for PLL
0x0010	TXDPHY_PLL_CFG1	configuration register 1 for Txdphy
0x0014	TXDPHY_PLL_OBS1	observe register 1 for PLL
0x0018	DSI_CFG	configuration register for DSI
0x001c	TXDPHY_STA	status register for Txdphy

8.4.5 Register description

Name=TXDPHY_CFG0,offset=0x0

bits	Access Mode	Name	Description	Reset Value

bits	Access Mode	Name	Description	Reset Value
29:23	RO	cont_data_1	"phy1 Each pin of this output bus is used to verify the continuity from the chips pin all the way to the PHYs analog ports (package, bonding, die, etc). For more information about test modes, see the test modes section. ■ cont_data[0]: rext ■ cont_data[1]: clk_n ■ cont_data[2]: clk_p ■ cont_data[3]: datan0 ■ cont_data[4]: datap0 ■ cont_data[5]: datan1 ■ cont_data[6]: datap1"	0x0
22:16	RO	cont_data_0	"phy0 Each pin of this output bus is used to verify the continuity from the chips pin all the way to the PHYs analog ports (package, bonding, die, etc). For more information about test modes, see the test modes section. ■ cont_data[0]: rext ■ cont_data[1]: clk_n ■ cont_data[2]: clk_p ■ cont_data[3]: datan0 ■ cont_data[4]: datap0 ■ cont_data[5]: datan1 ■ cont_data[6]: datap1"	0x0
15	RW	cont_en_1	"This signal places the macro in IO continuity test mode. All other interface signals should be placed in their default values"	0x0
14	RW	cont_en_0	"This signal places the macro in IO continuity test mode. All other interface signals should be placed in their default values"	0x0
13:8	RW	cfgclkfreqrang_1	"phy1 System clock frequency configuration preset. Should be set according to the following equation: $\text{cfgclkfreqrange} = \text{round}[(\text{Fcfg\_clk(MHz)} - 17) * 4]$"	0x0

bits	Access Mode	Name	Description	Reset Value
7:2	RW	cfgclkfreqrang_0	"phy0 System clock frequency configuration preset. Should be set according to the following equation: $\text{cfgclkfreqrange} = \text{round}[(\text{Fcfg\_clk(MHz)} - 17) * 4]$ "	0x0
1	RW	basedir_1	"Configures the Base Direction for Lane (TxDphy 1) ■ basedir_0 = 1: Configures lane as RX upon startup of the PHY ■ basedir_0 = 0: Configures lane as TX"	0x0
0	RW	basedir_0	"Configures the Base Direction for Lane (TxDphy 0) ■ basedir_0 = 1: Configures lane as RX upon startup of the PHY ■ basedir_0 = 0: Configures lane as TX"	0x0

Name=TXDPHY_CFG1,offset=0x4, Write_enable=no

bits	Access Mode	Name	Description	Reset Value
23	RW	sta_phy_clr	clear phy1 err sta	0x0
22	RW	phy_test_sel	1:config phy1 0: config phy0	0x0
21	RW	turndisable_1	"Disable Turn Around This signal is used to prevent the bi-directional lane from processing a turn-around request in the lane interconnection. Note: This is useful to prevent a potential ""lock-up"" situation when a unidirectional lane module is connected to a bi-directional lane module."	0x0
20	RW	turndisable_0	"Disable Turn Around This signal is used to prevent the bi-directional lane from processing a turn-around request in the lane interconnection. Note: This is useful to prevent a potential ""lock-up"" situation when a unidirectional lane module is connected to a bi-directional lane module."	0x0
19:13	RW	hsfreqrange_1	"High-speed Frequency Range Selection. This signal should be set with the D-PHY in power-down and must not change during operation. This signal can be overridden through D-PHY test control registers."	0x0

bits	Access Mode	Name	Description	Reset Value
12:6	RW	hsfreqrange_0	"High-speed Frequency Range Selection. This signal should be set with the D-PHY in power-down and must not change during operation. This signal can be overridden through D-PHY test control registers."	0x0
5	RW	forcetxstopmode_3	"lane 3 Force Lane Module into Transmit Mode and Generate Stop State This signal allows the protocol layer to force a bi-directional lane module into transmit mode and Stop state following an error indication (for example, Expired Timeout). When this signal is high, the lane module immediately transitions into transmit mode and the module state machine is forced into the Stop state"	0x0
4	RW	forcetxstopmode_2	"lane 2 Force Lane Module into Transmit Mode and Generate Stop State This signal allows the protocol layer to force a bi-directional lane module into transmit mode and Stop state following an error indication (for example, Expired Timeout). When this signal is high, the lane module immediately transitions into transmit mode and the module state machine is forced into the Stop state"	0x0
3	RW	forcetxstopmode_1	"lane 1 Force Lane Module into Transmit Mode and Generate Stop State This signal allows the protocol layer to force a bi-directional lane module into transmit mode and Stop state following an error indication (for example, Expired Timeout). When this signal is high, the lane module immediately transitions into transmit mode and the module state machine is forced into the Stop state"	0x0

bits	Access Mode	Name	Description	Reset Value
2	RW	forcetxstopmode_0	"lane 0 Force Lane Module into Transmit Mode and Generate Stop State This signal allows the protocol layer to force a bi-directional lane module into transmit mode and Stop state following an error indication (for example, Expired Timeout). When this signal is high, the lane module immediately transitions into transmit mode and the module state machine is forced into the Stop state"	0x0
1	RW	forcerxmode_1	"Force Lane Module into Receive Mode/Wait for Stop State This signal allows the protocol layer to initialize a lane module or force a bi-directional lane module into receive mode. This signal is used to solve a contention situation. When this signal is high, the lane module immediately transitions into receive control mode and waits for a Stop state to appear on the Lane interconnect"	0x0
0	RW	forcerxmode_0	"Force Lane Module into Receive Mode/Wait for Stop State This signal allows the protocol layer to initialize a lane module or force a bi-directional lane module into receive mode. This signal is used to solve a contention situation. When this signal is high, the lane module immediately transitions into receive control mode and waits for a Stop state to appear on the Lane interconnect"	0x0

Name=TXDPHY_PLL_CFG,offset=0x8, Write_enable=no

bits	Access Mode	Name	Description	Reset Value
30:27	RW	n	"Control of the input frequency division ratio N (1 to 16) for SoC direct PLL control. This signal can control the PLL if pll_shadow_control = 1'b0"	0x0

bits	Access Mode	Name	Description	Reset Value
26:17	RW	m	"Control of the feedback multiplication ratio M (40 to 625) for SoC direct PLL control. This signal can control the PLL if pll_shadow_control = 1'b0"	0x0
16:11	RW	int_cntrl	"Integral Charge Pump control for SoC direct PLL control. This signal can control the PLL if pll_shadow_control = 1'b0 Power down value=000000"	0x0
10	RW	gp_clk_en	Enable signal for clkout_gp clock	0x0
9:8	RW	gmp_cntrl	"Controls the effective loop-filter resistance (=1/gmp) to increase/decrease MPLL bandwidth and compensate for change in divider modulus (n_cntrl). For SoC direct PLL control. This signal can control the PLL if pll_shadow_control = 1'b0"	0x0
7	RW	force_lock	"Force lock to SOC: ■ 0 - according to lock detector (default) ■ 1 - lock indication forced"	0x0
6:0	RW	cpbias_cntrl	"Charge Pump bias control, for SoC direct PLL control. This signal can control the PLL if pll_shadow_control = 1'b0"	0x0

Name=TXDPHY_PLL_OBS,offset=0xc, Write_enable=no

bits	Access Mode	Name	Description	Reset Value
30:27	RO	n_obs	"Output of multiplexer controlled by pll_shadow_control signal. This output signal represents the control signal provided by PHY test control registers or SoC control depending on pll_shadow_control selection. Used for SOC observability of the n_obs that is controlling the PLL."	0x0
26:17	RO	m_obs	"Output of multiplexer controlled by pll_shadow_control signal. This output signal represents the control signal provided by PHY test control registers or SoC control depending on pll_shadow_control selection. Used for SOC observability of the m_obs that is controlling the PLL"	0x0

bits	Access Mode	Name	Description	Reset Value
16	RO	lock	"PLL Lock Signal When set, it signals that the PLL acquired lock with input refclk. This signal can be overridden through D-PHY test control registers"	0x0
15:10	RO	int_cntrl_obs	"Output of multiplexer controlled by pll_shadow_control signal. This output signal represents the control signal provided by PHY test control registers or SoC control depending on pll_shadow_control selection. Used for SOC observability of the int_cntrl_obs that is controlling the PLL."	0x0
9:8	RO	gmp_cntrl_obs	"Output of multiplexer controlled by pll_shadow_control signal. This output signal represents the control signal provided by PHY test control registers or SoC control depending on pll_shadow_control selection. Used for SOC observability of the gmp_cntrl_obs that is controlling the PLL."	0x0
7:1	RO	cpbias_cntrl_obs	"Output of multiplexer controlled by pll_shadow_control signal. This output signal represents the control signal provided by PHY test control registers or SoC control depending on pll_shadow_control selection. Used for SOC observability of the cpbias_cntrl_obs that is controlling the PLL"	0x0
0	RO	clkout_gp	Auxiliary, general purpose clock $F=VCO/2$	0x0

Name=TXDPHY_PLL_CFG1,offset=0x10, Write_enable=no

bits	Access Mode	Name	Description	Reset Value
16:11	RW	vco_cntrl	"VCO operating range for SoC direct PLL control, needs to be set appropriately to frequency of operation. This signal can control the PLL if pll_shadow_control = 1'b0"	0x0

bits	Access Mode	Name	Description	Reset Value
10	RW	updatepll	"Control for PLL operation frequency updated. When exercised, PLL SoC control signals are latched to PLL shadow_registers to set PLL operating frequency"	0x0
9	RW	shadow_clear	"Shadow registers clear. Rise edge sensitive."	0x0
8:3	RW	prop_cntrl	"Proportional Charge Pump control for SoC direct PLL control. This signal can control the PLL if pll_shadow_control = 1'b0 Power down value=000000"	0x0
2	RW	pll_shadow_control	"Selection of PLL configuration mechanism. PLL can be configured through D-PHY or through SoC control and shadow control registers: ■ 0: SoC control and shadow registers ■ 1: D-PHY contro"	0x0
1:0	RW	clkssel	"Control of PLL clock output selection. ■ 00 - Clocks stopped ■ 01 - Clock generation ■ 10 - Buffered clkext ■ 11 - Forbidden"	0x0

Name=TXDPHY_PLL_OBS1,offset=0x14, Write_enable=no

bits	Access Mode	Name	Description	Reset Value
12:7	RO	vco_cntrl_obs	"Output of multiplexer controlled by pll_shadow_control signal. This output signal represents the control signal provided by PHY test control registers or SoC control depending on pll_shadow_control selection. Used for SOC observability of the vco_cntrl_obs that is controlling the PLL."	0x0
6:1	RO	prop_cntrl_obs	"Output of multiplexer controlled by pll_shadow_control signal. This output signal represents the control signal provided by PHY test control registers or SoC control depending on pll_shadow_control selection. Used for SOC observability of the prop_cntrl_obs that is controlling the PLL"	0x0

bits	Access Mode	Name	Description	Reset Value
0	RO	pll_shadow_control_obs	"Output of multiplexer controlled by pll_shadow_control signal. This output signal represents the control signal provided by PHY test control registers or SoC control depending on pll_shadow_control selection. Used for SOC observability of the pll_shadow_control_obs that is controlling the PLL."	0x0

Name=DSI_CFG,offset=0x18, Write_enable=no

bits	Access Mode	Name	Description	Reset Value
2	RW	dpiupdatecfg	Control Signal. This signal is used to indicate that the next frame will have new video configuration	0x0
1	RW	dpicolorm	Control Signal. It is used to switch between normal color and reduced color mode	0x0
0	RW	dpishutdn	color mode.	0x0

Name=TXDPHY_STA,offset=0x1c, Write_enable=no

bits	Access Mode	Name	Description	Reset Value
4	RO	errsyncesc_0_phy1		0x0
3	RO	erresc_0_phy1		0x0
2	RO	errcontrol_0_phy1		0x0
1	RO	errcontentionlp1_0_phy1		0x0
0	RO	errcontentionlp0_0_phy1		0x0

8.4.6 Application description

8.4.6.1 Start-Up sequence for 2lane

Before D-PHY power-up, the D-PHY needs to be properly configured for the desired operation. This needs to be done with the PHY in shutdown mode and it is assumed that such configurations are static or at least are stable prior to leaving the Shutdown mode.

Before D-PHY power, set mppll_prg[0]=1 to set PLL regulator in bypass. This control signal can be configured to mppll_prg[0]=0 when in ULPS or power-down for reduced power consumption.

The D-PHY operating frequency and input reference clock frequency need to be configured through hsfreqrange[6:0] and cfgclkfreqrange[5:0].

■ IP operating frequency: hsfreqrange[6:0]

■ Input reference clock frequency: `cfgclkfreqrange[5:0]`

Regarding input system clock `cfg_clk` frequency, `cfgclkfreqrange[5:0]` needs to be configured according to the following formula:

$$\text{cfgclkfreqrange}[5:0] = \text{round}[(F_{\text{cfg\_clk}}(\text{MHz}) - 17) * 4]$$

To configure the bit rate operation you should directly configure `hsfreqrange` for the desired frequency range or set the register `hsfreqrange`, “Test Code: 0x44”, with the proper code. You can only configure `hsfreqrange` before the D-PHY power-up. You can not change `hsfreqrange` setting during active operation.

The PLL needs always to be properly configured for the desired operating frequency before D-PHY power-up. This configuration can be done through the D-PHY test and control interface or through the PLL SoC shadow registers interface.

The PHY consists of four data lanes, but applications can use four or lesser number data lanes. In such cases, you are granted access to individual enabling signals (`enable_n`) that control which lanes should be used and evolve through all the necessary initialization steps. It is not allowed for the enable signals to be changed during operation. They should be set in power-down. When the PHY is operating in HS mode data lane 0 must be enabled.

After the PLL is configured, the D-PHY is responsible for the power-up and power-down of the PLL within its power sequence logic; this control is transparent to you. After `rstz` and `shutdownz` are released, the D-PHY starts its power-up sequence and, with it enabled, the PLL will lock to the configured operation.

It is assumed that all configurations are static or at least are stable prior to leaving the Shutdown mode. This includes not only configurations through D-PHY SoC interface but all configurations done through the D-PHY test and control interface.

After the reset signals (`rstz` and `shutdownz`) are released, the PHY begins an initialization sequence to allow correct operation. The sequence of the release of signals is not critical, but it is recommended that `shutdownz` precede `rstz`. It is assumed that the `cfg_clk` signal is available and stable by that time. If there are no test or configuration operations to be performed, the `testclr` signal can be kept at logic high level.

Otherwise, the `testclr` must be de-asserted to bring the control logic out of reset and allow for the necessary configuration steps through the control interface.

The D-PHY specification has many timing intervals which have to be followed to ensure proper operation.

The initialization sequence ends when stopstate LP11 is driven in the lanes and the PHY reaches Control Mode state.

Table 8-4 Frequency Ranges and Defaults

Range (Mbps)	Default Bit Rate (Mbps)	hsfreqrange[6:0]
80 - 97.125	80	0000000
80 - 107.625	90	0010000
83.125 - 118.125	100	0100000
92.625 - 128.625	110	0110000
102.125 - 139.125	120	0000001
111.625 - 149.625	130	0010001
121.125 - 160.125	140	0100001
130.625 - 170.625	150	0110001
140.125 - 181.125	160	0000010

Range (Mbps)	Default Bit Rate (Mbps)	hsfreqrange[6:0]
149.625 - 191.625	170	0010010
159.125 - 202.125	180	0100010
168.625 - 212.625	190	0110010
182.875 - 228.375	205	0000011
197.125 - 244.125	220	0010011
211.375 - 259.875	235	0100011
225.625 - 275.625	250	0110011
249.375 - 301.875	275	0000100
273.125 - 328.125	300	0010100
296.875 - 354.375	325	0100101
320.625 - 380.625	350	0110101
368.125 - 433.125	400	0000101
415.625 - 485.625	450	0010110
463.125 - 538.125	500	0100110
510.625 - 590.625	550	0110111
558.125 - 643.125	600	0000111
605.625 - 695.625	650	0011000
653.125 - 748.125	700	0101000
700.625 - 800.625	750	0111001
748.125 - 853.125	800	0001001
795.625 - 905.625	850	0011001
843.125 - 958.125	900	0101001
890.625 - 1010.625	950	0111010
938.125 - 1063.125	1000	0001010
985.625 - 1115.625	1050	0011010
1033.125 - 1168.125	1100	0101010
1080.625 - 1220.625	1150	0111011
1128.125 - 1273.125	1200	0001011
1175.625 - 1325.625	1250	0011011
1223.125 - 1378.125	1300	0101011
1270.625 - 1430.625	1350	0111100
1318.125 - 1483.125	1400	0001100
1365.625 - 1535.625	1450	0011100
1413.125 - 1588.125	1500	0101100

Range (Mbps)	Default Bit Rate (Mbps)	hsfreqrange[6:0]
1460.625 - 1640.625	1550	0111101
1508.125 - 1693.125	1600	0001101
1555.625 - 1745.625	1650	0011101
1603.125 - 1798.125	1700	0101110
1650.625 - 1850.625	1750	0111110
1698.125 - 1903.125	1800	0001110
1745.625 - 1955.625	1850	0011110
1793.125 - 2008.125	1900	0101111
1840.625 - 2060.625	1950	0111111
1888.125 - 2113.125	2000	0001111
1935.625 - 2165.625	2050	1000000
1983.125 - 2218.125	2100	1000001
2030.625 - 2270.625	2150	1000010
2078.125 - 2323.125	2200	1000011
2125.625 - 2375.625	2250	1000100
2173.125 - 2428.125	2300	1000101
2220.625 - 2480.625	2350	1000110
2268.125 - 2500	2400	1000111
2315.625 - 2500	2450	1001000
2363.125 - 2500	2500	1001001

Power-up configuration examples are provided below. If the test and control interface is to be used before DPHY power-up for any additional configuration or debug purposes, it must be exercised after the testclr pulse and before shutdownz and rstz de-assertion which will start the D-PHY power-up sequence.

While shutdownz and rstz are de-asserted, the DPHY remains in POWERDWN state. If the PHY properly reached the end of its startup sequence, it will be in IDLE state.

Case 1 (operating rate 1 Gbps; cfg_clk=27Mhz; slew rate calibration enabled):

1. Set rstz = 1'b0.
2. Set shutdownz= 1'b0.
3. Set testclr = 1'b1.
4. Wait for 15 ns.
5. Set testclr to low.
6. Set hsfreqrange[6:0] = 7'b0001010.
7. Refer to table "Slew rate vs DDL oscillation target" on page 110 and configure test control registers with appropriate values for the specified rise/fall time.
8. Set bits [1:0] of test control register with address 0xA3 to 2'b01 to enable slew rate calibration.
9. Set mpll_prg[0]=1'b1 through test control register 0x1F.
10. Configure register 0x4a to set prg_on_lane0 (bit 6) to 1'b1

11. Set `cfgclkfreqrange[5:0] = round[(Fcfg_clk(MHz)-17)*4] = 6'b101000`.
12. Apply `cfg_clk` signal with 27Mhz frequency.
13. Configure PLL operating frequency through D-PHY test control registers or through PLL SoC shadow registers interface (see "Initialization" on page 55 for additional details).
14. Set `basedir_0 = 1'b0`.
15. Set all requests inputs to zero.
16. Wait for 15 ns.
17. Set `enable_n` and `enableclk=1'b1`.
18. Wait 5ns.
19. Set `shutdownz=1'b1`.
20. Wait 5ns.
21. Set `rstz=1'b1`.
22. Wait until `stopstatedata_n` and `stopstateclk` outputs are asserted indicating PHY is driving LP11 in enabled datalanes and clocklane.

8.4.6.2 Start-Up sequence for 4lane

The procedure to setup and operate both PHYs is the following:

- Turn on reset for both macros (`shutdownz=rstz=1'b0`)
- Release reset for 1st RX PHY (`shutdownz=rstz=1'b1`) and wait until all its lanes reach STOPSTATE
- Release reset for 2nd RX PHY (`shutdownz=rstz=1'b1`) and wait until all its lanes reach STOPSTATE

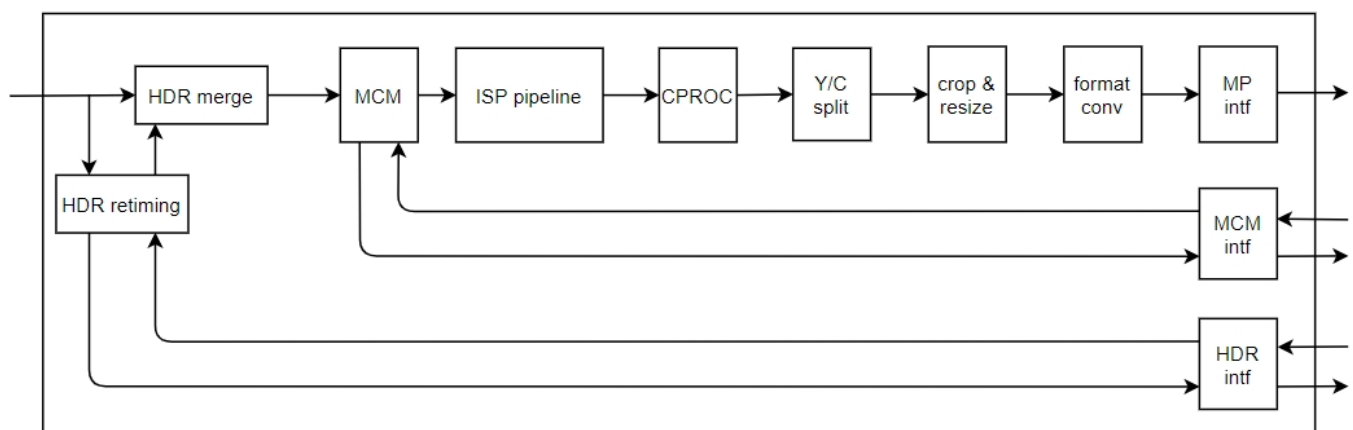
9 ISP

9.1 overview

The image signal processor (ISP) module supports standard sensor image data processing. The ISP module provides basic functions such as automatic white balance, automatic exposure, demosaic, defect pixel correction, and lens shading correction as well as advanced functions such as wide dynamic range, Multi-exposure HDR, and denoising.

9.2 Block diagram

Figure 9-1-1 isp core block diagram



9.3 Function description

9.3.1 Features

- Test Pattern Generator (TPG)
- Black level Compensation
- Sensor Linear Correction
- Defect Pixel Cluster Correction (DPCC)
- Green Equalization
- Lens Shade Correction (De-Vignetting)
- Digital Gain
- 2-Stage Adaptive Noise Filter (Noise reduction, Sharpness, Blurring)
- Enhanced Color Interpolation (Bayer De-mosaic filter)
- Chromatic Aberration Correction (CAC)
- Color Correction (Xtalk) Matrix (CCM)
- Color Space Conversion (CSC)
- Programmable gamma correction for sensor adaptation and display correction
- Auto Focus measurement (AF)
- Auto White Balance measurement (AWB)
- Auto Exposure measurement (AE)
- Histogram Calculation
- Anti-flicker
- Cropping of the output picture (to crop interpolation artifacts), also used for windowing
- Color Processing: Contrast, Saturation, Brightness, Hue (CPROC)
- Sharpen/Blur Filter
- Flash light control
- Mechanical shutter control
- Image size: horizontal size multiple of 8 and vertical size multiple of 4
- WDR in single wide dynamic frame
- Multi-exposure HDR (Native/Built-in HDR interface, 12~16-bit HDR input). The performance is up to 2960x1666@60fps for DOL2 and 1920x1080@90fps for DOL3 HDR.
- CPD, compand function to support native HDR sensor which performs HDR merging in sensor
- Variable sensor interface for RGB-Bayer Sensors, DOL or stagger HDR sensors
- 20-bit HDR Processing, includes multi exposure stitching and local tone mapping
- Advanced 2DNR spatial noise reduction
- Advanced 3DNR: spatial and temporal noise reduction
- Enhanced Chroma Noise Reduction (CNR)
- Video Stabilization (VSM)
- Multi-Context Management to support three sensors within a single ISP Core (MCM)
- Tile mechanism for 4K resolution
- Configurable 10/12-bit YUV output
- RGB-IR 4x4 pattern

9.3.2 module functions

9.3.2.1 Test Pattern Generator

Internal TPG to facilitate the ISP test

- 3x3 color blocks
- Color bar

- Grey bar
- Highlighted Grid
- Random data

9.3.2.2 Bad Pixel Replacement

- State-of-the-art edge-sensitive on-the-fly algorithm
- Additional table based mode
- Uses surrounding neighbours for replacement
- Detection of hot, dead and floating pixels
- Repair of defect pixel clusters
- On-the-fly algorithm for support of ultra low cost sensors

9.3.2.3 Lens Shade Compensation

- Lens shade compensation Block
 - Processing module for 2-D piece-wise bi-linear gain adjustment
 - Optimum compensation parameter set is determined once for each sensor module using VeriSilicon calibration tool
- Adaptive Lens Shade Compensation (SW)
 - Illuminant-adaptively
- Gain-adaptive
 - Compensating for color shading and vignetting
 - Vignetting compensation flows more control on noise in lowlight conditions when analog sensor gain is high

9.3.2.4 De-noising for Low Light Image Capturing

- Applications
 - Efficient reduction of noise for image capturing below 10 lux
 - Digital image stabilization (DIS) by increasing digital gain for shorter shutter speed
- Features
 - Automatic noise detection and selection of filter settings
 - State-of-the-art edge preserving filtering scheme

9.3.2.5 Bayer De-mosaicing and Filtering

- Generate RGB pixel from Bayer pattern RGB data from camera by interpolation
 - Adaptive filter on 5x5 pixel matrix
 - Edge sensitive to prevent artifacts on edges
- Denoising / Sharpening / Blurring Function
 - Denoising with edge preserving filtering
 - Programmable blurring and sharpening filter

9.3.2.6 Chromatic Aberration

- Low-cost lens show visible chromatic aberration
 - Visible color fringes at black / white edges
 - Error increases with distance to image center
- Chromatic aberration correction (CAC) significantly reduces the visible artefacts
- Benefit: Good image quality even with low cost lens

9.3.2.7 Auto Exposure (AE) Control

- AE measurement functions
 - 5x5 matrix measurement and histogram measurement
- AEC Features
 - Scene evaluation for front light / backlight detection
 - Flicker correction for 50/60 Hz environments
 - Support of manual exposure compensation

9.3.2.8 Autowhite Balancing (AWB)

- Measurement of white point of image
- Estimation of a mixture of illuminant sources
- Model-based calculation of probability for the used light source
- Automatic computation of an optimal color profile (aCC)
- Automatic computation of optimal cross-talk removal function (aLSC)
- Fast convergence speed and intelligent temporal damping behavior
- User selectable color profiles (e.g. sunny, cloudy, tungsten, fluorescent, etc.)

9.3.2.9 Autofocus (AF)

- Autofocus Measurement Hardware
 - Focus measurement block
 - 3 measurement windows
- System Control Software
 - Analysis of HW measurements
 - Optimization of image sharpness
- Control of lens system

9.3.2.10 Wide Dynamic Range Processing

- Capturing of high contrast scenes usually leads to
 - Highlight clipping or
 - Lowlight clipping
- Wide dynamic range processing prevents loss of detail
 - Automatic checking for image details in dark and bright areas
 - Scene dependent sensitivity adaptation
 - Improvement of color reproduction

9.3.2.11 Color Processing Features

The following features are provided by the color processing (CPROC) sub module:

- Color depth: 10 bits for each color component
- Operates on 4:2:2 YCbCr format
- Full range and ITU-R BT.601 range are independently programmable for input and output. (Gain adjustment must be done with contrast and saturation settings)
- Contrast adjustment in the range [0..1.992x] (256 steps)
- Brightness adjustment in the range [-512..+508] (256 steps)
- Saturation adjustment in the range [0..1.992x] (256 steps)
- Hue adjustment in the range [-90..+87.788] degrees (256 steps, 8 bit + sign bit sine/cosine coefficients)

9.3.2.12 Resize and Format Conversion Features

- Supports 4:2:2 and 4:2:0 YCbYCr input format

- Different output formats (4:2:2, 4:2:0, 4:4:4) are possible by choosing different scaling factors for luminance and chrominance components and 8-bit and 10-bit YUV color depth.
- Support for RGB888 output
- Support for co-sited and non co-sited output formats by programmable phase offsets.
- Separate bypassing of each sub module possible
- Input and output pictures up to 8192x8192 pixels
- Separate scaling in horizontal and vertical directions, and for chrominance and luminance components
- Lookup table for programmable up-scaling characteristics
- Shadow registers for on-the-fly reprogramming of control registers
- Cropping of the Output Picture

9.3.2.13 Output Data Format Support

- Main Path Format Support
 - YCbCr 4:2:2 or 4:2:0 either planar, semi-planar or interleaved, configurable 8-bit ,10-bit or 12-bit color depth.
 - RGB888 format
 - 8 to 12 bit RAW data with data end signaling.
- Self Path Format Support
 - YCbCr 4:4:4, 4:2:2 or 4:2:0 either planar or semi-planar, configurable 8-bit ,10-bit or 12-bit color depth.
 - RGB888 format
 - RGB and YCbCr 4:2:2 data can be interleaved.

10 3D Processing Engine

10.1 dpu

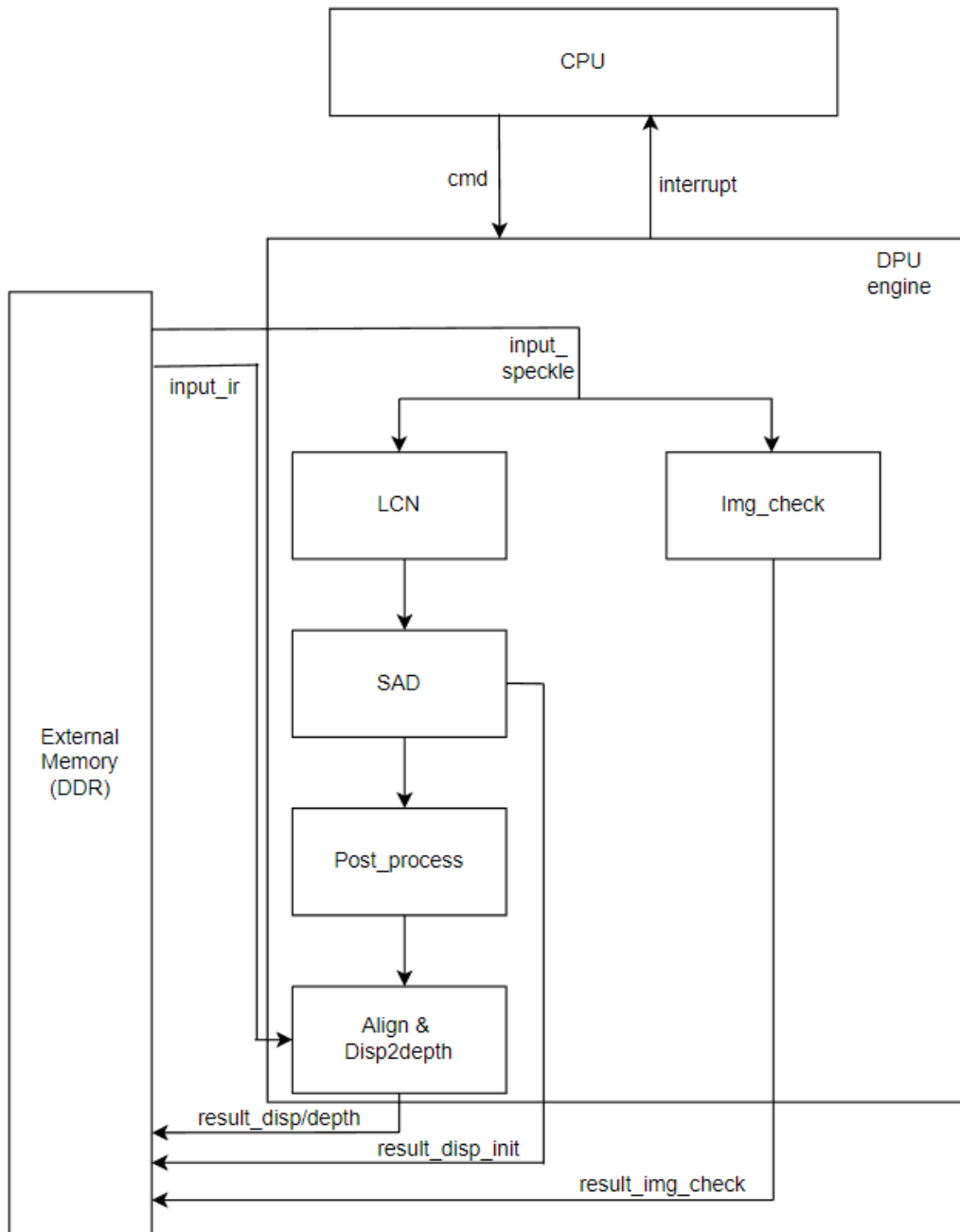
10.1.1 Overview

The DPU is mainly responsible for the depth calculation of 3D structured light, providing depth information for 3D face recognition. It reads the ISP-processed speckle image and infrared image from the DDR, and performs multi-step pipeline calculations under the control of the CPU, and finally outputs the calculated depth image/infrared image result to the DDR. Among them, the calculations to be performed by the DPU include image quality detection (Img_check), local contrast normalization (LCN: Local_contrast_normalization), stereo matching (SAD: sum of absolute difference), post-processing (Post_process), three-image alignment (Align), disparity to depth (Disptodepth). Only the Img_check operator performs calculations independently, and the rest of the operators are executed in a pipeline, that is, the output result of the former operator is used as the input data of the latter operator. The resolutions supported by the DPU are as follows:

1. The maximum resolution of the vertical image is 1920*1080, and the maximum resolution of the horizontal image is 1080*1440
2. Typical performance is 1280*800@30fps, 1280*1080@15fps, 1920*1080@9fps

10.1.2 Block Diagram

[Figure 10-1-1](#) DPU overall architecture diagram



10.1.3 Features

The DPU includes the following features:

- Both read and write AXI bus support crop
- Support standard axi-4 protocol, data bit width 64bit, address bit width 32bit, burst_length up to 16, read and write outstanding up to 8
- Support standard apb3.0 protocol, data bit width 32bit, address bit width 32bit
- **Img_check**: The precision of the input speckle image is int8, the precision of the output calculation result is int1, and it supports output after downsampling
- **LCN**: The precision of the input speckle image is int8, and the precision of the output calculation result is int12, which supports normalized calculation of the source image
- **SAD**: Input the normalized speckle image/reference image, the accuracy is int12, the output row and column parallax accuracy is int16, support Init_SAD single/double layer enable, support SAD double layer enable, and support the first layer input of Init_SAD/SAD is involved in the calculation after 1/2 downsampling

- Post_proc: The precision of the input column parallax is int16, the precision of the output calculation result is int1, supports Post_proc enable, and supports Post_proc inner mask denoising enable
- Align: The precision of input column parallax is int16, the precision of mask is int1, the precision of output depth/disparity map is int16, the precision of input/output infrared map is int8, supports height block processing parallax/infrared alignment, supports parallax/infrared Alignment enable, support Align inner parallax denoising enable, support parallax/infrared up and down sampling enable, support Disp2depth enable
- Support normal calculation end interrupt, timeout interrupt, instruction exception interrupt, configuration instruction overflow interrupt, AXI write response exception interrupt
- Support independent soft reset, interrupt clear operation

10.2 GDMA

10.2.1 Overview

GDMA (Graphic Direct Memory Access) controller is used for the transfer of graphic data and the rotation of images , it has the capability of high-speed data transmission without using the CPU, it supports data transfer within DDR or SRAM, or between DDR and SRAM.

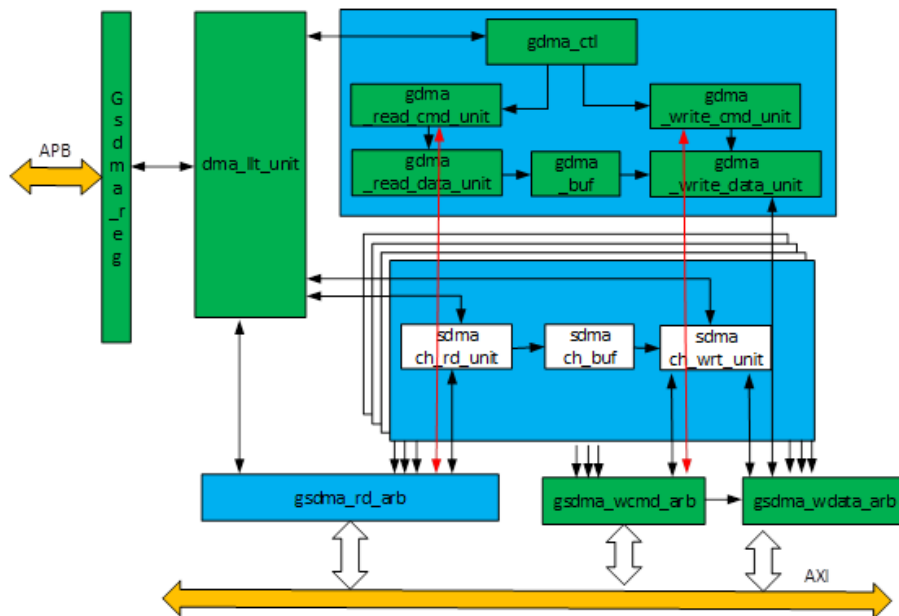
10.2.2 Features

The GDMA has the following features:

- Provides one physical DMA channel
- The axi data width is 64-bit
- Supports configurable axi transaction outstanding numbers: the max outstanding number is 8
- Supports data transfer with the linked list
- Supports configurable weight-based channel priority arbitration
- Supports lowpower mode : each channel supports auto clock gate
- Supports image 90°, 180°, 270° rotation and 0° copy
- Supports image x-mirror, y-mirror, x-mirror + y-mirror
- Supports configurable image row stride
- The maximum image resolution supports 64Kx64K
- The data width of the image pixel supports 8/16/24/32bits
- Max Burst length can be set to 8 or 4 for non-24bit width pixel, and 12 for 24bit pixel
- Alignments:
 - The base address and stride of 8bit pixel both need 1byte alignment
 - The base address and stride of 16bit pixel both need 2byte alignment
 - The base address and stride of 32bit pixel both need 4byte alignment
 - The base address and stride of the 24bit pixel 0°copy mode both need 8byte alignment
 - The base address and stride of the 24bit pixel non-0° copy mode both need 24byte alignment, and the number of the pixels in a row must be the multiple of 8

10.2.3 Block Diagram

Figure 10-2-1 Gsdma Block Diagram



The sdma and gdma share one

dma architecture, names as gsdma.the gsdma has 1 master data transfer interface and 1 slave config interface, the master interface is axi4 and the slave interface is apb2.0. Both the sdma and gdma access to the master interface through the arbiter and they share the base address of the register space.

10.2.4 Functional Description

channel priority arbitration since sdma and gdma are based on the gsdma architecture, they both follow the same arbitration mechanism.A weight-based channel arbitration mechanism is used here. The arbitration for reading/writing the master port are both implemented based on this method. The mechanism is described below The GSDMA_WEIGHT register can be divided into eight 3-bit fields, and each field corresponds to an arbitration code. These eight arbitration codes are connected in series base on a circular manner, and a shift is performed each time an arbitration action is completed. Every 8 consecutive requests for arbitration actions constitute an arbitration cycle (there may be only one channel in a certain arbitration requesting to use the AXI master port, but this situation is still regarded as a valid arbitration action), and the current status is updated every time an arbitration action is completed. The arbitration code has 5 values: 0, 1, 2, 3, 4. When the arbitration code is '0', the channel priority is defined as: ch0 > ch1 > ch2 > ch3>gdma; When the arbitration code is '1', the channel priority is defined as: ch1 > ch2 > ch3 > gdma>ch0; When the arbitration code is '2', the channel priority is defined as: ch2 > ch3 >gdma> ch0 > ch1; When the arbitration code is '3', the channel priority is defined as: ch3 >gdma> ch0 > ch1 > ch2; When the arbitration code is '4', the channel priority is defined as: gdma > ch0 > ch1 > ch2 > ch3. Assuming that the value of the GSDMA_WEIGHT register is configured as 0_1_2_4_4_4_3_0, these 8 values correspond to 8 arbitration actions. From this configuration value, it can be seen that channel 4 (gdma) has the largest weight, followed by channel 0, and finally channels 1, 2, and 3.

Linked List Transfer The GDMA linked list takes 24bytes as a node. The content of the linked list is placed in the memory space and The register GDMA_LLT_SADDR is used to specify the entry address of the starting linked list. The linked list address needs to be 8bytes aligned. When next_lli_addr is 0, the linked list ends. The structure of the GDMA linked list is as follows:

Table 10.2.1 gdma Linked List

Bits	Description	Remarks
1:0	rotation mode	0: 0° 1: 90° 2: 180° 3: 270°
2	x-mirror	1: enable
3	y-mirror	1: enable

Bits	Description	Remarks
9:8	data width of the image pixel	0 : 8bit 1: 16bit 2: 24bit 3: 32bit
18:16	the identification of the image channel	0 : graphic channel 0 1 : graphic channel 1 2 : graphic channel 2 3 : graphic channel 3 4 : graphic channel 4 5 : graphic channel 5 6 : graphic channel 6 7 : graphic channel 7
29	pause_enable	1: dma enters the pause state after the current node is completed
30	Interrupt_enable	1: generate the node interrupt after the current node is completed
63:32	src_addr	source address
79:64	width	the number of pixels per line (row) in the current image
95:80	height	the number of pixels per column in the current image
111:96	src_stride	the number of bytes between two lines of the source image
127:112	dst_stride	the number of bytes between two lines of the destination image
159:128	dst_addr	destination address
191:160	next_ll_i_addr	0: means the last node

Pause and stop When the pause function is set in the linked list, dma enters the pause state after the current dma transfer is completed, and a pause interrupt will be issued. When the register gdma_resume is enabled, obtain the next linked list and continue the dma transmission. When the stop function is enabled during dma transmission, the current transmission is canceled, but the ongoing bus transaction will be completed. After the stop function is enabled, the software needs to wait at least 1ms before starting a new transmission.

10.2.5 Working Mode

linked-list work mode

1. Read the register DMA_CH_EN make sure the gdma channel is available
2. Configure the register DMA_INT_STAT to clear the interrupt status of the gdma channel
3. Configure the linked list and write it into memory
4. Configure the physical address of the first linked list to register GDMA_LLT_SADDR
5. Configuration register DMA_WEIGHT, configure channel arbitration weight, make sure that DMA_CKG.bit0=1 when configuring this register
6. Configure the register GDMA_CTL to start the GDMA
7. Wait for the interrupt, according to the configuration, wait for the pull-up of the linked list node interrupt or DMA completion interrupt
8. After receiving the DMA interrupt, the DMA transfer ends
9. Clear the interrupt status and start the next transfer

10. After all transfers are completed, configure the register DMA_CH_EN to close the channel

10.2.6 Register Summary

The base address of the GDMA Register reference chapter 1 Table 10-2-2 GDMA registers

Offset	Register	Description
0x0	DMA_CH_EN	dma enable register
0xc	DMA_CFG	dma configure register
0x4	DMA_INT_MASK	dma interrupt mask register
0x8	DMA_INT_STAT	dma interrupt status register
0x10	GDMA_CTRL	gdma control register
0x14	GDMA_LLI_BASE	gdma linked list base address register
0x18	GDMA_CH0_CNT	gdma graphic channel0 counter register
0x1c	GDMA_CH1_CNT	gdma graphic channel1 counter register
0x20	GDMA_CH2_CNT	gdma graphic channel2 counter register
0x24	GDMA_CH3_CNT	gdma graphic channel3 counter register
0x28	GDMA_CH4_CNT	gdma graphic channel4 counter register
0x2c	GDMA_CH5_CNT	gdma graphic channel5 counter register
0x30	GDMA_CH6_CNT	gdma graphic channel6 counter register
0x34	GDMA_CH7_CNT	gdma graphic channel7 counter register
0x38	GDMA_CURRENT_LLT	gdma current linked list address register
0x48	DMA_WEIGHT	dma arbitration weight register

10.2.7 Register Description

DMA_CH_EN DMA_CH_EN is a dma enable register Offset Address:0x0 , Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
4	RW	gdma_enable	gdma channel enable 0x1: enable channel 0x0: close the channel	0x0
3:0	RW	chx_channel_enable	sdma channel enable bit0: channel 0 bit1: channel 1 bit2: channel 2 bit3: channel 3 0x1: enable channel 0x0: close the channel	0x0

DMA_CFG

DMA_CFG is a dma configure register Offset Address:0xc , Total Reset Value:0x0000_07ff

Bits	Access	Name	Description	Reset
12	RW	gdma_burst_len4_en	0b1: gdma burst length is 4 0b0: gdma burst length is 8	0x0
10:8	RW	gdma_max_outstanding	Maximum outstanding transactions 0x0: maximum is 1 0x1: maximum is 2 0x2: maximum is 3 ... 0x7: maximum is 8	0x7
7:0	RW	ckg_bypass	bit0: 1 indicates gsdma read/write arbitration clock automatic gate bypass bit1: 1 indicates gdma control module clock automatic gate bypass bit2: 1 indicates sdma channel 0 automatic gate bypass bit3: 1 indicates sdma channel 1 automatic gate bypass bit4: 1 indicates sdma channel 2 automatic gate bypass bit5: 1 indicates sdma channel 3 automatic gate bypass bit6: 1 indicates gdma read channel automatic gate bypass bit7: 1 indicates gdma write channel automatic gate bypass	0xff

DMA_INT_MASK DMA_INT_MASK is a dma interrupt mask register Offset Address:0x4 , Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
18	RW	gdma_tr_done_mask	1: Mask the gdma transmission complete interrupt	0x0
17	RW	gdma_lli_item_int_mask	1: Mask the gdma linked list node interrupt	0x0
16	RW	gdma_lli_pause_mask	1: Mask the gdma pause interrupt	0x0
11:8	RW	chx_llt_pause_mask	Mask the sdma pause interrupt bit8: channel 0 bit9: channel 1 bit10: channel 2 bit11: channel 3	0x0
7:4	RW	chx_llt_item_int_mask	Mask the sdma linked list node interrupt bit4: channel 0 bit5: channel 1 bit6: channel 2 bit7: channel 3	0x0
3:0	RW	chx_tr_done_mask	Mask the sdma transmission complete interrupt bit0: channel 0 bit1: channel 1 bit2: channel 2 bit3: channel 3	0x0

DMA_INT_STAT DMA_INT_STAT is a dma interrupt status register Offset Address:0x8 , Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
18	W1C	gdma_tr_done_int	gdma transmission completed interrupt flag, write 1 to clear	0x0
17	W1C	gdma_lli_item_int	gdma linked list node interrupt flag, write 1 to clear	0x0
16	W1C	gdma_lli_pause_int	gdma pause interrupt flag, write 1 to clear	0x0
11:8	W1C	chx_llt_pause_int	sdma pause interrupt flag, write 1 to clear bit8: channel 0 bit9: channel 1 bit10: channel 2 bit11: channel 3	0x0
7:4	W1C	chx_llt_item_int	sdma linked list node interrupt flag, write 1 to clear bit4: channel 0 bit5: channel 1 bit6: channel 2 bit7: channel 3	0x0
3:0	W1C	chx_tr_done_int	sdma transmission completed interrupt flag, write 1 to clear bit0: channel 0 bit1: channel 1 bit2: channel 2 bit3: channel 3	0x0

GDMA_CTRL GDMA_CTRL is a gdma control register Offset Address:0x10,Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
2	WO1	gdma_resume	gdma resume after pause	0x0
1	WO1	gdma_stop	gdma stop working	0x0
0	WO1	gdma_start	gdma start	0x0

GDMA_LLI_BASE

GDMA_LLI_BASE is a gdma linked list base address register Offset Address:0x14,Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:0	RW	gdma_lli_base	base address of the linked list	0x0

GDMA_CH0_CNT GDMA_CH0_CNT is a gdma graphic channel0 counter register Offset Address:0x18,Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:0	W0C	gdma_ch0_cnt	The number of nodes that have been transferred which been identified as graphic channel 0	0x0

GDMA_CH1_CNT GDMA_CH1_CNT is a gdma graphic channel1 counter register Offset Address:0x1c,Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset

Bits	Access	Name	Description	Reset
31:0	W0C	gdma_ch1_cnt	The number of nodes that have been transferred which been identified as graphic channel 1	0x0

GDMA_CH2_CNT GDMA_CH2_CNT is a gdma graphic channel2 counter register Offset Address:0x20,Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:0	W0C	gdma_ch2_cnt	The number of nodes that have been transferred which been identified as graphic channel 2	0x0

GDMA_CH3_CNT GDMA_CH3_CNT is a gdma graphic channel3 counter register Offset Address:0x24,Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:0	W0C	gdma_ch3_cnt	The number of nodes that have been transferred which been identified as graphic channel 3	0x0

GDMA_CH4_CNT GDMA_CH4_CNT is a gdma graphic channel4 counter register Offset Address:0x28,Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:0	W0C	gdma_ch4_cnt	The number of nodes that have been transferred which been identified as graphic channel 4	0x0

GDMA_CH5_CNT GDMA_CH5_CNT is a gdma graphic channel5 counter register Offset Address:0x2c,Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:0	W0C	gdma_ch5_cnt	The number of nodes that have been transferred which been identified as graphic channel 5	0x0

GDMA_CH6_CNT GDMA_CH6_CNT is a gdma graphic channel6 counter register Offset Address:0x30, Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:0	W0C	gdma_ch6_cnt	The number of nodes that have been transferred which been identified as graphic channel 6	0x0

GDMA_CH7_CNT GDMA_CH7_CNT is a gdma graphic channel7 counter register Offset Address:0x34,Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:0	W0C	gdma_ch7_cnt	The number of nodes that have been transferred which been identified as graphic channel 7	0x0

GDMA_CURRENT_LLT

GDMA_CURRENT_LLT is a gdma current linked list address register Offset Address:0x38, Total Reset

Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:0	RW	gdma_current_linklist	The linked list address of the current transaction, used for debug	0x0

DMA_WEIGHT DMA_WEIGHT is a dma arbitration weight register Offset Address:0x48,Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
23:0	RW	gsdma_weight	Used to control the access priority weight of the DMA channels. This register is divided into 8 fields, each field is 3-bit, and each 3-bit field can be used to set an arbitration code. This arbitration code is used to determine the priority of the DMA channel when requesting to use the master AXI port. The arbitration code has 5 valid values: 0, 1, 2, 3, 4 0: ch0 > ch1 > ch2 > ch3 > gdma ; 1: ch1 > ch2 > ch3 > gdma > ch0; 2: ch2 > ch3 > gdma > ch0 > ch1; 3: ch3 > gdma > ch0 > ch1 > ch2; 4: gdma > ch0 > ch1 > ch2 > ch3 ;	0x0

11 Audio Interface

11.1 Audio

11.1.1 Overview

The audio is used to connect to the on-chip or off-chip audio coder/decoder (codec) to input and output audio data, implementing the recording, talkback, and playback functions. Audio supports stereo/mono inputs in I2S and pdm format, and outputs in I2S format.

11.1.2 Feature

K230 audio supports the following features:

- PDM audio
 - PDM audio input with data sampling rate of 2.048/2.8224 MHz, 1-bit data width, sampling clock rate of 0.256MHz/0.384MHz/0.512MHz/0.768MHz/1.024MHz/1.4112MHz/1.536MHz/2.048MHz/2.8224MHz/3.072 MHz/4.096MHz/5.6448MHz/6.144MHz/12.288MHz/24.576MHz, corresponding PCM sampling rate of 8kHz/12kHz/16kHz/24kHz/32kHz/44.1kHz/48kHz/96kHz/192kHz.
 - Oversampling rate is configurable with x128, x64 and x32.
 - 1-4 IOs are used for input PDM audio.
 - The input can be configured with 1-8 PDM channels. It supports left/right mono mode and dual mode of PDM. All IO channel modes are unified. The maximum number of IO in dual mode is 4.
 - The serial numbers of enabled channels are from small to big. Random enabling of each channel is not supported.
 - The skew between PDM input data and pdm_clk is up to 18 PCLK cycles.
 - Conversion from input PDM audio data to PCM audio data.

- I2S audio
 - PCM audio in I2S Phillips, left-aligned and right-aligned format.
 - I2S audio input/output with data sampling rate of 8kHz/12kHz/16kHz/24kHz/32kHz/44.1kHz/48kHz/96kHz/192kHz, and data width of 32 bits. The valid data width(32/24/20/16/12bits) needs to be customized. The sampling clock rate supports $2 \times 32(\text{data width}) \times \text{data sampling rate}$.
 - 1-2 IO to input, and 1-2 IO to output I2S audio data. The full-duplex mode is supported.
 - Connect to on-chip or off-chip i2s codec to input and output audio data.
 - Each RX/TX channel FIFO depth is 8. FIFO threshold can be configured.
 - DMA mode enable and disable.
 - io number and left/right channel flag can be fetched from higher bits of data when valid data width is less than 32bits in DMA mode.
- Support master mode, which means clock(pdm_clk and i2s_sclk) and ws(only I2S) are output from audio.
- I2S input, pdm input and I2s output audio modes can work simultaneously.
- APB2.0 interface is used to configure registers, and read/write PCM data. The default frequency of configurable APB working clock PCLK is 100MHz, and the data interface is 32 bits.
- PCM data can be transferred by PDMA. Burst length in PDMA is configurable.
- The audio module can start working again after being disabled and re-enabled.
- The audio module can start working again after asserting reset and de-asserting reset.

Figure 11-1-1 shows the block diagram of audio.

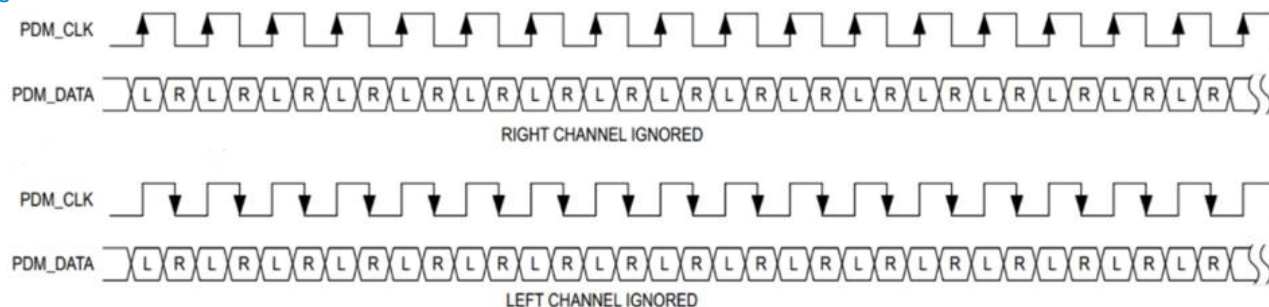


11.1.4 Function Description

Audio only supports PDM input master mode, which means pdm microphone working clock pdm_clk is supplied by K230(as depicted in [Figure 11-1-1](#). pdm_clk could be inverted by setting [PDM_CONTROL_REG_0](#) bit14 pdm_clk_sel to 0x1.

1. Mono Mode

[Figure 11-1-2 PDM Mono Mode](#)

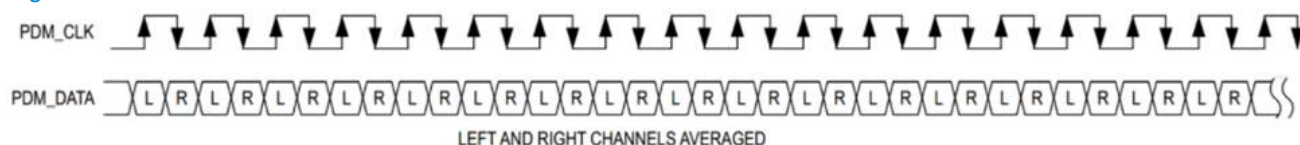


When configuring bit5-bit4 of register [pdm_channel_enable](#) to 0x2, audio supports mono PDM audio captured by posedge of pdm_clk_root for left channel; When configuring bit5-bit4 to 0x3, audio supports mono PDM audio captured by negedge of pdm_clk_root for right channel.

In mono mode PDM channel number is bigger than 0 and equal or less than 4, if it is less than 4, then front channel serial number will be used, for example: if bit3-bit0 of register [pdm_channel_enable](#) are configured to 0x3, then channel0~channel2 are enabled and 3 chip IOs will be used as PDM audio data IO, there is no other situation.

2. Stereo mode

[Figure 11-1-3 PDM Stereo Mode](#)



When configuring bit5 of register [pdm_channel_enable](#) to 0x0, audio supports stereo PDM audio captured by posedge of pdm_clk_root for left channel and negedge of pdm_clk_root for right channel.

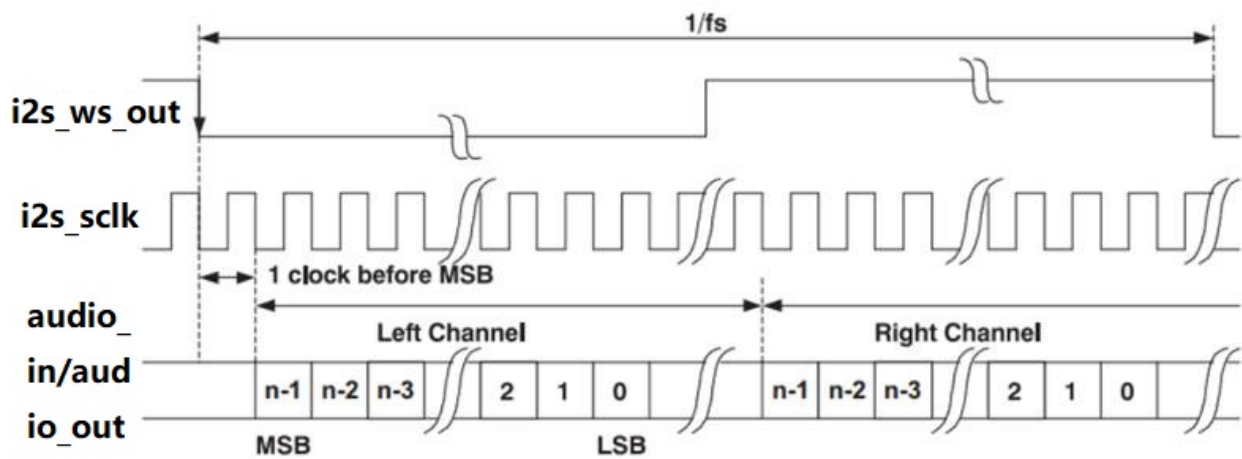
In stereo mode PDM channel number is bigger than 0 and equal or less than 8, if it is less than 8, it should be even number and front channel serial number will be used, for example: if bit3-bit0 of register [pdm_channel_enable](#) are configured to 0x4, then channel0~channel3 are enabled and 2 chip IOs will be used as PDM audio data IO, there is no other situation.

11.1.4.1.2 I2S Mode

Audio supports I2S input and output master mode, which means I2S microphone or codec working clock i2s_sclk(audio_clk) and word select signal i2s_ws_out are supplied by K230(as depicted in [Figure 11-1](#). Data width of audio_in/audio_out is 32bits, and valid data width(32/24/20/16/12bits) can be configured by [RCRx](#) (for x = 0; x <= 1) and [TCRx](#) (for x = 0; x <= 1) . The I2S interface transmits data in most significant bit (MSB) first mode. Data(audio_in/audio_out) and WS signal(i2s_ws_out) are sent at the falling edge of the i2s_sclk and sampled at the rising edge of the i2s_sclk. i2s_sclk could be inverted by setting [PDM_CONTROL_REG_0](#) bit13 audio_dev_clk_sel to 0x1.

1. Philips Mode

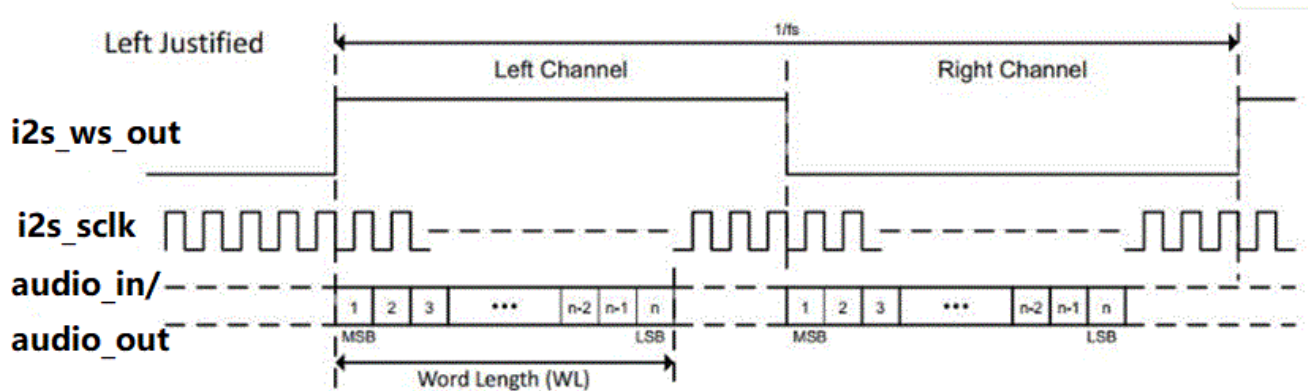
[Figure 11-1-4 I2S Philips Mode](#)



The data(`audio_in/audio_out`) is delayed by one `i2s_sclk` cycle compared with `WS` signals(`i2s_ws_out`). For `WS` signal `i2s_ws_out`, it goes low to indicate left channel and high to right channel.

2. Left Justified Mode

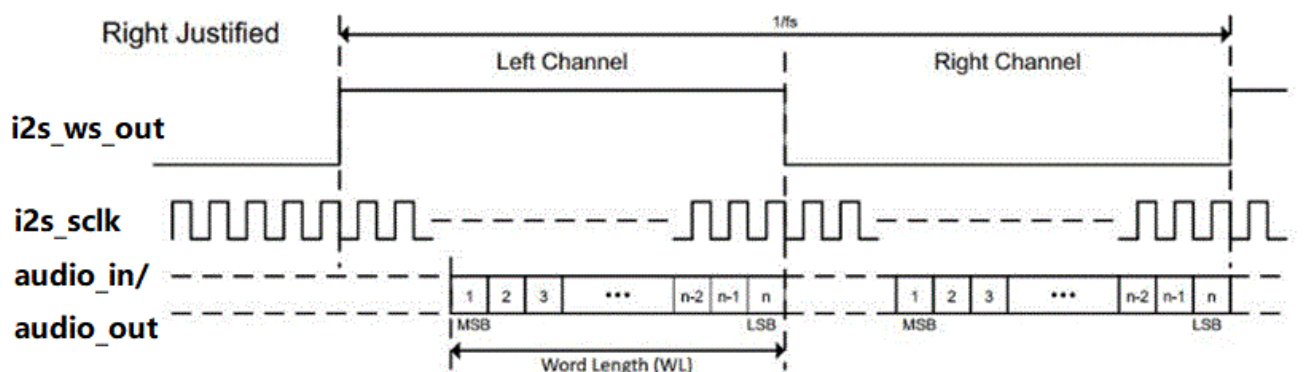
Figure 11-1-5 I2S Left Justified Mode



The data(`audio_in/audio_out`) starts sending the first bit at the same time when `i2s_ws_out` changes. For `WS` signal `i2s_ws_out`, it goes high to indicate left channel and low to right channel.

3. Right Justified Mode

Figure 11-1-6 I2S Right Justified Mode



The data(`audio_in/audio_out`) ends sending the last bit aligned to the transition edge of `i2s_ws_out`. For `WS` signal `i2s_ws_out`, it goes high to indicate left channel and low to right channel.

11.1.4.2 Clock Description

Clock domains are colored in Figure 11-1-1, there are 2 clock domains in `audio_top`: `audio_dev_clk`, `audio_pclk`. Frequencies of `audio_dev_clk` is listed in [Table 11-2](Table_11-2). Frequency of `audio_pclk` is 100MHz after reset, it can be configured by `LS_CLKDIV_CFG` bit2-0(`ls_pclk_div`).

Table 11-1 frequencies of `audio_dev_clk` for i2s

mode	IO	channel/IO	in/out	bit/channel	sample rate(kHz)	audio_dev_clk(MHz)
I2S	1~2	2	out	32	8	0.512
I2S	1~2	2	out	32	12	0.768
I2S	1~2	2	out	32	16	1.024
I2S	1~2	2	out	32	24	1.536
I2S	1~2	2	out	32	32	2.048
I2S	1~2	2	out	32	44.1	2.8224
I2S	1~2	2	out	32	48	3.072
I2S	1~2	2	out	32	96	6.144
I2S	1~2	2	out	32	192	12.288
I2S	1~2	2	in	32	8	0.512
I2S	1~2	2	in	32	12	0.768
I2S	1~2	2	in	32	16	1.024
I2S	1~2	2	in	32	24	1.536
I2S	1~2	2	in	32	32	2.048
I2S	1~2	2	in	32	44.1	2.8224
I2S	1~2	2	in	32	48	3.072
I2S	1~2	2	in	32	96	6.144
I2S	1~2	2	in	32	192	12.288

Table 11-2 frequencies of pdm_clk for pdm

mode	IO	channel/IO	in/out	PCM bit	sample rate(kHz)	oversample	pdm_clk(MHz)
PDM	1~4	1/2	in	24	8	128	1.024
PDM	1~4	1/2	in	24	12	128	1.536
PDM	1~4	1/2	in	24	16	128	2.048
PDM	1~4	1/2	in	24	24	128	3.072
PDM	1~4	1/2	in	24	32	128	4.096
PDM	1~4	1/2	in	24	44.1	128	5.6448
PDM	1~4	1/2	in	24	48	128	6.144
PDM	1~4	1/2	in	24	96	128	12.288
PDM	1~4	1/2	in	24	192	128	24.576
PDM	1~4	1/2	in	24	8	64	0.512
PDM	1~4	1/2	in	24	12	64	0.768
PDM	1~4	1/2	in	24	16	64	1.024
PDM	1~4	1/2	in	24	24	64	1.536

mode	IO	channel/IO	in/out	PCM bit	sample rate(kHz)	oversample	pdm_clk(MHz)
PDM	1~4	1/2	in	24	32	64	2.048
PDM	1~4	1/2	in	24	44.1	64	2.8224
PDM	1~4	1/2	in	24	48	64	3.072
PDM	1~4	1/2	in	24	96	64	6.144
PDM	1~4	1/2	in	24	192	64	12.288
PDM	1~4	1/2	in	24	8	32	0.256
PDM	1~4	1/2	in	24	12	32	0.384
PDM	1~4	1/2	in	24	16	32	0.512
PDM	1~4	1/2	in	24	24	32	0.768
PDM	1~4	1/2	in	24	32	32	1.024
PDM	1~4	1/2	in	24	44.1	32	1.4112
PDM	1~4	1/2	in	24	48	32	1.536
PDM	1~4	1/2	in	24	96	32	3.072
PDM	1~4	1/2	in	24	192	32	6.144

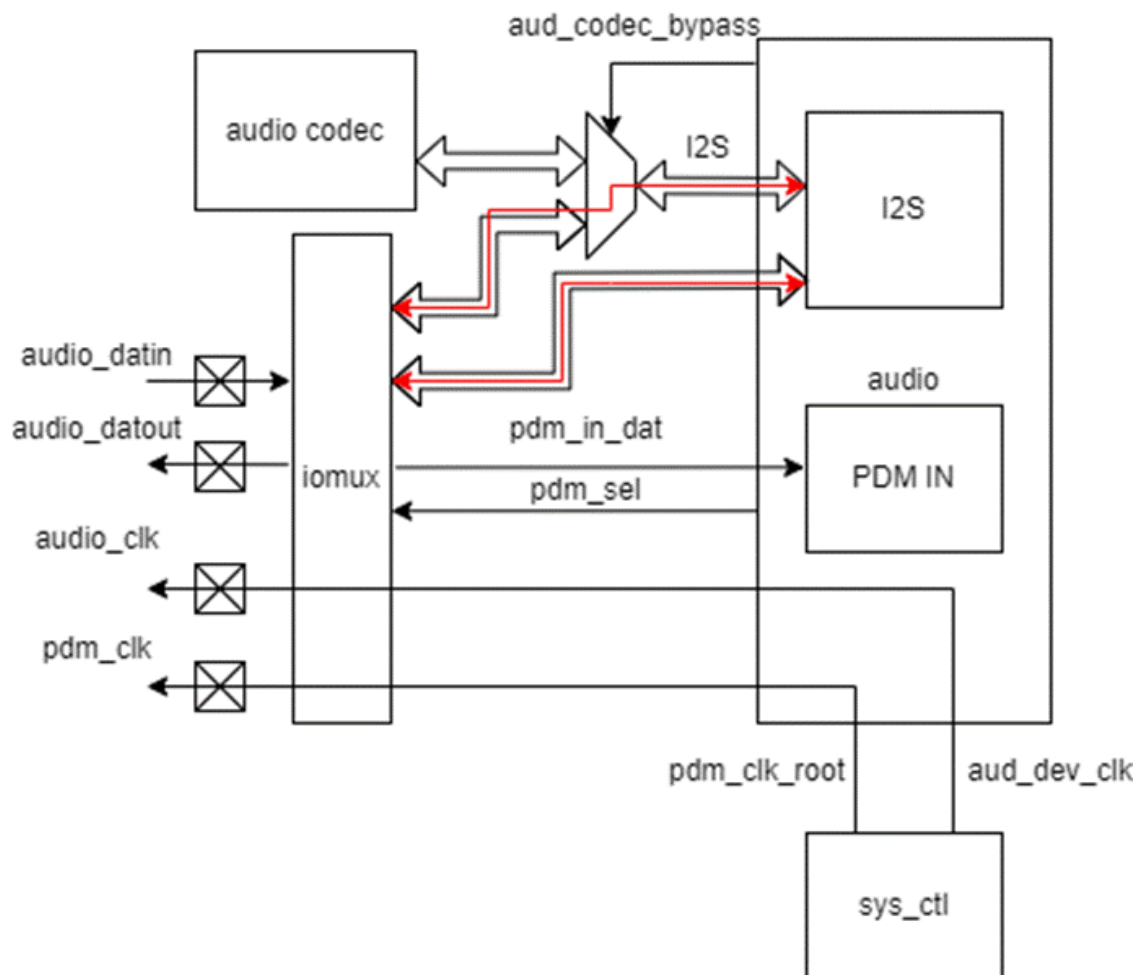
Oversample rate is configured by PDM_CONTROL_REG_1 bit0 ord_sel and pdm2pcm_filtr_bypass bit0 hbf_dsmpl_fir_bypass:

Table 11-3 Oversample rate

ord_sel	hbf_dsmpl_fir_bypass	oversample rate
1	0	128
0	0	64
0	1	32

11.1.4.3 Application Description

1. Connected to internal codec
Figure 11-1-7 Connected to internal codec

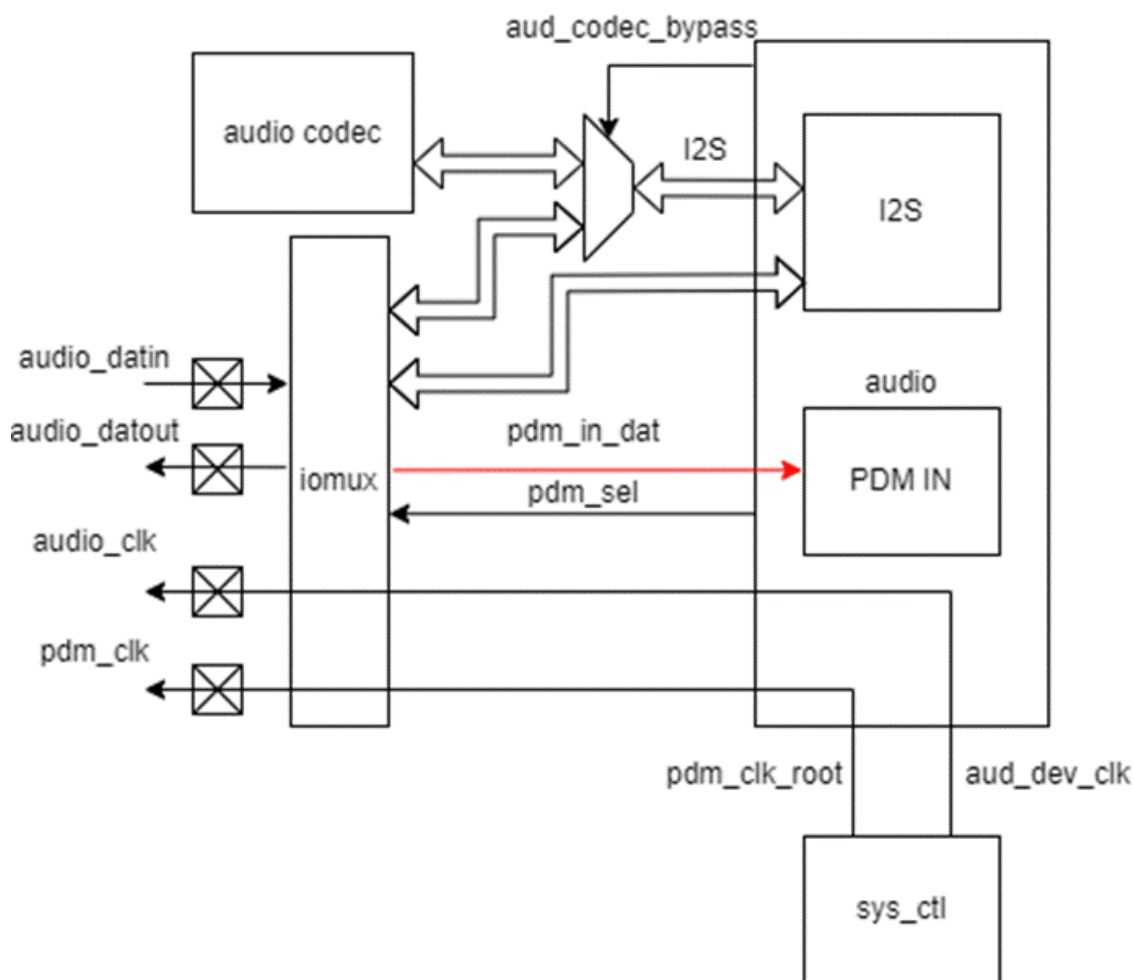


Set `PDM_CONTROL_REG_0` bit5 `aud_codec_bypass` to 0x1 to not use audio codec, bit0 `pdm_en` to 0x0 to gate `pdm_clk` and disable `pdm`. Set `pdm_clk_enable`(`LS_CLKEN_CFG0` bit13) to 0x0 to gate `pdm_clk_root`. Set `IER` to 0x1 to enable I2S, set `IRER` to 0x1 to enable receiver, set `ITER` to 0x1 to enable transmitter. `sdi0` and `sdi1` can be enabled separately by configuring `RERx` (for `x = 0; x <= 1`), `sdo0` and `sdo1` can be enabled separately by configuring `TERx` (for `x = 0; x <= 1`).

Signal from external digital microphone is received by I2S sdi0/sdi1; Signal from I2S in audio is transmitted through sdo0/sdo1 of I2S to external coec and PA.

i2s_sclk outputs to IO32(IIS_CLK), i2s_ws_out outputs to IO33(IIS_WS), sdi0 inputs from IO34(IIS_D_IN0), sdi1 inputs from IO36(IIS_D_IN1), sdo0 outputs to IO35(IIS_D_OUT0, need to set [PDM_CONTROL_REG_0](#) bit7-bit6(audio_in_mode) to 0x2), sdo1 outputs to IO37(IIS_D_OUT1, need to set [PDM_CONTROL_REG_0](#) bit7-bit6(audio in mode) to 0x2).

3. Connected to external pdm microphone [Figure 11-1-9 Connected to external pdm microphone](#)

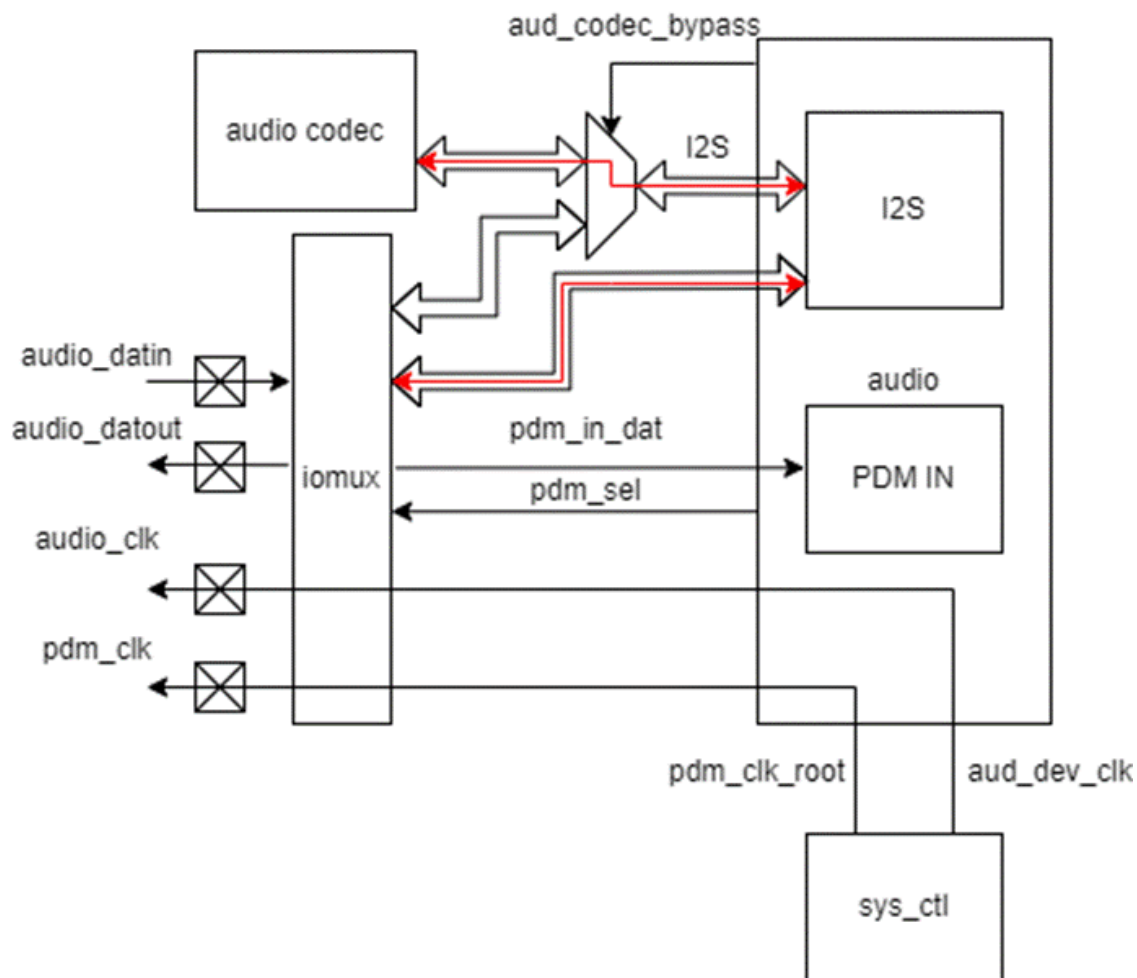


Set [PDM_CONTROL_REG_0](#) bit0(pdm_en) to 0x1 to enable pdm. Set aud_dev_clk_enable([LS_CLKEN_CFG0](#) bit28) to 0x0 to gate aud_dev_clk, and set [IER](#) to 0x0 to disable I2S.

Signal from external digital microphone is received by pdm_in0/1/2/3 or pdm_in0/1/2 or pdm_in0/1 or pdm_in0(configured by [pdm_channel_enable](#), ex: 0x8 for 8 channels with 4 IOs pdm_in0/1/2/3 double edge; 0x23 for 3 channels with 3 IOs pdm_in0/1/2 posedge edge).

pdm_clk outputs to IO26(PDM_CLK), pdm_in0 inputs from IO27/IO37(PDM_IN0 ; If IO37, need to set [PDM_CONTROL_REG_0](#) bit7-bit6(audio_in_mode) to other than 0x2), pdm_in1 inputs from IO28/IO35(PDM_IN1; If IO35, need to set [PDM_CONTROL_REG_0](#) bit7-bit6(audio_in_mode) to other than 0x2), pdm_in2 inputs from IO36(PDM_IN2), pdm_in3 inputs from IO34(PDM_IN3).

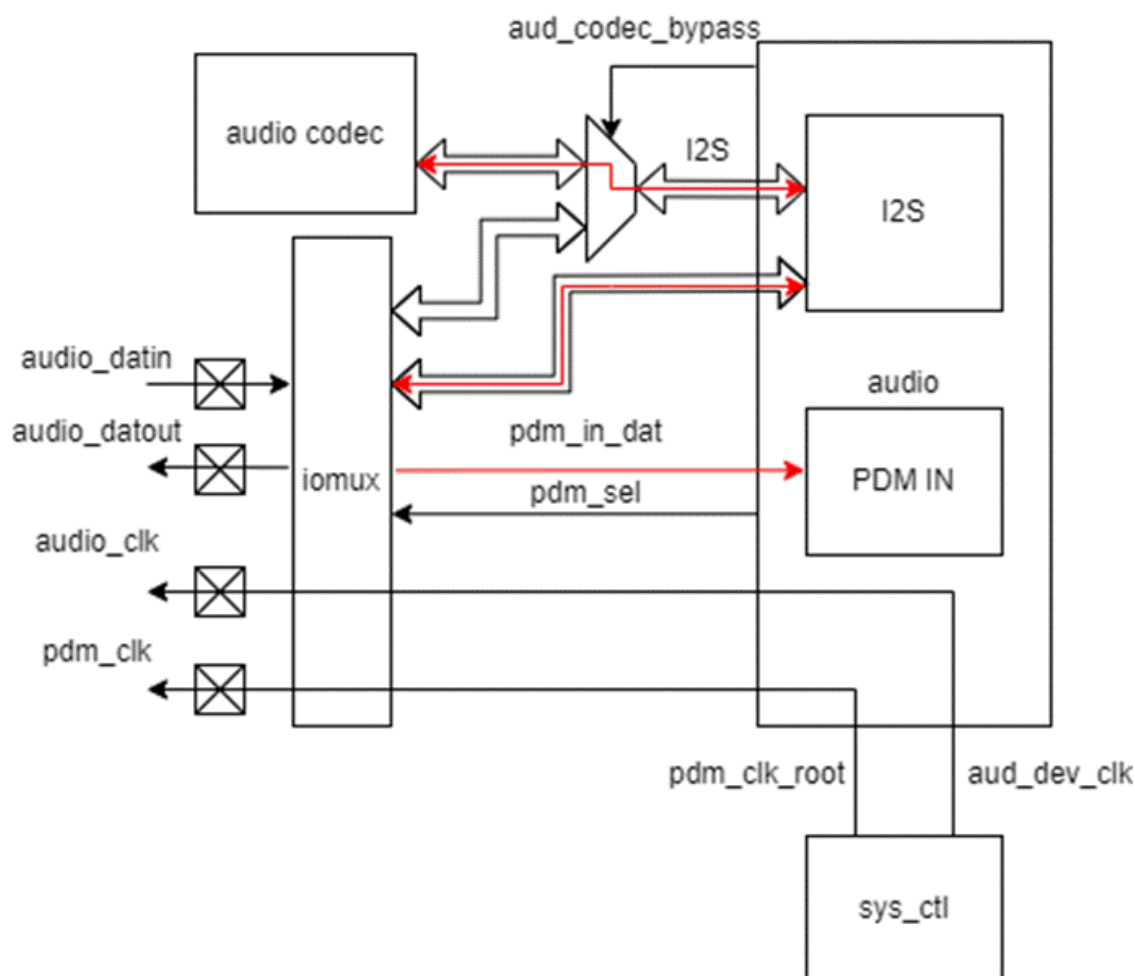
4. Connected to internal codec and external I2S codec [Figure 11-1-10 Connected to internal and external I2S codec](#)



Set [PDM_CONTROL_REG_0](#) bit5 `aud_codec_bypass` to 0x1 to not use audio codec, bit0 `pdm_en` to 0x0 to gate `pdm_clk` and disable `pdm`. Set `pdm_clk_enable`([LS_CLKEN_CFG0](#) bit13) to 0x0 to gate `pdm_clk_root`. Set [IER](#) to 0x1 to enable I2S, set [IRER](#) to 0x1 to enable receiver, set [ITER](#) to 0x1 to enable transmitter. `sdi0/sdi1` are enabled by configuring [RERx](#) (for $x = 0; x \leq 1$), `sdo0/sdo1` are enabled by configuring [TERx](#) (for $x = 0; x \leq 1$). Audio connect audio codec with I2S `sdi0/sdo0`, connect external I2S digital microphone with I2S `sdi1`, and connect external coec and PA with I2S `sdo1`. `i2s_sclk` outputs to IO32(IIS_CLK) and is also connected to audio codec, `i2s_ws_out` outputs to IO33(IIS_WS) and is also connected to audio codec. `sdi1` inputs from IO36(IIS_D_IN1), and `sdo1` outputs to IO37(IIS_D_OUT1), need to set [PDM_CONTROL_REG_0](#) bit7-bit6(`audio_in_mode`) to 0x2).

5. Connected to internal codec and external pdm codec [Figure 11-1-11 Connected to internal and external pdm codec](#)

7. Connected to internal codec, external I2S microphone/codec and pdm microphone [Figure 11-1-13 Connected to internal codec, external I2S microphone/codec and pdm microphone](#)



Set **PDM_CONTROL_REG_0** bit5 **aud_codec_bypass** to 0x0 to use audio codec, bit0(**pdm_en**) to 0x1 to enable pdm. Set **IER** to 0x1 to enable I2S, set **IRER** to 0x1 to enable receiver, set **ITER** to 0x1 to enable transmitter. **sdi0** and **sdi1** are enabled separately by configuring **RER0** and **RER1**, **sdo0** and **sdo1** are enabled separately by configuring **TER0** and **TER1**.

Audio connect audio codec with I2S **sdi0/sdo0**. Signal from external I2S digital microphone is received by I2S **sdi1**; Signal from I2S in audio is transmitted through **sdo1** of I2S to external coec and PA. Signal from external PDM digital microphone is received by **pdm_in0/1** or **pdm_in0**(configured by **pdm_channel_enable**, ex: 0x1 for 2 channels with 2 IOs **pdm_in0/1** double edge; 0x22 for 2 channels with 2 IOs **pdm_in0/1** posedge edge).

i2s_sclk outputs to IO32(IIS_CLK) and is also connected to audio codec, **i2s_ws_out** outputs to IO33(IIS_WS) and is also connected to audio codec. **sdi1** inputs from IO36(IIS_D_IN1), and **sdo1** outputs to IO37(IIS_D_OUT1), need to set **PDM_CONTROL_REG_0** bit7-bit6(**audio_in_mode**) to 0x2). **pdm_clk** outputs to IO26(PDM_CLK), **pdm_in0** inputs from IO27/IO37(PDM_IN0 ; If IO37, need to set **PDM_CONTROL_REG_0** bit7-bit6(**audio_in_mode**) to other than 0x2), **pdm_in1** inputs from IO28/IO35(PDM_IN1; If IO35, need to set **PDM_CONTROL_REG_0** bit7-bit6(**audio_in_mode**) to other than 0x2).

11.1.5 Register Description

11.1.5.1 Internal Address Mapping

base address: 0x9140_F000

I2S address: 1kB (0x9140_F000~0x9140_F3FF)

PDM IN address: 1kB (0x9140_F400~0x9140_F7FF)

11.1.5.2 Registers Summary

Table 11-4 and Table 11-5 describe audio registers:

Table 11-4 Summary of audio I2S registers

Offset Address	Register	Description
0x0	IER	This register acts as a global enable/disable for DW_apb_i2s.
0x4	IRER	This register acts as an enable/disable for the DW_apb_i2s Receiver block.
0x8	ITER	This register acts as an enable/disable for the DW_apb_i2s Transmitter block.
0xc	CER	This register acts as an enable/disable for the DW_apb_i2s Clock Generation block.
0x10	CCR	This register configures the ws_out and sclk_gate signals when DW_apb_i2s is a master.
0x14	RXFFR	This register specifies the Receiver Block FIFO Reset Register.
0x18	TXFFR	This register specifies the Transmitter Block FIFO Reset Register.
0x020 + 0x40*x	LRBRx (for x = 0; x <= 1)	This specifies the Left Receive Buffer Register.
0x020 + 0x40*x	LTHRx (for x = 0; x <= 1)	This specifies the Left Transmit Holding Register.
0x024 + 0x40*x	RRBRx (for x = 0; x <= 1)	This specifies the Right Receive Buffer Register.
0x024 + 0x40*x	RTHRx (for x = 0; x <= 1)	This specifies the Right Transmit Holding Register.
0x028 + 0x40*x	RERx (for x = 0; x <= 1)	This specifies the Receive Enable Register.
0x02C + 0x40*x	TERx (for x = 0; x <= 1)	This specifies the Transmit Enable Register.
0x030 + 0x40*x	RCRx (for x = 0; x <= 1)	This specifies the Receive Configuration Register.
0x034 + 0x40*x	TCRx (for x = 0; x <= 1)	This specifies the Transmit Configuration Register.
0x038 + 0x40*x	ISRx (for x = 0; x <= 1)	This specifies the Interrupt Status Register.
0x03C + 0x40*x	IMRx (for x = 0; x <= 1)	This specifies the Interrupt Mask Register.
0x040 + 0x40*x	RORx (for x = 0; x <= 1)	This specifies the Receive Overrun Register.
0x044 + 0x40*x	TORx (for x = 0; x <= 1)	This specifies the Transmit Overrun Register.
0x048 + 0x40*x	RFCRx (for x = 0; x <= 1)	This specifies the Receive FIFO Configuration Register.

Offset Address	Register	Description
0x04C + 0x40*x	TFCRx (for x = 0; x <= 1)	This specifies the Transmit FIFO Configuration Register.
0x050 + 0x40*x	RFFx (for x = 0; x <= 1)	This specifies the Receive FIFO Flush Register.
0x054 + 0x40*x	TFFx (for x = 0; x <= 1)	This specifies the Transmit FIFO Flush.
0x5c0	[RXDMA] (#RXDMA)	Receiver Block DMA Register.
0x5c4	RRXDMA	Reset Receiver Block DMA Register.
0x5c8	TXDMA	Transmitter Block DMA Register.
0x5cc	RTXDMA	Reset Transmitter Block DMA Register.

Table 11-5 Summary of audio pdm registers

Offset Address	Register	Description
0x400	PDM_CONTROL_REG_0	PDM input configuration registerpdm input channel
0x404	PDM_CONTROL_REG_1	pdm input/output channel CIC compensation filter coefficient
0x408	mp_para_tap_coef_0	pdm input/output channel CIC compensation filter coefficient
0x40C	mp_para_tap_coef_1	pdm input/output channel CIC compensation filter coefficient
0x410	mp_para_tap_coef_2	pdm input/output channel CIC compensation filter coefficient
0x414	mp_para_tap_coef_3	pdm input/output channel CIC compensation filter coefficient
0x418	mp_para_tap_coef_4	pdm input/output channel CIC compensation filter coefficient
0x41c	mp_para_tap_coef_5	pdm input/output channel CIC compensation filter coefficient
0x420	mp_para_tap_coef_6	pdm input/output channel CIC compensation filter coefficient
0x424	mp_para_tap_coef_7	pdm input/output channel CIC compensation filter coefficient
0x428	mp_para_tap_coef_8	pdm input/output channel CIC compensation filter coefficient
0x42C	mp_para_tap_coef_9	pdm input/output channel CIC compensation filter coefficient
0x430	mp_para_tap_coef_10	pdm input/output channel CIC compensation filter coefficient
0x434	mp_para_tap_coef_11	pdm input/output channel CIC compensation filter coefficient
0x438	mp_para_tap_coef_12	pdm input/output channel CIC compensation filter coefficient
0x43c	mp_para_tap_coef_13	pdm input/output channel CIC compensation filter coefficient
0x440	mp_para_tap_coef_14	pdm input/output channel CIC compensation filter coefficient
0x444	mp_para_tap_coef_15	pdm input/output channel CIC compensation filter coefficient
0x448	hbf_para_tap_coef_0	pdm input/output channel HBF filter coefficient
0x44c	hbf_para_tap_coef_1	pdm input/output channel HBF filter coefficient
0x450	hbf_para_tap_coef_2	pdm input/output channel HBF filter coefficient

Offset Address	Register	Description
0x454	hbf_para_tap_coef_3	pdm input/output channel HBF filter coefficient
0x458	hbf_para_tap_coef_4	pdm input/output channel HBF filter coefficient
0x45c	hbf_para_tap_coef_5	pdm input/output channel HBF filter coefficient
0x460	hbf_para_tap_coef_6	pdm input/output channel HBF filter coefficient
0x464	hbf_para_tap_coef_7	pdm input/output channel HBF filter coefficient
0x468	hbf_para_tap_coef_8	pdm input/output channel HBF filter coefficient
0x46c	hbf_para_tap_coef_9	pdm input/output channel HBF filter coefficient
0x470	hbf_para_tap_coef_10	pdm input/output channel HBF filter coefficient
0x474	hbf_para_tap_coef_11	pdm input/output channel HBF filter coefficient
0x478	hbf_para_tap_coef_12	pdm input/output channel HBF filter coefficient
0x47c	hbf_para_tap_coef_13	pdm input/output channel HBF filter coefficient
0x480	hbf_para_tap_coef_14	pdm input/output channel HBF filter coefficient
0x484	hbf_para_tap_coef_15	pdm input/output channel HBF filter coefficient
0x488	hbf_para_tap_coef_16	pdm input/output channel HBF filter coefficient
0x48c	hbf_para_tap_coef_17	pdm input/output channel HBF filter coefficient
0x490	hbf_para_tap_coef_18	pdm input/output channel HBF filter coefficient
0x494	hbf_para_tap_coef_19	pdm input/output channel HBF filter coefficient
0x498	hbf_para_tap_coef_20	pdm input/output channel HBF filter coefficient
0x49c	hbf_para_tap_coef_21	pdm input/output channel HBF filter coefficient
0x4A0	hbf_para_tap_coef_22	pdm input/output channel HBF filter coefficient
0x4A4	hbf_para_tap_coef_23	pdm input/output channel HBF filter coefficient
0x4A8	hbf_para_tap_coef_24	pdm input/output channel HBF filter coefficient
0x4AC	hbf_para_tap_coef_25	pdm input/output channel HBF filter coefficient
0x4B0	hbf_para_tap_coef_26	pdm input/output channel HBF filter coefficient
0x4B4	hbf_para_tap_coef_27	pdm input/output channel HBF filter coefficient
0x4B8	hbf_para_tap_coef_28	pdm input/output channel HBF filter coefficient
0x4BC	pdm_channel_enable	CFG
0x4C0	pdm2pcm_out_data	CFG
0x4C4	pdm_interrupt	CFG
0x640	pdm_in_clk_spike	CFG
0x644	pdm_in_ch_spike	CFG
0x648	pdm2pcm_filtr_bypass	CFG

11.1.5.3 Detail Registers Description

1. I2S Registers

IER, DW_apb_i2s Enable Register acts as a global enable/disable for DW_apb_i2s.

Offset Address: 0x400 Total Reset Value: 0x0000_0000

Bits	Name	Access	Description	Reset
[31:1]	RSVD_IER	R	RSVD_IER Reserved bits - Read Only Exists: Always	0x0000_0000
[0]	IEN	R/W	DW_apb_i2s enable. This bit enables or disables DW_apb_i2s. A disable on this bit overrides any other block or channel enables and flushes all FIFOs. Values: 0x0 (DISABLED): DW_apb_i2s disabled. 0x1 (ENABLED): DW_apb_i2s enabled Value Exists: Always	0x0

IRER, I2S Receiver Block Enable Register acts as an enable/disable for the DW_apb_i2s Receiver block.

Offset Address: 0x404 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	R	RSVD_IRER	RSVD_IRER Reserved bits - Read Only Exists: Always	0x0000_0000
[0]	R/W	RXEN	Receiver block enable. This bit enables or disables the receiver. A disable on this bit overrides any individual receive channel enables. Values: 0x0 (DISABLED): Receiver disabled 0x1 (ENABLED): Receiver enabled Exists: I2S_TRANSMITTER_BLOCK = 1	0x0

ITER, I2S Transmitter Block Enable Register acts as an enable/disable for the DW_apb_i2s Transmitter block.

Offset Address: 0x408 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	R	RSVD_ITER	RSVD_ITER Reserved bits - Read Only Exists: Always	0x0000_0000
[0]	R/W	TXEN	Transmitter block enable. This bit enables or disables the transmitter. A disable on this bit overrides any individual transmit channel enables. Values: 0x0 (DISABLED): Transmitter disabled 0x1 (ENABLED): Transmitter enabled Exists: I2S_TRANSMITTER_BLOCK = 1	0x0

CER, Clock Enable Register acts as an enable/disable for the DW_apb_i2s Clock Generation block| the clock signals sclk_en| which is only relevant in master mode (I2S_MODE_EN = 1). When this block is enabled| ws_out| and sclk_gate appear on the interface.

Offset Address: 0x40c Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	R	RSVD_CER	RSVD_CER Reserved bits - Read Only Exists: Always	0x0000_0000
[0]	R/W	CLKEN	"Clock generation enable/disable. This bit enables/disables the clock generation signals when DW_apb_i2s is a master: sclk_en	ws_out

CCR, Clock Configuration Register configures the ws_out and sclk_gate signals when DW_apb_i2s is a master.

Offset Address: 0x410 Total Reset Value: 0x0000_0010

Bits	Access	Name	Description	Reset
[31:5]	R	RSVD_CCR	RSVD_CCR Reserved bits - Read Only Exists: Always	0x000_0000

Bits	Access	Name	Description	Reset
[4:3]	R/W	WSS	These bits are used to program the number of sclk cycles for which the word select line (ws_out) stays in the left or right sample mode. The I2S Clock Generation block must be disabled (CER[0] = 0) prior to any changes in this value. Values: 0x0 (CLOCK_CYCLES_16): 16 sclk cycles; 0x1 (CLOCK_CYCLES_24): 24 sclk cycles; 0x2 (CLOCK_CYCLES_32): 32 sclk cycles; Exists: This register is only relevant when component is configured to be a master (I2S_MODE_EN = 1).	0x2
[2:0]	R/W	SCLKG	These bits are used to program the gating of sclk. The programmed gating value must be less than or equal to the largest configured/programmed audio resolution to prevent the truncating of RX/TX data. The I2S Clock Generation block must be disabled (CER[0] = 0) before making any changes in this value. Values: 0x0 (NO_CLOCK_GATING): Clock gating is disabled; 0x1 (CLOCK_CYCLES_12): Gating after 12 sclk cycles; 0x2 (CLOCK_CYCLES_16): Gating after 16 sclk cycles; 0x3 (CLOCK_CYCLES_20): Gating after 20 sclk cycles; 0x4 (CLOCK_CYCLES_24): Gating after 24 sclk cycles; Exists: This register is only relevant when component is configured to be a master (I2S_MODE_EN = 1).	0x0

[RXFFR, Receiver Block FIFO Reset Register specifies the Receiver Block FIFO Reset Register.](#)

Offset Address: 0x414 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	W	RSVD_RXFFR	RSVD_RXFFR Reserved bits - Write Only Exists: Always Volatile: true	0x0000_0000
[0]	W	RXFFR	Receiver FIFO Reset. Writing a 1 to this register flushes all the RX FIFOs (this is a self clearing bit). The Receiver Block must be disabled before writing to this bit. Values: 0x0 (NO_FLUSH): Does not flush the RX FIFO 0x1 (FLUSH): Flushes the RX FIFO Exists: I2S_RECEIVER_BLOCK = 1 Volatile: true	0x0

[TXFFR, Transmitter Block FIFO Reset Register specifies the Transmitter Block FIFO Reset Register.](#)

Offset Address: 0x418 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	W	RSVD_TXFFR	RSVD_TXFFR Reserved bits - Write Only Exists: Always Volatile: true	0x0000_0000
[0]	W	TXFFR	Transmitter FIFO Reset. Writing a 1 to this register flushes all the TX FIFOs (this is a self clearing bit). The Transmitter Block must be disabled prior to writing this bit. Values: 0x0 (NO_FLUSH): Does not flush the TX FIFO 0x1 (FLUSH): Flushes the TX FIFO Exists: I2S_TRANSMITTER_BLOCK = 1 Volatile: true	0x0

[LRBRx \(for x = 0; x <= 1\), left receive buffer register x specifies the register.](#)

Offset Address: 0x420 + 0x40*x Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	R	LRBRx	"The left stereo data received serially from the receive channel input (sdix). If the RX FIFO is full and the two-stage read operation (for instance, a read from LRBRx followed by a read from RRBRx) is not performed before the start of the next stereo pair, then the new data is lost and an overrun interrupt occurs. (data already in the RX FIFO is preserved.) Note: Before reading this register again, the right stereo data must be read from RRBRx or the status/interrupts will not be valid. Exists: I2S_RX_CHANNELS > x "	0x0000_0000

LTHR_x (for x = 0; x <= 1), left transmit holding register x specifies the register.

Offset Address: 0x420 + 0x40*x Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	W	LTHR _x	The left stereo data to be transmitted serially through the transmit channel output (sdox) is written through this register. Writing is a two-stage process: 1. A write to this register passes the left stereo sample to the transmitter. 2. This MUST be followed by writing the right stereo sample to the RTHR _x register. Data must only be written to the FIFO when it is not full. Any attempt to write to a full FIFO results in that data being lost and an overrun interrupt being generated. Exists: I2S_TX_CHANNELS > x	NA

RRBR_x (for x = 0; x <= 1), right transmit holding register x specifies the receive buffer register.

Offset Address: 0x424 + 0x40*x Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	R	RRBR _x	"The right stereo data received serially from the receive channel input (sdix) is read through this register. If the RX FIFO is full and the two-stage read operation (for instance, read from LRBR _x followed by a read from RRBR _x) is not performed before the start of the next stereo pair, then the new data is lost and an overrun interrupt occurs. (Data already in the RX FIFO is preserved.) Note: Prior to reading this register, the left stereo data MUST be read from LRBR _x , or the status/interrupts will not be valid. Exists: I2S_RX_CHANNELS > x Volatile: true Range Variable[x]: I2S_RX_WORDSIZE_x - 1"	0x0000_0000

RTHR_x (for x = 0; x <= 1), right transmit holding register x specifies the register.

Offset Address: 0x424 + 0x40*x

Bits	Access	Name	Description	Reset
[31:0]	W	RTHR _x	The right stereo data to be transmitted serially through the transmit channel output (sdox) is written through this register. Writing is a two-stage process: 1. A left stereo sample MUST be written to the LTHR _x register. 2. A write to this register passes the right stereo sample to the transmitter. Data should only be written to the FIFO when it is not full. Any attempt to write to a full FIFO results in that data being lost and an overrun interrupt being generated. Exists: I2S_TX_CHANNELS > x	NA

RER_x (for x = 0; x <= 1), receive enable register x specifies the register.

Offset Address: 0x428 + 0x40*x Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:1]	R	RSVD_RER _x	RSVD_RER _x Reserved bits - Read Only Exists: Always	0x0000_0000

Bits	Access	Name	Description	Reset
[0]	R/W	RXCHENx	"Receive channel enable. This bit enables/disables a receive channel, independently of all other channels. On enable, the channel begins receiving on the next left stereo cycle. A global disable of DW_apb_i2s (IER[0] = 0) or the Receiver block (IRER[0] = 0) overrides this value. Values: 0x0 (DISABLED): Receive Channel Disable 0x1 (ENABLED): Receive Channel Enable Exists: I2S_RX_CHANNELS > x"	0x1

TERx (for x = 0; x <= 1), transmit enable register x specifies the register.

Offset Address: 0x42C + 0x40*x Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:1]	R	RSVD_TERx	RSVD_TERx Reserved bits - Read Only Exists: Always	0x0000_0000
[0]	R/W	TXCHENx	"Transmit channel enable. This bit enables/disables a transmit channel, independently of all other channels. On enable, the channel begins transmitting on the next left stereo cycle. A global disable of DW_apb_i2s (IER[0] = 0) or Transmitter block (ITER[0] = 0) overrides this value. Values: 0x0 (DISABLED): Transmit Channel Disable 0x1 (ENABLED): Transmit Channel Enable Exists: I2S_TX_CHANNELS > x"	0x1

RCRx (for x = 0; x <= 1), receive configuration register x specifies the register.

Offset Address: 0x430 + 0x40*x Total Reset Value: 0x0000_0005

Bits	Access	Name	Description	Reset
[31:3]	R	RSVD_RCRx	RSVD_RCRx Reserved bits - Read Only Exists: Always	0x0000_0000
[2:0]	R/W	WLEN	"These bits are used to program the desired data resolution of the receiver and enables the LSB of the incoming left (or right) word to be placed in the LSB of the LRBRx (or RRBRx) register. Programmed data resolution must be less than or equal to I2S_RX_WORDSIZE_x. If the selected resolution is greater than the I2S_RX_WORDSIZE_x, the receive channel defaults back to I2S_RX_WORDSIZE_RESET_VALUE_x. The channel must be disabled prior to any changes in this value (RER0[0] = 0). Values: 0x0 (IGNORE_WORD_LENGTH): Ignore the word length 0x1 (RESOLUTION_12_BIT): 12-bit data resolution of the receiver. 0x2 (RESOLUTION_16_BIT): 16-bit data resolution of the receiver. 0x3 (RESOLUTION_20_BIT): 20-bit data resolution of the receiver. 0x4 (RESOLUTION_24_BIT): 24-bit data resolution of the receiver. 0x5 (RESOLUTION_32_BIT): 32-bit data resolution of the receiver. Exists: I2S_RX_CHANNELS > x"	0x5

TCRx (for x = 0; x <= 1), transmit configuration register x specifies the register.

Offset Address: 0x434 + 0x40*x Total Reset Value: 0x0000_0005

Bits	Access	Name	Description	Reset
[31:3]	R	RSVD_TCRx	RSVD_TCRx Reserved bits - Read Only Exists: Always	0x0000_0000

Bits	Access	Name	Description	Reset
[2:0]	R/W	WLEN	"These bits are used to program the data resolution of the transmitter and ensures the MSB of the data is transmitted first. Programmed resolution must be less than or equal to I2S_TX_WORDSIZE_x. If the selected resolution is greater than I2S_TX_WORDSIZE_x, the transmit channel defaults back to I2S_TX_WORDSIZE_RESET_VALUE_x value. The channel must be disabled prior to any changes in this value (TER0[0] = 0). Values: 0x0 (IGNORE_WORD_LENGTH): Ignore the word length 0x1 (RESOLUTION_12_BIT): 12-bit data resolution of the transmitter. 0x2 (RESOLUTION_16_BIT): 16-bit data resolution of the transmitter. 0x3 (RESOLUTION_20_BIT): 20-bit data resolution of the transmitter. 0x4 (RESOLUTION_24_BIT): 24-bit data resolution of the transmitter. 0x5 (RESOLUTION_32_BIT): 32-bit data resolution of the transmitter. Exists: I2S_TX_CHANNELS > x"	0x5

ISR_x (for x = 0; x <= 1), interrupt status register x specifies the register.

Offset Address: 0x438 + 0x40*x Total Reset Value: 0x0000_0010

Bits	Access	Name	Description	Reset
[31:6]	R	RSVD31_6	RSVD31_6 Reserved bits - Read Only Exists: Always Volatile: true	0x000_0000
[5]	R	TXFO	Status of Data Overrun interrupt for the TX channel. This bit specifies whether the TX FIFO write is valid or an overrun. Attempt to write to full TX FIFO. Values: 0x0 (WRITE_VALID): TX FIFO write valid 0x1 (WRITE_OVERRUN): TX FIFO write overrun Exists: (I2S_TX_CHANNELS>x && I2S_TRANSMITTER_BLOCK==1) Volatile: true	0x0
[4]	R	TXFE	Status of Transmit Empty Trigger interrupt. This bit specifies whether the TX FIFO trigger level has reached or not. TX FIFO is empty. Values: 0x0 (REACHED_TRIGGER_LEVEL): TX FIFO trigger level is reached 0x1 (NOT_REACHED): TX FIFO trigger level is not reached Exists: (I2S_TX_CHANNELS>x && I2S_TRANSMITTER_BLOCK==1) Volatile: true	0x1
[3:2]	R	RSVD3_2	RSVD3_2 Reserved bits - Read Only Exists: Always Volatile: true	0x0
[1]	R	RXFO	Status of Data Overrun interrupt for the RX channel. Incoming data lost due to a full RX FIFO. Values: 0x0 (WRITE_VALID): RX FIFO write valid 0x1 (WRITE_OVERRUN): RX FIFO write overrun Exists: (I2S_RX_CHANNELS>x && I2S_RECEIVER_BLOCK==1) Volatile: true	0x0
[0]	R	RXDA	Status of Receive Data Available interrupt. This bit denotes the status of the RX FIFO trigger level. Values: 0x1 (REACHED_TRIGGER_LEVEL): RX FIFO trigger level is reached 0x0 (NOT_REACHED): RX FIFO trigger level is not reached Exists: (I2S_RX_CHANNELS>x && I2S_RECEIVER_BLOCK==1) Volatile: true	0x0

IMRx (for x = 0; x <= 1), interrupt mask register x specifies the register.

Offset Address: 0x43C + 0x40*x Total Reset Value: 0x0000_0033

Bits	Access	Name	Description	Reset
[31:6]	R	RSVD_IMR0_6_31	RSVD_IMR0_6_31 Reserved bits - Read Only Exists: Always	0x000_0000
[5]	R/W	TXFOM	Mask TX FIFO Overrun interrupt. This bit masks or unmask a TX FIFO overrun interrupt. Values: 0x1 (MASK_INTERRUPT): Masks TX FIFO Overrun interrupt 0x0 (UNMASK_INTERRUPT): Unmasks TX FIFO Overrun interrupt	0x1
[4]	R/W	TXFEM	Mask TX FIFO Empty interrupt. This bit masks or unmask a TX FIFO Empty interrupt. Values: 0x1 (MASK_INTERRUPT): Masks TX FIFO Empty interrupt 0x0 (UNMASK_INTERRUPT): Unmasks TX FIFO Empty interrupt Exists: Always	0x1
[3:2]	R	RSVD_IMR0_2_3	RSVD_IMR0_2_3 Reserved bits - Read Only Exists: Always	0x0
[1]	R/W	RXFOM	Mask RX FIFO Overrun interrupt. This bit masks or unmask an RX FIFO Overrun interrupt. Values: 0x1 (MASK_INTERRUPT): Masks RX FIFO Overrun interrupt 0x0 (UNMASK_INTERRUPT): Unmasks RX FIFO Overrun interrupt	0x1
[0]	R/W	RXDAM	Mask RX FIFO Data Available interrupt. This bit masks or unmask an RX FIFO Data Available interrupt. Values: 0x1 (MASK_INTERRUPT): Masks RX FIFO data available interrupt 0x0 (UNMASK_INTERRUPT): Unmasks RX FIFO data available interrupt	0x1

RORx (for x = 0; x <= 1), receive overrun register x specifies the register.

Offset Address: 0x440 + 0x40*x Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	R	RSVD_RORx	"RSVD_RORx Reserved bits - Read Only Exists: Always Volatile: true"	0x0000_0000
[0]	R	RXCHO	"Read this bit to clear the RX FIFO Data Overrun interrupt. Values: 0x0 (WRITE_VALID): RX FIFO write valid 0x1 (WRITE_OVERRUN): RX FIFO write overrun Exists: I2S_RX_CHANNELS > x Volatile: true"	0x0

TORx (for x = 0; x <= 1), transmit overrun register x specifies the register.

Offset Address: 0x444 + 0x40*x Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	R	RSVD_TORx	"RSVD_TORx Reserved bits - Read Only Exists: Always Volatile: true"	0x0000_0000

Bits	Access	Name	Description	Reset
[0]	R	TXCHO	"Read this bit to clear the TX FIFO Data Overrun interrupt. Values: 0x0 (WRITE_VALID): TX FIFO write valid 0x1 (WRITE_OVERRUN): TX FIFO write overrun Exists: I2S_TX_CHANNELS > x Volatile: true"	0x0

RFCRx (for x = 0; x <= 1), receive fifo configuration register specifies the register.

Offset Address: 0x448 + 0x40*x Total Reset Value: 0x0000_0003

Bits	Access	Name	Description	Reset
[31:4]	R	RSVD_RFCRx	RSVD_RFCRx Reserved bits - Read Only Exists: Always	0x000_0000
[3:0]	R/W	RXCHDT	"These bits program the trigger level in the RX FIFO at which the Received Data Available interrupt is generated. Trigger Level = Programmed Value + 1 (for example, 1 to I2S_RX_FIFO_DEPTH_x) Valid RXCHDT values: 0 to (I2S_RX_FIFO_x - 1) If an illegal value is programmed, these bits saturate to (I2S_RX_FIFO_x - 1). The channel must be disabled prior to any changes in this value (that is, RER0[0] = 0). Values: 0x0 (TRIGGER_LEVEL_1): Interrupt trigger when FIFO level is 1. 0x1 (TRIGGER_LEVEL_2): Interrupt trigger when FIFO level is 2. 0x2 (TRIGGER_LEVEL_3): Interrupt trigger when FIFO level is 3. 0x3 (TRIGGER_LEVEL_4): Interrupt trigger when FIFO level is 4. 0x4 (TRIGGER_LEVEL_5): Interrupt trigger when FIFO level is 5. 0x5 (TRIGGER_LEVEL_6): Interrupt trigger when FIFO level is 6. 0x6 (TRIGGER_LEVEL_7): Interrupt trigger when FIFO level is 7. 0x7 (TRIGGER_LEVEL_8): Interrupt trigger when FIFO level is 8. 0x8 (TRIGGER_LEVEL_9): Interrupt trigger when FIFO level is 9. 0x9 (TRIGGER_LEVEL_10): Interrupt trigger when FIFO level is 10. 0xa (TRIGGER_LEVEL_11): Interrupt trigger when FIFO level is 11. 0xb (TRIGGER_LEVEL_12): Interrupt trigger when FIFO level is 12. 0xc (TRIGGER_LEVEL_13): Interrupt trigger when FIFO level is 13. 0xd (TRIGGER_LEVEL_14): Interrupt trigger when FIFO level is 14. 0xe (TRIGGER_LEVEL_15): Interrupt trigger when FIFO level is 15. 0xf (TRIGGER_LEVEL_16): Interrupt trigger when FIFO level is 16. Exists: I2S_RX_CHANNELS > x"	0x3

TFCRx (for x = 0; x <= 1), transmit fifo configuration register x specifies the register.

Offset Address: 0x44C + 0x40*x Total Reset Value: 0x0000_0003

Bits	Access	Name	Description	Reset
[31:4]	R	RSVD_TFCRx	RSVD_TFCRx Reserved bits - Read Only Exists: Always	0x000_0000

Bits	Access	Name	Description	Reset
[3:0]	R/W	TXCHET	"These bits program the trigger level in the TX FIFO at which the Empty Threshold Reached Interrupt is generated. Trigger Level = TXCHET TXCHET values: 0 to (I2S_TX_FIFO_x - 1) If an illegal value is programmed, these bits saturate to (I2S_TX_FIFO_x - 1). The channel must be disabled prior to any changes in this value (that is, TER0[0] = 0). Values: 0x0 (TRIGGER_LEVEL_1): Interrupt trigger when FIFO level is 1. 0x1 (TRIGGER_LEVEL_2): Interrupt trigger when FIFO level is 2. 0x2 (TRIGGER_LEVEL_3): Interrupt trigger when FIFO level is 3. 0x3 (TRIGGER_LEVEL_4): Interrupt trigger when FIFO level is 4. 0x4 (TRIGGER_LEVEL_5): Interrupt trigger when FIFO level is 5. 0x5 (TRIGGER_LEVEL_6): Interrupt trigger when FIFO level is 6. 0x6 (TRIGGER_LEVEL_7): Interrupt trigger when FIFO level is 7. 0x7 (TRIGGER_LEVEL_8): Interrupt trigger when FIFO level is 8. 0x8 (TRIGGER_LEVEL_9): Interrupt trigger when FIFO level is 9. 0x9 (TRIGGER_LEVEL_10): Interrupt trigger when FIFO level is 10. 0xa (TRIGGER_LEVEL_11): Interrupt trigger when FIFO level is 11. 0xb (TRIGGER_LEVEL_12): Interrupt trigger when FIFO level is 12. 0xc (TRIGGER_LEVEL_13): Interrupt trigger when FIFO level is 13. 0xd (TRIGGER_LEVEL_14): Interrupt trigger when FIFO level is 14. 0xe (TRIGGER_LEVEL_15): Interrupt trigger when FIFO level is 15. 0xf (TRIGGER_LEVEL_16): Interrupt trigger when FIFO level is 16. Exists: I2S_TX_CHANNELS > x"	0x3

RFFx (for x = 0; x <= 1), receive fifo flush register x specifies the register.

Offset Address: 0x450 + 0x40*x Total Reset Value: NA

Bits	Access	Name	Description	Reset
[31:1]	W	RSVD_RFFx	RSVD_RFFx Reserved bits - Write Only Exists: Always Volatile: true	NA
[0]	W	RXCHFR	Receive Channel FIFO Reset. Writing a 1 to this register flushes an individual RX FIFO (This is a self clearing bit.). A Rx channel or block must be disabled prior to writing to this bit. Values: 0x0 (NO_FLUSH): Does not flush an individual RX FIFO 0x1 (FLUSH): Flushes an individual RX FIFO Exists: I2S_RX_CHANNELS > x Volatile: true	NA

TFFx (for x = 0; x <= 1), transmit fifo flush register x specifies the register.

Offset Address: 0x454 + 0x40*x Total Reset Value: NA

Bits	Access	Name	Description	Reset
[31:1]	W	RSVD_TFFx	RSVD_TFFx Reserved bits - Write Only Exists: Always Volatile: true	NA
[0]	W	TXCHFR	Transmit Channel FIFO Reset. Writing a 1 to this register flushes channels TX FIFO (This is a self clearing bit.). The TX channel or block must be disabled prior to writing to this bit. Values: 0x0 (NO_FLUSH): Do not flushes channels TX FIFO. 0x1 (FLUSH): Flushes channels TX FIFO. Exists: I2S_TX_CHANNELS > x Volatile: true	NA

RXDMA, Receiver Block DMA Register allows access to all enabled Receive channels via a single point rather than through the LRBRx and RRB Rx registers. The Receive channels are targeted in a cyclical fashion (starting at the

lowest numbered enabled channel) and takes two reads (left and right stereo data) before the component points to the next channel.

The following example describes the behavior of this register for a component that has been configured with four Receive channels where Channels 0 and 3 are enabled:

Order of returned read data:

1. Ch0 - Left Data
2. Ch0 - Right Data
3. Ch3 - Left Data
4. Ch3 - Right Data
5. Ch0 - Left Data
6. Ch0 - Right Data and so on

Note: There is no read coherency logic; hence the APB_DATA_WIDTH must be greater than or equal to the largest Receive channel word size to ensure all half data pairs can be accessed using a single read.

Channels can be enabled or disabled during the read cycles; however DW_apb_i2s does not support disabling a channel in the middle of a stereo pair.

Offset Address: 0x5c0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RXDMA	R	"Receiver Block DMA Register. These bits are used to cycle repeatedly through the enabled receive channels (from lowest numbered to highest), reading stereo data pairs. Exists: I2S_RECEIVER_BLOCK = 1 Volatile: true"	0x0000_0000

RRXDMA, Reset Receiver Block DMA Register is The RXDMA can be reset to the lowest enabled Channel via the RRXDMA register. The RRXDMA register can be written to at any stage of the RXDMAs read cycle" however, it has no effect when the component is in the middle of a stereo pair read. The following example describes the operation of this register for a system with four Receive channels, where channels 0,1,2, and 3 are enabled.

Order of returned read data:

1. Ch0 - Left Data
2. Ch0 - Right Data
3. RRXDMA Reset
4. Ch0 - Left Data
5. Ch0 - Right Data
6. Ch1 - Left Data
7. RRXDMA Reset - No effect (read not complete)
8. Ch1 - Right Data etc.
9. Ch2 - Left Data
10. Ch2 - Right Data
11. RRXDMA Reset
12. Ch0 - Left Data
13. Ch0 - Right Data

Offset Address: 0x5c4 Total Reset Value: NA

Bits	Access	Name	Description	Reset
[31:1]	W	RSVD_RRXDMA	RSVD_RRXDMA Reserved bits - Write Only Exists: Always Volatile: true	NA

Bits	Access	Name	Description	Reset
[0]	W	RRXDMA	"Reset Receiver Block DMA Register. Writing a 1 to this self-clearing register resets the RXDMA register mid-cycle to point to the lowest enabled Receive channel. Note: Writing to this register has no effect if the component is performing a stereo pair read (such as, when left stereo data has been read but not right stereo data). Values: 0x1 (RESET): Reset Receiver Block DMA Register Exists: I2S_RECEIVER_BLOCK = 1 and I2S_RX_CHANNELS > 1 Volatile: true"	NA

TXDMA, Transmitter Block DMA Register is The TXDMA register functions similar to the RXDMA register and allows write accesses to all of the enabled Transmit channels via a single point rather than through the LTHRx and RTHRx registers.

Note: There is no write coherency logic the APB_DATA_WIDTH must be greater than or equal to the largest Transmit channel word size to ensure all half data pairs can be written using a single write.

Channels can be enabled or disabled during the write cycles; however DW_apb_i2s does not support disabling a channel in the middle of a stereo pair.

Offset Address: 0x5c8 Total Reset Value: NA

Bits	Access	Name	Description	Reset
[31:0]	W	TXDMA	Transmitter Block DMA Register. The register bits can be used to cycle repeatedly through the enabled Transmit channels (from lowest numbered to highest) to allow writing of stereo data pairs. Exists: I2S_TRANSMITTER_BLOCK = 1 Volatile: true	NA

RTXDMA, Reset Transmitter Block DMA Register provides the same functionality as the RRXDMA register but targets TXDMA instead.

Offset Address: 0x5cc Total Reset Value: NA

Bits	Access	Name	Description	Reset
[31:1]	W	RSVD_RTXDMA	RSVD_RTXDMA Reserved bits - Write Only Exists: Always Volatile: true	NA
[0]	W	RTXDMA	"Reset Transmitter Block DMA Register. Writing a 1 to this self-clearing register resets the TXDMA register mid-cycle to point to the lowest enabled Transmit channel. Note: This register has no effect in the middle of a stereo pair write (such as, when left stereo data has been written but not right stereo data). Values: 0x1 (RESET): Reset Transmitter Block DMA Register Exists: I2S_TRANSMITTER_BLOCK = 1 and I2S_TX_CHANNELS > 1 Volatile: true"	NA

2. PDM Registers

PDM_CONTROL_REG_0 is PDM input configuration register.

Offset Address: 0x400, Total Reset Value: 0x0000_0000

bits	Access	Name	Description	Reset
[31:17]	RO	RSV	Reserved bit, return 0s when read.	0x0
[16]	RO	pdm_buf_rdy	buffer data ready can be read	0x0
[15]	RO	RSV	Reserved bit, return 0s when read.	0x0

bits	Access	Name	Description	Reset
[14]	RW	pdm_clk_sel	pdm_clk invert control signal: 1'b0: pdm_clk is same with pdm_clk_root 1'b1: pdm_clk is inverted of pdm_clk_root	0x0
[13]	RW	audio_dev_clk_sel	audio_clk invert control signal: 1'b0: audio_clk same as audio_dev_clk 1'b1: audio_clk is inverter of audio_dev_clk	0x0
[12:8]	RW	pdm_fifo_th	pdm fifo threshold for making dma request. Should match cfg in DMA.	0x0
[7:6]	RW	audio_in_mode	pdm_in0 and i2s_sdo1 multiplex IO37, 2'b10: IO37 is i2s_sdo1, others: IO37 is pdm_in0; pdm_in1 and i2s_sdo0 multiplex IO35, 2'b10: IO37 is i2s_sdo0, others: IO37 is pdm_in1.	0x0
[5]	RW	aud_codec_bypass	bypass audio codec, use I2S directly to IO for digital I2S microphone	0x0
[4]	RW	pdm_fifo_clr	pdm fifo clear	0x0
[3]	RW	pdm_out_endian	pdm output endian select, 28bits valid data location in 32bits data pdm2pcm_out_data: 1'b0: high 28bit(4bit channel number, 24bit pcm data) 1'b1: low 28bit(4bit channel number, 24bit pcm data)	0x0
[2]	RW	hbf_enable	hbf filter enable,hbf filter enable 0: disable 1: enable	0x0
[1]	RW	mpf_enable	mpf filter enable,cic compensate filter enable 0: disable 1: enable	0x0
[0]	RW	pdm_en	pdm module enable 0: disable 1: enable	0x0

PDM_CONTROL_REG_1 is pdm oversampling rate control register.

Offset Address: 0x404, Total Reset Value: 0x0000_0000

bits	Access	Name	Description	Reset
[31:1]	RO	RSV	Reserved bit, return 0s when read.	0x0000_0000
[0]	RW	ord_sel	cic filter order select: 0 for 16; 1 for 32	0x0

mp_para_tap_coef_0 is cic compensate filter coefficient.

Offset Address: 0x408, Total Reset Value: 0x00ff_fcc5

bits	Access	Name	Description	Reset
[31:24]	RO	RSV	Reserved bit, return 0s when read.	0x00
[23:0]	WO	mp_para_tap_coef_0	mp_para_tap_coef_0,pdm input cic compensate filter coefficient	0xffcc5

mp_para_tap_coef_1 is cic compensate filter coefficient.

Offset Address: 0x40C, Total Reset Value: 0x00ff_e5aa

bits	Access	Name	Description	Reset
[31:24]	RO	RSV	Reserved bit, return 0s when read.	0x00
[23:0]	WO	mp_para_tap_coef_1	mp_para_tap_coef_1,pdm input cic compensate filter coefficient	0xffe5aa

mp_para_tap_coef_2 is cic compensate filter coefficient.

Offset Address: 0x410, Total Reset Value: 0x00ff_f1f0

bits	Access	Name	Description	Reset

bits	Access	Name	Description	Reset
[31:24]	RO	RSV	Reserved bit, return 0s when read.	0x00
[23:0]	WO	mp_para_tap_coef_2	mp_para_tap_coef_2,pdm input cic compensate filter coefficient	0xffff1f0

mp_para_tap_coef_3 is cic compensate filter coefficient.

Offset Address: 0x414, Total Reset Value: 0x0000_689f

bits	Access	Name	Description	Reset
[31:24]	RO	RSV	Reserved bit, return 0s when read.	0x00
[23:0]	WO	mp_para_tap_coef_3	mp_para_tap_coef_3,pdm input cic compensate filter coefficient	0x00689f

mp_para_tap_coef_4 is cic compensate filter coefficient.

Offset Address: 0x418, Total Reset Value: 0x0000_1e72

bits	Access	Name	Description	Reset
[31:24]	RO	RSV	Reserved bit, return 0s when read.	0x00
[23:0]	WO	mp_para_tap_coef_4	mp_para_tap_coef_4,pdm input cic compensate filter coefficient	0x001e72

mp_para_tap_coef_5 is cic compensate filter coefficient.

Offset Address: 0x41c, Total Reset Value: 0x00fe_c0ab

bits	Access	Name	Description	Reset
[31:24]	RO	RSV	Reserved bit, return 0s when read.	0x00
[23:0]	WO	mp_para_tap_coef_5	mp_para_tap_coef_5,pdm input cic compensate filter coefficient	0xfec0ab

mp_para_tap_coef_6 is cic compensate filter coefficient.

Offset Address: 0x420, Total Reset Value: 0x0000_1b7d

bits	Access	Name	Description	Reset
[31:24]	RO	RSV	Reserved bit, return 0s when read.	0x00
[23:0]	WO	mp_para_tap_coef_6	mp_para_tap_coef_6,pdm input cic compensate filter coefficient	0x001b7d

mp_para_tap_coef_7 is cic compensate filter coefficient.

Offset Address: 0x424, Total Reset Value: 0x0002_f9f2

bits	Access	Name	Description	Reset
[31:24]	RO	RSV	Reserved bit, return 0s when read.	0x00
[23:0]	WO	mp_para_tap_coef_7	mp_para_tap_coef_7,pdm input cic compensate filter coefficient	0x02f9f2

mp_para_tap_coef_8 is cic compensate filter coefficient.

Offset Address: 0x428, Total Reset Value: 0x00fe_9b02

bits	Access	Name	Description	Reset
[31:24]	RO	RSV	Reserved bit, return 0s when read.	0x00
[23:0]	WO	mp_para_tap_coef_8	mp_para_tap_coef_8,pdm input cic compensate filter coefficient	0xfe9b02

mp_para_tap_coef_9 is cic compensate filter coefficient.

Offset Address: 0x42C, Total Reset Value: 0x00fa_331b

bits	Access	Name	Description	Reset
[31:24]	RO	RSV	Reserved bit, return 0s when read.	0x00
[23:0]	WO	mp_para_tap_coef_9	mp_para_tap_coef_9,pdm input cic compensate filter coefficient	0xfa331b

mp_para_tap_coef_10 is cic compensate filter coefficient.

Offset Address: 0x430, Total Reset Value: 0x0005_2b14

bits	Access	Name	Description	Reset
[31:24]	RO	RSV	Reserved bit, return 0s when read.	0x00
[23:0]	WO	mp_para_tap_coef_10	mp_para_tap_coef_10,pdm input cic compensate filter coefficient	0x052b14

mp_para_tap_coef_11 is cic compensate filter coefficient.

Offset Address: 0x434, Total Reset Value: 0x0009_c3ec

bits	Access	Name	Description	Reset
[31:24]	RO	RSV	Reserved bit, return 0s when read.	0x00
[23:0]	WO	mp_para_tap_coef_11	mp_para_tap_coef_11,pdm input cic compensate filter coefficient	0x09c3ec

mp_para_tap_coef_12 is cic compensate filter coefficient.

Offset Address: 0x438, Total Reset Value: 0x00f1_a38d

bits	Access	Name	Description	Reset
[31:24]	RO	RSV	Reserved bit, return 0s when read.	0x00
[23:0]	WO	mp_para_tap_coef_13	mp_para_tap_coef_12,pdm input cic compensate filter coefficient	0xf1a38d

mp_para_tap_coef_13 is cic compensate filter coefficient.

Offset Address: 0x43c, Total Reset Value: 0x00ef_4245

bits	Access	Name	Description	Reset
[31:24]	RO	RSV	Reserved bit, return 0s when read.	0x00
[23:0]	WO	mp_para_tap_coef_13	mp_para_tap_coef_13,pdm input cic compensate filter coefficient	0xef4245

mp_para_tap_coef_14 is cic compensate filter coefficient.

Offset Address: 0x440, Total Reset Value: 0x002a_6908

bits	Access	Name	Description	Reset
[31:24]	RO	RSV	Reserved bit, return 0s when read.	0x00
[23:0]	WO	mp_para_tap_coef_14	mp_para_tap_coef_14,pdm input cic compensate filter coefficient	0x2a6908

mp_para_tap_coef_15 is cic compensate filter coefficient.

Offset Address: 0x444, Total Reset Value: 0x0055_7221

bits	Access	Name	Description	Reset
[31:24]	RO	RSV	Reserved bit, return 0s when read.	0x00

bits	Access	Name	Description	Reset
[23:0]	WO	mp_para_tap_coef_15	mp_para_tap_coef_15,pdm input cic compensate filter coefficient	0x557221

[hbf_para_tap_coef_0](#) is hbf filter coefficient.

Offset Address: 0x448, Total Reset Value: 0x00ff_fe94

bits	Access	Name	Description	Reset
[31:24]	RO	RSV	Reserved bit, return 0s when read.	0x00
[23:0]	WO	hbf_para_tap_coef_0	hbf_para_tap_coef_0,pdm input hbf filter coefficient	0xfffe94

[hbf_para_tap_coef_1](#) is hbf filter coefficient.

Offset Address: 0x44c, Total Reset Value: 0x0000_0236

bits	Access	Name	Description	Reset
[31:24]	RO	RSV	Reserved bit, return 0s when read.	0x00
[23:0]	WO	hbf_para_tap_coef_1	hbf_para_tap_coef_1,pdm input hbf filter coefficient	0x000236

[hbf_para_tap_coef_2](#) is hbf filter coefficient.

Offset Address: 0x450, Total Reset Value: 0x00ff_fca6

bits	Access	Name	Description	Reset
[31:24]	RO	RSV	Reserved bit, return 0s when read.	0x00
[23:0]	WO	hbf_para_tap_coef_2	hbf_para_tap_coef_2,pdm input hbf filter coefficient	0xfffca6

[hbf_para_tap_coef_3](#) is hbf filter coefficient.

Offset Address: 0x454, Total Reset Value: 0x0000_04fa

bits	Access	Name	Description	Reset
[31:24]	RO	RSV	Reserved bit, return 0s when read.	0x00
[23:0]	WO	hbf_para_tap_coef_3	hbf_para_tap_coef_3,pdm input hbf filter coefficient	0x0004fa

[hbf_para_tap_coef_4](#) is hbf filter coefficient.

Offset Address: 0x458, Total Reset Value: 0x00ff_f8c1

bits	Access	Name	Description	Reset
[31:24]	RO	RSV	Reserved bit, return 0s when read.	0x00
[23:0]	WO	hbf_para_tap_coef_4	hbf_para_tap_coef_4,pdm input hbf filter coefficient	0xfff8c1

[hbf_para_tap_coef_5](#) is hbf filter coefficient.

Offset Address: 0x45c, Total Reset Value: 0x0000_0a5c

bits	Access	Name	Description	Reset
[31:24]	RO	RSV	Reserved bit, return 0s when read.	0x00
[23:0]	WO	hbf_para_tap_coef_5	hbf_para_tap_coef_5,pdm input hbf filter coefficient	0x000a5c

[hbf_para_tap_coef_6](#) is hbf filter coefficient.

Offset Address: 0x460, Total Reset Value: 0x00ff_f172

bits	Access	Name	Description	Reset
[31:24]	RO	RSV	Reserved bit, return 0s when read.	0x00
[23:0]	WO	hbf_para_tap_coef_6	hbf_para_tap_coef_6,pdm input hbf filter coefficient	0xff172

[hbf_para_tap_coef_7](#) is hbf filter coefficient.

Offset Address: 0x464, Total Reset Value: 0x0000_141d

bits	Access	Name	Description	Reset
[31:24]	RO	RSV	Reserved bit, return 0s when read.	0x00
[23:0]	WO	hbf_para_tap_coef_7	hbf_para_tap_coef_7,pdm input hbf filter coefficient	0x00141d

[hbf_para_tap_coef_8 is hbf filter coefficient.](#)

Offset Address: 0x468, Total Reset Value: 0x00ff_e498

bits	Access	Name	Description	Reset
[31:24]	RO	RSV	Reserved bit, return 0s when read.	0x00
[23:0]	WO	hbf_para_tap_coef_8	hbf_para_tap_coef_8,pdm input hbf filter coefficient	0xffe498

[hbf_para_tap_coef_9 is hbf filter coefficient.](#)

Offset Address: 0x46c, Total Reset Value: 0x0000_24ca

bits	Access	Name	Description	Reset
[31:24]	RO	RSV	Reserved bit, return 0s when read.	0x00
[23:0]	WO	hbf_para_tap_coef_9	hbf_para_tap_coef_9,pdm input hbf filter coefficient	0x0024ca

[hbf_para_tap_coef_10 is hbf filter coefficient.](#)

Offset Address: 0x470, Total Reset Value: 0x00ff_cf3c

bits	Access	Name	Description	Reset
[31:24]	RO	RSV	Reserved bit, return 0s when read.	0x00
[23:0]	WO	hbf_para_tap_coef_10	hbf_para_tap_coef_10,pdm input hbf filter coefficient	0xffcf3c

[hbf_para_tap_coef_11 is hbf filter coefficient.](#)

Offset Address: 0x474, Total Reset Value: 0x0000_3fdc

bits	Access	Name	Description	Reset
[31:24]	RO	RSV	Reserved bit, return 0s when read.	0x00
[23:0]	WO	hbf_para_tap_coef_11	hbf_para_tap_coef_11,pdm input hbf filter coefficient	0x003fdc

[hbf_para_tap_coef_12 is hbf filter coefficient.](#)

Offset Address: 0x478, Total Reset Value: 0x00ff_ad4d

bits	Access	Name	Description	Reset
[31:24]	RO	RSV	Reserved bit, return 0s when read.	0x00
[23:0]	WO	hbf_para_tap_coef_12	hbf_para_tap_coef_12,pdm input hbf filter coefficient	0xffad4d

[hbf_para_tap_coef_13 is hbf filter coefficient.](#)

Offset Address: 0x47c, Total Reset Value: 0x0000_6a04

bits	Access	Name	Description	Reset
[31:24]	RO	RSV	Reserved bit, return 0s when read.	0x00
[23:0]	WO	hbf_para_tap_coef_13	hbf_para_tap_coef_13,pdm input hbf filter coefficient	0x006a04

[hbf_para_tap_coef_14 is hbf filter coefficient.](#)

Offset Address: 0x480, Total Reset Value: 0x00ff_7954

bits	Access	Name	Description	Reset
[31:24]	RO	RSV	Reserved bit, return 0s when read.	0x00
[23:0]	WO	hbf_para_tap_coef_14	hbf_para_tap_coef_14,pdm input hbf filter coefficient	0xff7954

hbf_para_tap_coef_15 is hbf filter coefficient.

Offset Address: 0x484, Total Reset Value: 0x0000_a9bb

bits	Access	Name	Description	Reset
[31:24]	RO	RSV	Reserved bit, return 0s when read.	0x00
[23:0]	WO	hbf_para_tap_coef_15	hbf_para_tap_coef_15,pdm input hbf filter coefficient	0x00a9bb

hbf_para_tap_coef_16 is hbf filter coefficient.

Offset Address: 0x488, Total Reset Value: 0x00ff_2b83

bits	Access	Name	Description	Reset
[31:24]	RO	RSV	Reserved bit, return 0s when read.	0x00
[23:0]	WO	hbf_para_tap_coef_16	hbf_para_tap_coef_16,pdm input hbf filter coefficient	0xff2b83

hbf_para_tap_coef_17 is hbf filter coefficient.

Offset Address: 0x48c, Total Reset Value: 0x0001_089d

bits	Access	Name	Description	Reset
[31:24]	RO	RSV	Reserved bit, return 0s when read.	0x00
[23:0]	WO	hbf_para_tap_coef_17	hbf_para_tap_coef_17,pdm input hbf filter coefficient	0x01089d

hbf_para_tap_coef_18 is hbf filter coefficient.

Offset Address: 0x490, Total Reset Value: 0x00fe_b791

bits	Access	Name	Description	Reset
[31:24]	RO	RSV	Reserved bit, return 0s when read.	0x00
[23:0]	WO	hbf_para_tap_coef_18	hbf_para_tap_coef_18,pdm input hbf filter coefficient	0xfeb791

hbf_para_tap_coef_19 is hbf filter coefficient.

Offset Address: 0x494, Total Reset Value: 0x0001_9729

bits	Access	Name	Description	Reset
[31:24]	RO	RSV	Reserved bit, return 0s when read.	0x00
[23:0]	WO	hbf_para_tap_coef_19	hbf_para_tap_coef_19,pdm input hbf filter coefficient	0x019729

hbf_para_tap_coef_20 is hbf filter coefficient.

Offset Address: 0x498, Total Reset Value: 0x00fe_0641

bits	Access	Name	Description	Reset
[31:24]	RO	RSV	Reserved bit, return 0s when read.	0x00
[23:0]	WO	hbf_para_tap_coef_20	hbf_para_tap_coef_20,pdm input hbf filter coefficient	0xfe0641

hbf_para_tap_coef_21 is hbf filter coefficient.

Offset Address: 0x49c, Total Reset Value: 0x0002_7807

bits	Access	Name	Description	Reset
[31:24]	RO	RSV	Reserved bit, return 0s when read.	0x00
[23:0]	WO	hbf_para_tap_coef_21	hbf_para_tap_coef_21,pdm input hbf filter coefficient	0x027807

hbf_para_tap_coef_22 is hbf filter coefficient.

Offset Address: 0x4A0, Total Reset Value: 0x00fc_e06b

bits	Access	Name	Description	Reset
[31:24]	RO	RSV	Reserved bit, return 0s when read.	0x00

bits	Access	Name	Description	Reset
[23:0]	WO	hbf_para_tap_coef_22	hbf_para_tap_coef_22,pdm input hbf filter coefficient	0xfce06b

[hbf_para_tap_coef_23](#) is hbf filter coefficient.

Offset Address: 0x4A4, Total Reset Value: 0x0004_0a20

bits	Access	Name	Description	Reset
[31:24]	RO	RSV	Reserved bit, return 0s when read.	0x00
[23:0]	WO	hbf_para_tap_coef_23	hbf_para_tap_coef_23,pdm input hbf filter coefficient	0x040a20

[hbf_para_tap_coef_24](#) is hbf filter coefficient.

Offset Address: 0x4A8, Total Reset Value: 0x00fa_9112

bits	Access	Name	Description	Reset
[31:24]	RO	RSV	Reserved bit, return 0s when read.	0x00
[23:0]	WO	hbf_para_tap_coef_24	hbf_para_tap_coef_24,pdm input hbf filter coefficient	0xfa9112

[hbf_para_tap_coef_25](#) is hbf filter coefficient.

Offset Address: 0x4AC, Total Reset Value: 0x0007_de27

bits	Access	Name	Description	Reset
[31:24]	RO	RSV	Reserved bit, return 0s when read.	0x00
[23:0]	WO	hbf_para_tap_coef_25	hbf_para_tap_coef_25,pdm input hbf filter coefficient	0x07de27

[hbf_para_tap_coef_26](#) is hbf filter coefficient.

Offset Address: 0x4B0, Total Reset Value: 0x00f2_96cf

bits	Access	Name	Description	Reset
[31:24]	RO	RSV	Reserved bit, return 0s when read.	0x00
[23:0]	WO	hbf_para_tap_coef_26	hbf_para_tap_coef_26,pdm input hbf filter coefficient	0xf296cf

[hbf_para_tap_coef_27](#) is hbf filter coefficient.

Offset Address: 0x4B4, Total Reset Value: 0x0028_afcd

bits	Access	Name	Description	Reset
[31:24]	RO	RSV	Reserved bit, return 0s when read.	0x00
[23:0]	WO	hbf_para_tap_coef_27	hbf_para_tap_coef_27,pdm input hbf filter coefficient	0x28afcd

[hbf_para_tap_coef_28](#) is hbf filter coefficient.

Offset Address: 0x4B8, Total Reset Value: 0x0040_0000

bits	Access	Name	Description	Reset
[31:24]	RO	RSV	Reserved bit, return 0s when read.	0x00
[23:0]	WO	hbf_para_tap_coef_28	hbf_para_tap_coef_28,pdm input hbf filter coefficient	0x400000

[pdm_channel_enable](#) is to control pdm channel number and pdm data valid clock edge.

Offset Address: 0x4BC, Total Reset Value: 0x0000_0000

bits	Access	Name	Description	Reset
[31:6]	RO	RSV	Reserved bit, return 0s when read.	0x0000000
[5:4]	RW	pdm_pn_sel	2'b10: posedge 2'b11: negedge others: double edge	0x0
[3:0]	RW	pdm_chl_num	pdm input channel num, the serial numbers of enabled channels are from small to big.	0x0

pdm2pcm_out_data is pdm out data valid data width 28bits Offset Address: 0x4C0, Total Reset Value: 0x0000_0000

bits	Access	Name	Description	Reset
[31:0]	RO	pdm2pcm_out_data	pdm out data valid data width 28bits: [27:24] is channel number, [23:0] is pcm data	0x0000_0000

pdm_interrupt is pdm error interrupt. Offset Address: 0x4C4, Total Reset Value: 0x0000_0000

bits	Access	Name	Description	Reset
[31:1]	RO	RSV	Reserved bit, return 0s when read.	0x0000_0000
[0]	W0C	pdm_intr	pdm interrupt register, means pdm input fifo is overflow, write 0 to clear. (no interrupt status register)	0x0

pdmin_clk_spike is pdm clock debounce configure register. Offset Address: 0x640, Total Reset Value: 0x_0x0000_0008

bits	Access	Name	Description	Reset
[31:4]	RO	RSV	Reserved bit, return 0s when read.	0x00000000
[3:2]	RW	pdmin_clk_spike_th	debounce window number while using pclk to debounce audio_dev_clk: 0:2 windows 1:3 windows 2:4 windows 3:5 windows	0x2
[1:0]	RW	pdmin_clk_spike_cnt	debounce window granularity of pclk cycle while using pclk to debounce audio_dev_clk: 0:1 pclk cycle 1:2 pclk cycle 2:3 pclk cycle 3:4 pclk cycle	0x0

pdmin_ch_spike is pdm input data debounce configure register. Offset Address: 0x644, Total Reset Value: 0x0000_cccc

bits	Access	Name	Description	Reset
[31:16]	RO	RSV	Reserved bit, return 0s when read.	0x0000
[15:14]	RW	pdmin_ch3_spike_th	debounce window number while using pclk to debounce pdm_in3: 0:2 windows 1:3 windows 2:4 windows 3:5 windows	0x3
[13:12]	RW	pdmin_ch3_spike_cnt	debounce window granularity of pclk cycle while using pclk to debounce pdm_in3: 0:1 pclk cycle 1:2 pclk cycle 2:3 pclk cycle 3:4 pclk cycle	0x0
[11:10]	RW	pdmin_ch2_spike_th	debounce window number while using pclk to debounce pdm_in2: 0:2 windows 1:3 windows 2:4 windows 3:5 windows	0x3
[9:8]	RW	pdmin_ch2_spike_cnt	debounce window granularity of pclk cycle while using pclk to debounce pdm_in2: 0:1 pclk cycle 1:2 pclk cycle 2:3 pclk cycle 3:4 pclk cycle	0x0
[7:6]	RW	pdmin_ch1_spike_th	debounce window number while using pclk to debounce pdm_in1: 0:2 windows 1:3 windows 2:4 windows 3:5 windows	0x3
[5:4]	RW	pdmin_ch1_spike_cnt	debounce window granularity of pclk cycle while using pclk to debounce pdm_in1: 0:1 pclk cycle 1:2 pclk cycle 2:3 pclk cycle 3:4 pclk cycle	0x0

bits	Access	Name	Description	Reset
[3:2]	RW	pdmin_ch0_spike_th	debounce window number while using pclk to debounce pdm_in0: 0:2 windows 1:3 windows 2:4 windows 3:5 windows	0x3
[1:0]	RW	pdmin_ch0_spike_cnt	debounce window granularity of pclk cycle while using pclk to debounce pdm_in0: 0:1 pclk cycle 1:2 pclk cycle 2:3 pclk cycle 3:4 pclk cycle	0x0

[pdm2pcm_ftr_bypass](#) is pdm filters bypass control signal.

Offset Address: 0x448, Total Reset Value: 0x0000_0000

bits	Access	Name	Description	Reset
[31:3]	RO	RSV	Reserved bit, return 0s when read.	0x00000000
[2]	RW	cic_dsmp1_fir_bypass	bypass input cic filter	0x0
[1]	RW	cic_mp_dsmp1_fir_bypass	bypass input cic compensate filter	0x0
[0]	RW	hbf_dsmp1_fir_bypass	bypass input HBF fileter	0x0

11.1.6 Interface Description

The following table shows the audio interface description.

Table 11-6 Audio Interface Description

Module Pin	Direction	Pad Name	IOMUX Setting	Audio Setting
audio_in0	I/O	PAD_IO_27	io_27_sel = 3'h3	NA
audio_in0	I/O	PAD_IO_37	io_37_sel = 3'h2	audio_in_mode != 2'h2
audio_in1	I/O	PAD_IO_28	io_28_sel = 3'h3	NA
audio_in1	I/O	PAD_IO_35	io_35_sel = 3'h2	audio_in_mode != 2'h2
audio_in2	I/O	PAD_IO_36	io_36_sel = 3'h2	NA
audio_in3	I/O	PAD_IO_34	io_34_sel = 3'h2	NA
audio_out0	I/O	PAD_IO_35	io_35_sel = 3'h2	audio_in_mode = 2'h2
audio_out1	I/O	PAD_IO_37	io_37_sel = 3'h2	audio_in_mode = 2'h2
audio_clk	I/O	PAD_IO_32	io_32_sel = 3'h2	NA
i2s_ws_out	I/O	PAD_IO_33	io_33_sel = 3'h2	NA
pdm_clk	I/O	PAD_IO_26	io_26_sel = 3'h3	NA

11.1.7 Application Notes

11.1.7.1 PDM Receiver

1. Set registers according to senario in Application Description;
2. Set pdm_out_endian of [PDM_CONTROL_REG_0](#), set oversamle rate in [PDM_CONTROL_REG_1](#) and [hbf_dsmp1_fir_bypass](#) according to working mode.
3. If work in DMA mode, pdm_fifo_th should be configured according to configuration in peripheral DMA.
4. If coefficient of filters in pdm2pcm_filter need to be changed, [mp_para_tap_coef_0--mp_para_tap_coef_15](#) and [hbf_para_tap_coef_0--hbf_para_tap_coef_28](#) should be written with dedicated value.

5. If a fine adjustment of capture clock edge related to PDM input data need to be taken, value of `pdmn_clk_spike_th` in `pdmn_clk_spike` and `pdmn_chx_spike_th`(x is 0-7 channel number) in `pdmn_ch_spike`.
6. Clear `pdmn_fifo` by asserting `pdm_fifo_clr`(bit4 of `PDM_CONTROL_REG_0`), then deassert `pdm_fifo_clr`.
7. Enable PDM receiver by setting bit2-bit0 of `PDM_CONTROL_REG_0` to 0x7.

11.1.7.2 I2S Receiver

1. Enable the DW_apb_i2s by setting bit 0 of the DW_apb_i2s Enable Register (IER) to 1.
2. Enable the I2S Receiver block by writing a 1 in bit 0 (RXEN) of the I2S Receiver Block Enable Register (IRER).
3. Enable the I2S Clock Generation block by writing a 1 in bit 0 (CLKEN) of the Clock Enable Register (CER).
4. Read bit 0 (RXDA) of the Interrupt Status Register (ISR). When bit 0 of the goes high, the default trigger has been reached.
5. Read the contents of the Left Receive Buffer Register (LRBR) and Right Receive Buffer Register (RRBR). The bit is dependent on how you have configured (or programmed) the trigger level. When RX DMA is enabled, the data contents will be read from `RXDMA` instead of the Left Receive Buffer Register (LRBR) and the Right Receive Buffer Register (RRBR), as defined in step 5 of the previous procedure (receiver, master mode).

11.1.7.3 I2S Transmitter

1. Enable the DW_apb_i2s by setting bit 0 of the DW_apb_i2s Enable Register (IER) to 1.
2. If it works in non-DMA mode, fill the TX-FIFO by writing data to the Left Transmit Holding Register (LTHR) and the Right Transmit Holding Register (RTHR), respectively. Keep writing in this order—left and then right— until the FIFO is filled with data. The DW_apb_i2s starts transmitting stereo data on the first left cycle when the ws signal goes low.
3. Enable the I2S Transmitter block by writing a 1 in bit 0 (TXEN) of the 2S Transmitter Block Enable Register (ITER).
4. If it works in DMA mode, fill the TX-FIFO by writing to the Transmitter Block DMA Register (TXDMA).
5. Enable the I2S Clock Generation block by writing a 1 in bit 0 (CLKEN) of the Clock Enable Register (CER).

11.1.7.4 Disable-enable audio

Audio can be disable at any stage of process by `pdm_en`(bit0 of `PDM_CONTROL_REG_0` 0x0400) or by reset. If reenable audio is needed, reset audio(write 0x1 to bit0 of `AUDIO_RST_CTL`, and confirm reset done by observing bit31 of `AUDIO_RST_CTL` to be set) first to clear datapath(if not reset before), after release reset, initialize and enable audio.

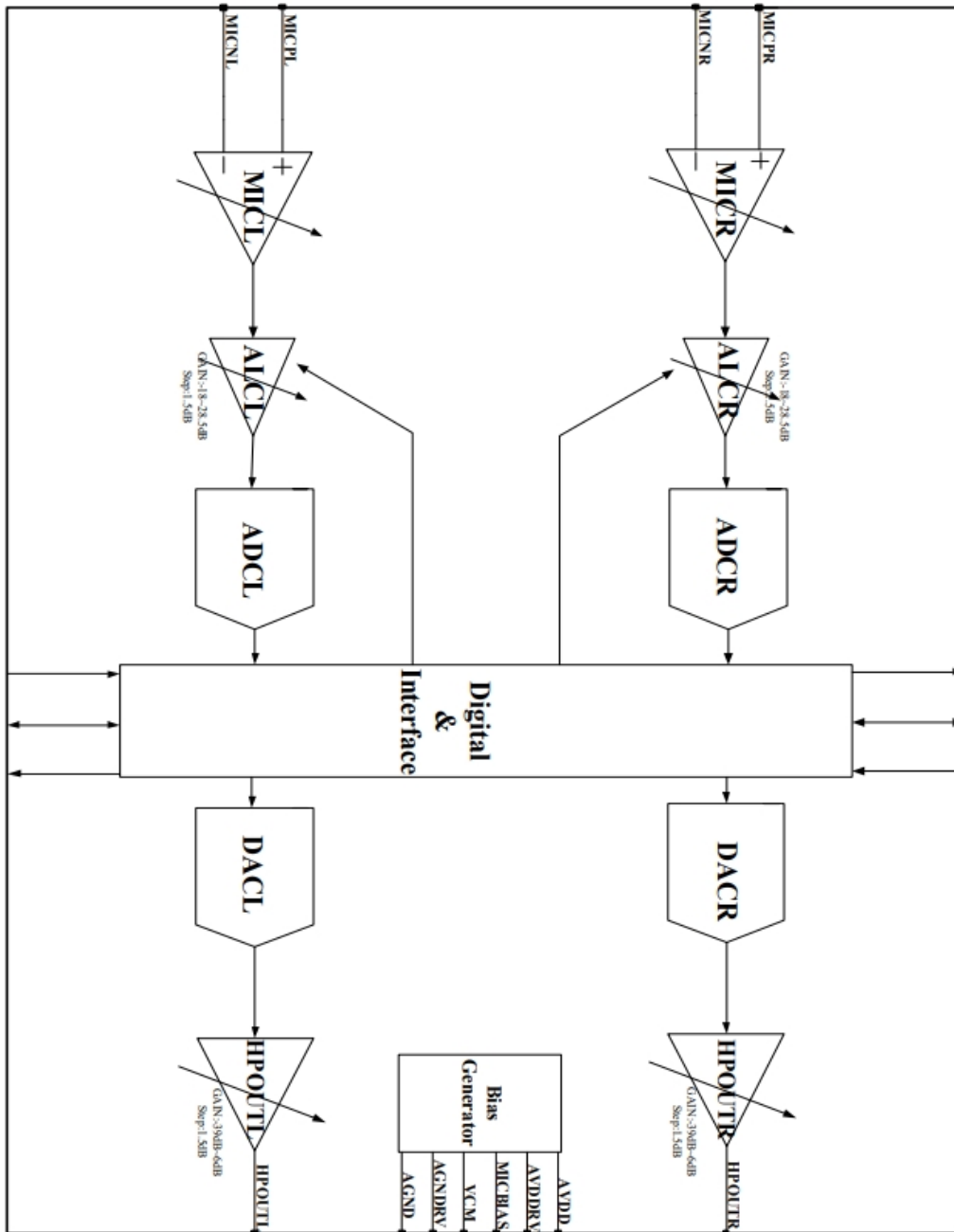
11.2 Audio Codec

11.2.1 Overview

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

11.2.2 Block Diagram

The block diagram of Codec IP is shown in Figure 11-1. The default Codec IP contains 2 DAC channels for stereo playback and 2 ADC channels for stereo recording from microphone. [Figure 11-2-1 CODEC IP Block Diagram](#)



11.2.3 FEATURE

- 24 bits DAC with 94dB SNR
- Support 16 to 32Ω headphone output or line output
- Low power: 4mA for stereo playback
- 24 bits ADC with 92dB SNR
- Support differential and single-ended microphone or line input
- Low power: 4mA for stereo recording
- Automatic Level Control (ALC) for smooth audio recording
- TSMC 12nm FFC process
- Small die size:
 - 0.18mm² for per DAC path
 - 0.19mm² for per ADC path
 - 0.77mm²(810μm×948μm) for entire stereo application
- Low power: less than 0.05mA for standby
- Support Mono, Stereo, 5.1 and 7.1 HiFi channel performance
- Programmable input and output analog gains

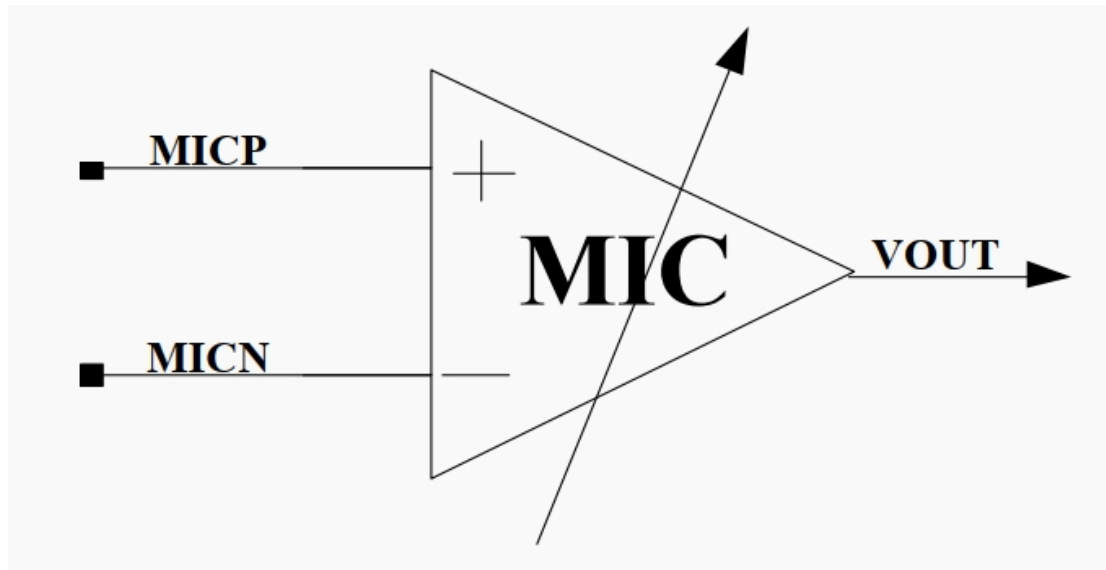
- Digital interpolation and decimation filter integrated
- Sampling rate of 8kHz/12kHz/16kHz/24kHz/32kHz/44.1kHz/48kHz/96kHz/192kHz
- 1.8V supply for analog and 0.8V supply for digital

11.2.4 Function Description

This section describes some function configuration for operation. Meanwhile this section provides some application guide. Please refer to section 8 to obtain the detail information of relevant registers.

11.2.4.1 MICROPHONE INPUT

Figure 11-2-2 MIC Diagram



There are two input channels named left ADC channel and right ADC channel. In each channel, there are two inputs which are configured as differential input by the microphone PGA (MICL and MICR).

In the left channel, microphone inputs are MICPL and MICNL. In the right channel, microphone inputs are MICPR and MICRL.

Microphone PGA has a gain range from 0dB to 20dB.

11.2.4.2 ALC

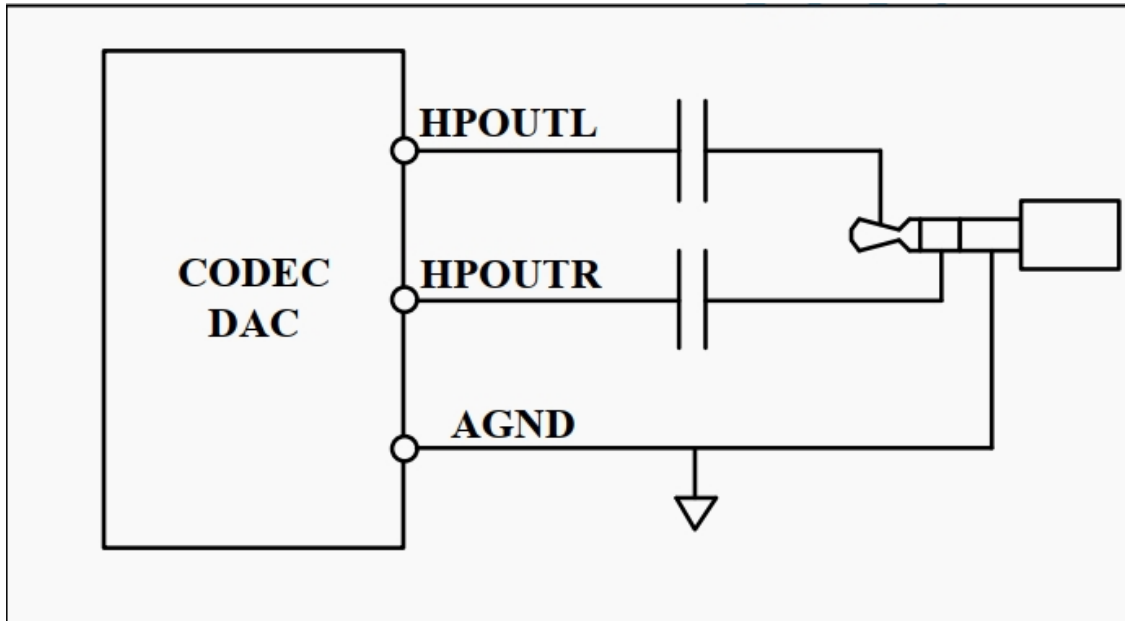
Automatic Level Control (ALC) function is included to adjust the signal level, which is input into ADC. ALC will measure the signal magnitude and compare it to a defined threshold. Then it will adjust the ALC controlled PAG (ALC_L and ALC_R) gain according to the comparison result.

The programmable gain range of ALC controlled PAG is from -18dB to +28.5dB. The tuning step is 1.5dB.

11.2.4.3 DAC OUTPUT

Codec supports headphone output or line output configurations. The output can drive load through DC-blocking capacitor. In the configuration using DC-blocking capacitor, shown in the following figure, the headphone ground is connected to the real ground. The capacitance and the load resistance determine the lower cut-off frequency. For instance, if 600Ω load and 4.7μF DC-blocking capacitor are used, the lower cut-off frequency is: $f = 56.5\text{Hz}$

The DC-blocking capacitor can be increased to lower the cut-off frequency for better bass response. [Figure 11-2-3 Headphone Output Configuration](#)



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

11.2.4.4 MICROPHONE BIAS

The output of the Microphone bias is used for bias external microphones. The bias voltage can vary from $0.8 \cdot AVDD$ to $0.975 \cdot AVDD$ with a step of $0.025 \cdot AVDD$.

11.2.4.5 FREQUENCY AMPLITUDE RESPONSE

With high pass filter and low pass filter which can be independently controlled, the overall filter amplitude response will be much better than PCM4 specification. The in-band ripple is within 0.1dB and output of band rejection is over 93dB. [Figure 11-2-4](#) ADC filter performance (<0.1dB ripple achieved)

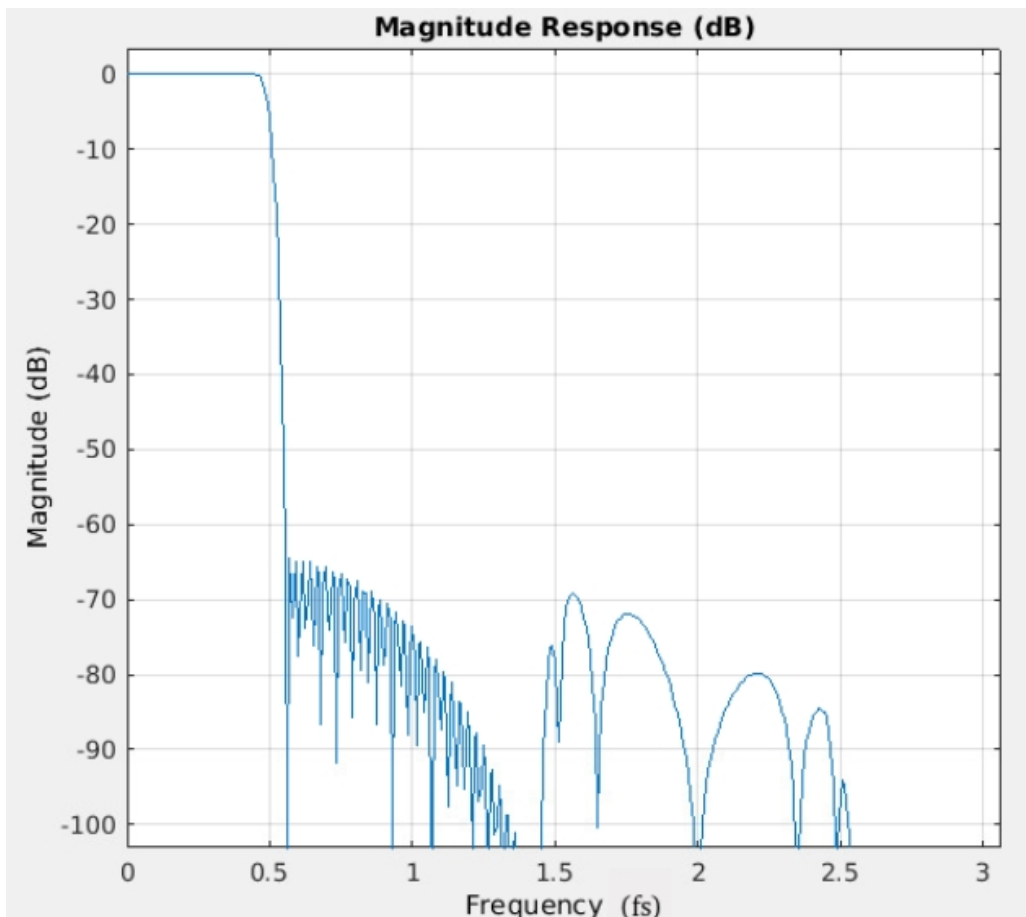
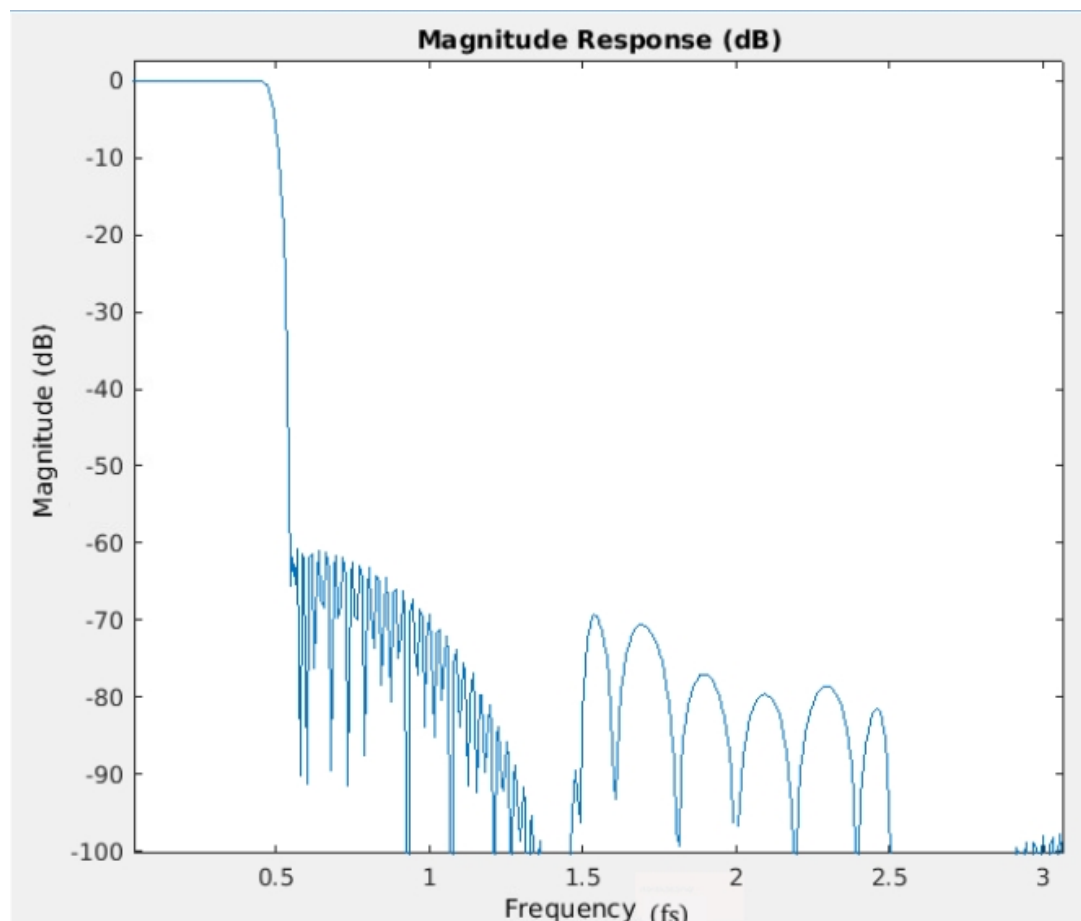
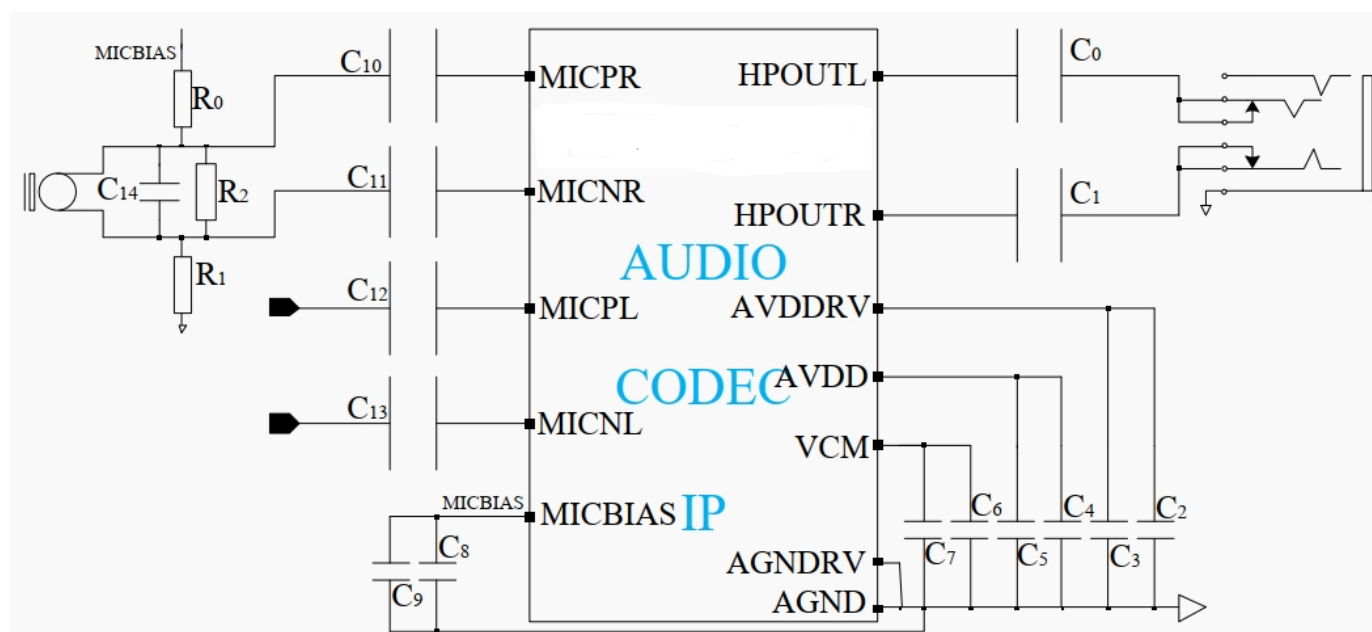


Figure 11-2-5 DAC filter performance (<0.1dB ripple achieved)



11.2.4.6 PCB LAYOUT GUIDE

The typical connection of Innosilicon Codec is shown in the following figure. Customer can modify the connection according to the practical application. [Figure 11-2-6 PCB Layout](#)



HPOUTL and HPOUTR are connected to jack through DC-blocking capacitors.

VCM is recommended to be connected to ground through 4.7uF and 0.1uF decoupling capacitor for better noise performance. The capacitor should be placed as close as possible to VCM pin.

MICPL and MICNL,MICPR and MICNR are connected to microphones or lien input through DC-blocking capacitor.

MICBIAS is used for bias the microphone through a bias resistor.The resistor value should be adjusted according to the microphone type.

AVDD and AVDDRV should be connected to 3.3V supply with 4.7uF and 0.1uF decoupling capacitor.The decoupling capacitor should be placed as close as possible to AVDD and AVDDRV pin.

11.2.5 Register Description

11.2.5.1 Register Summary

Table 11-1 describes CODEC registers.

Table 11-1 Summary of CODEC registers (base address: 9140_E000)

Offset Address	Register	DESCRIPTIONS
0x0000	RST_CFG	RST configuration register
0x0004	DAC_MUTE_CFG	DAC MUTE configuration register
0x0008	ADC_I2S_TX_CFG	ADC I2S TX configuration register
0x000C	I2S_CFG	I2S configuration register
0x0010	DAC_I2S_CFG	DAC I2S configuration register
0x0014	CODEC_0014_CFG	-
0x0018	DAC_VOL_CFG	DAC digital gain selection
0x001C	DAC_BIST_SEL	DAC left and right Channel Bist Mode select
0x0020	ADCL_VOL_CFG	Left ADC Digital Volume Control
0x0024	ADCR_VOL_CFG	Right ADC Digital Volume Control
0x0028	CODEC_0028_CFG	-
0x002C	-	Reserved Register
0x0030	-	Reserved Register

Offset Address	Register	DESCRIPTIONS
0x0034	-	Reserved Register
0x0038	-	Reserved Register
0x003C	-	Reserved Register
0x0040	-	Reserved Register
0x0044	-	Reserved Register
0x0048	-	Reserved Register
0x004C	-	Reserved Register
0x0050	-	Reserved Register
0x0054	-	Reserved Register
0x0058	-	Reserved Register
0x005C	-	Reserved Register
0x0060	-	Reserved Register
0x0064	-	Reserved Register
0x0068	-	Reserved Register
0x006C	-	Reserved Register
0x0070	-	Reserved Register
0x0074	-	Reserved Register
0x0078	-	Reserved Register
0x007C	-	Reserved Register
0x0080	CODEC_0080_CFG	-

Offset Address	Register	DESCRIPTIONS
0x0084	SEL_VREF	The control signal to select current to pre-charge/discharge.
0x0088	SEL_IBIAS	The signal to adjust ADC and DAC bias current I _{bias} .
0x008C	CODEC_008C_CFG	-
0x0090	CODEC_0090_CFG	-
0x0094	CODEC_0094_CFG	-
0x0098	CODEC_0098_CFG	-
0x009C	CODEC_009C_CFG	-
0x0100	CODEC_0100_CFG	-
0x0104	CODEC_0104_CFG	-
0x0108	HPOUTL_CFG	HPOUTL module configuration register.
0x010C	GAIN_HPOUTL_CFG	Select gain of HPOUTL module configuration register.
0x0110	CODEC_0110_CFG	-
0x0114	HPOUTR_CFG	HPOUTR module configuration register.
0x0118	GAIN_HPOUTR_CFG	Select gain of HPOUTR module configuration register.
NOTE: From the 0x011C to 0x0154 is the description of the agc register for the left channel. The right channel has the same registers but different address from 0x0158 to 0x0190.		
0x011C	CODEC_011C_CFG	-
0x0120	DecayAttack_Time_CFG	Decay and Attack (gain ramp-up) time configuration register.
0x0124	CODEC_0124_CFG	-

Offset Address	Register	DESCRIPTIONS
0x0128	Left_PGA_CFG	Left channel input PGA configuration register.
0x012C	CODEC_012C_CFG	-
0x0130	CODEC_0130	The low 8 bits of the AGC maximum level.
0x0134	CODEC_0134	The high 8 bits of the AGC maximum level.
0x0138	CODEC_0138	The low 8 bits of the AGC minimum level.
0x013C	CODEC_013C	The high 8 bits of the AGC minimum level.
0x0140	CODEC_0140	-
0x0144	CODEC_0144	Reserved Registers.
0x0148	CODEC_0148	Peak detect value decrease rate.
0x014C	CODEC_014C	Timeout counter for slow clock when zero cross not occur.
0x0150	CODEC_0150	Time delay for noise judge when ALC gain changed.
0x0154	CODEC_0154	AGC gain probe signal.

11.2.5.2 Detail Register Description

RST_CFG

RST_CFG is RESET configuration register.

Offset Address: 0x0000 Total Reset Value: 0x0000_0003

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	-

Bits	Access	Name	Description	Reset
[7]	RW	BIST_RSTN	Bist module reset. 0: Reset 1: Work	0x0
[6:2]	-	reserved	Reserved	-
[1]	RW	DIGCORE_RSTN	Codec digital core reset. This reset only reset the codec data path. 0: Reset 1: Work	0x1
[0]	RW	SYS_RSTN	Codec system reset. This signal will reset the registers which control all the digital and analog part. 0: Reset 1: Work	0x1

DAC_MUTE_CFG

DAC_MUTE_CFG is DAC MUTE configuration register.

Offset Address: 0x0004 Total Reset Value: 0x0000_0050

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	-
[7]	RW	DAC_MUTE_EN	DAC digital volume change flow control. 0: update immediately 1: update step by step to reduce POP	0x0
[6:4]	RW	DAC_MUTE_SR [2:0]	The DAC sample rate indication to DAC digital volume change step selection when DAC_MUTE_EN is set to 1. 000: 8K 001: 16K 010: 24K 011: 32K 100: 44.1K 101: 48K 110: 88.2K 111: 96K	0x5
[3:0]	-	reserved	Reserved	-

ADC_I2S_TX_CFG

ADC_I2S_TX_CFG is ADC I2S TX configuration register.

Offset Address: 0x0008 Total Reset Value: 0x0000_0050

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	-

Bits	Access	Name	Description	Reset
[7]	RW	I2S_TX_LRP	ADC I2S Configuration Register[7]. LRC Polarity 0: Normal 1: Reversal	0x0
[6:5]	RW	I2S_TX_WL [1:0]	ADC I2S Configuration Register[6:5]. Valid Word Length in one 1/2Frame 11: 32 bits 10: 24 bits 01: 20 bits 00: 16 bits	0x2
[4:3]	RW	I2S_TX_FMT [1:0]	ADC I2S Configuration Register[4:3]. Mode 11: PCM Mode 10: I2S Mode 01: Left Justified Mode 00: Right Justified Mode Note: Same word length in 1/2frame and valid data is not supported in Right Justified Mode. For example, 32/24 or 24/20 is supported, but 32/32 or 24/24 is not supported. (1/2frame length/valid data length)	0x2
[2]	-	reserved	Reserved	-
[1:0]	RW	I2S_TX_DATSEL [1:0]	I2S adc data select 00: left data=left ADC; right data =right ADC 01: left data =left ADC; right data = left ADC 10: left data = right ADC; right data =right ADC 11: left data = right ADC; right data = left ADC	0x0

I2S_CFG

I2S_CFG is I2S configuration register.

Offset Address: 0x000C Total Reset Value: 0x0000_000E

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	-
[7]	RW	I2S_RX_PIN_MST	DAC I2S Mode Select for IO pin. 1: Master mode 0: Slave mode	0x0
[6]	RW	I2S_RX_FUN_MST	DAC I2S Mode Select for inner module. 1: Master mode 0: Slave mode	0x0
[5]	RW	I2S_TX_PIN_MST	ADC I2S Mode Select for IO pin. 1: Master mode 0: Slave mode	0x0

Bits	Access	Name	Description	Reset
[4]	RW	I2S_TX_FUN_MST	ADC I2S Mode Select for inner module. 1: Master mode 0: Slave mode	0x0
[3:2]	RW	I2S_TX_LEN [1:0]	ADC I2S Configuration Register[11:10]. 1/2Frame Word Length 11: 32 bits 10: 24 bits 01: 20 bits 00: 16 bits	0x3
[1]	RW	I2S_TX_RSTN	ADC I2S Configuration Register[9]. 0: Reset 1: Work	0x1
[0]	RW	I2S_TX_BCLKINV	ADC I2S Configuration Register[8]. Bit Clock Polarity 0: Normal 1: Reversal	0x0

DAC_I2S_CFG

DAC_I2S_CFG is DAC I2S configuration register.

Offset Address: 0x0010 Total Reset Value: 0x0000_0050

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	-
[7]	RW	I2S_RX_LRP	DAC I2S Configuration Register[7]. LRC Polarity 0: Normal 1: Reversal	0x0
[6:5]	RW	I2S_RX_WL	DAC I2S Configuration Register[6:5]. Valid Word Length in one 1/2Frame 11: 32 bits 10: 24 bits 01: 20 bits 00: 16 bits	0x2
[4:3]	RW	I2S_RX_FMT	DAC I2S Configuration mode 11: PCM Mode 10: I2S Mode 01: Left Justified Mode 00: Right Justified Mode Note-1: Same word length in 1/2frame and valid data is not supported in Right Justified Mode. For example, 32/24 or 24/20 is supported, but 32/32 or 24/24 is not supported. (1/2frame length/valid data length)	0x2

Bits	Access	Name	Description	Reset
[2]	RW	I2S_LR_SWAP	DAC Left-Right SWAP 0: Normal 1: Swap	0x0
[1:0]	-	reserved	Reserved	-

CODEC_0014_CFG

Offset Address: 0x0014 Total Reset Value: 0x0000_0002

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	-
[7:6]	-	reserved	Reserved	-
[5:4]	RW	DAC_DEM [1:0]	DAC de-emphasis filter selection. 00: No de-emphasis filter work 01: 32K sample de-emphasis filter 10: 44.1K sample de-emphasis filter 11: 48K sample de-emphasis filter	0x0
[3:2]	-	reserved	Reserved	-
[1]	RW	I2S_RX_RSTN	DAC I2S Configuration Register[9]. 0: Reset 1: Work	0x1
[0]	RW	I2S_RX_BCLKINV	DAC I2S Configuration Register[8]. Bit Clock polarity 0: Normal 1: Reversal	0x0

DAC_VOL_CFG

DAC_VOL_CFG is DAC digital gain selection configuration register.

Offset Address: 0x0018 Total Reset Value: 0x0000_00F1

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	-
[7:0]	RW	DAC_VOL [7:0]	DAC digital gain selection, every step 0.5dB. 11110001: 0dB 11111111: 7dB 11111110: 6.5dB ... 00000001: -120dB 00000000: Digital mute	0xf1

DAC_BIST_SEL

DAC_BIST_SEL is DAC left and right Channel Bist Mode selection configuration register.

Offset Address: 0x001C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	-
[7:6]	RW	DACR_BIST_SEL [1:0]	DAC Right Channel Bist Mode select. 01: Sine Wave to the PCM(bist mode enable and need to set 0x00[7] to 1) Other: Normal Right Channel data to the PCM (normal mode)	0x0
[5:4]	RW	DACL_BIST_SEL [1:0]	DAC Left Channel Bist Mode select. 01: Sine Wave to the PCM(bist mode enable and need to set 0x00[7] to 1) Other: Normal Left Channel data to the PCM (normal mode)	0x0
[3:0]	-	reserved	Reserved	-

ADCL_VOL_CFG

ADCL_VOL_CFG is Left ADC Digital Volume Control configuration register.

Offset Address: 0x0020 Total Reset Value: 0x0000_00C3

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	-
[7:0]	RW	ADCL_VOL [7:0]	Left ADC Digital Volume Control. 00000000 = Digital Mute 00000001 = -97dB 00000010 = -96.5dB ... 0.5dB steps up to 11111111 = +30dB	0xc3

ADCR_VOL_CFG

ADCR_VOL_CFG is Right ADC Digital Volume Control configuration register.

Offset Address: 0x0024 Total Reset Value: 0x0000_00C3

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	-
[7:0]	RW	ADCR_VOL [7:0]	Right ADC Digital Volume Control. 00000000 = Digital Mute 00000001 = -97dB 00000010 = -96.5dB ... 0.5dB steps up to 11111111 = +30dB	0xc3

CODEC_0028_CFG

Offset Address: 0x0028 Total Reset Value: 0x0000_0004

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	-

Bits	Access	Name	Description	Reset
[7:6]	-	reserved	Reserved	-
[5]	RW	ALCR_EN	1'b0: choose the register0x27[4:0] to control the gain of the PGA 1'b1: choose the ALCL module to control the gain of the PGA	0x0
[4]	RW	ALCL_EN	1'b0: choose the register0x28[4:0] to control the gain of the PGA 1'b1: choose the ALCR module to control the gain of the PGA	0x0
[3]	-	reserved	Reserved	-
[2]	RW	ADC_HPF_DIS	ADC HPF disable control 1: Disable HPF 0: Enable HPF	0x1
[1]	RW	ADCR_POL	ADC right channel data polarity invert. 1: Right channel data polarity invert 0: Right channel data polarity not inverted	0x0
[0]	RW	ADCL_POL	ADC left channel data polarity invert. 1: left channel data polarity invert 0: left channel data polarity not inverted	0x0

Note: 0x002C~0x007C is Reserved Register.

CODEC_0080_CFG

Offset Address: 0x0080 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	-
[7]	-	reserved	Reserved	-
[6]	RW	EN_VREF	The enable signal of reference voltage module 0: Disable 1: Enable	0x0
[5]	RW	EN_IBIAS_ADC	The enable signal of current source for ADC. 0: Disable 1: Enable	0x0
[4]	RW	EN_IBIAS_DAC	The enable signal of current source for DAC. 0: Disable 1: Enable	0x0
[3]	RW	EN_MICBIAS	The enable signal of MIC bias voltage (MICBIAS) buffer 0: Stop Working 1: Work	0x0
[2:0]	RW	GAIN_MICBIAS	The level range control signal of MIC bias voltage (MICBIAS) 000: 1.6*VREF 111: 1.95*VREF Step: 0.05*VREF VREF = AVDD/2	0x0

SEL_VREF

SEL_VREF is the control register to select current to pre-charge/discharge.

Offset Address: 0x0084 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	-
[7]	RW	SEL_VREF[7]	0: Choose the Current 128*I0 1: Don't choose the current 128*I0	0x0
[6]	RW	SEL_VREF[6]	0: Choose the Current 64*I0 1: Don't choose the current 64*I0	0x0
[5]	RW	SEL_VREF[5]	0: Choose the Current 32*I0 1: Don't choose the current 32*I0	0x0
[4]	RW	SEL_VREF[4]	0: Choose the Current 16*I0 1: Don't choose the current 16*I0	0x0
[3]	RW	SEL_VREF[3]	0: Choose the Current 8*I0 1: Don't choose the current 8*I0	0x0
[2]	RW	SEL_VREF[2]	0: Choose the Current 4*I0 1: Don't choose the current 4*I0	0x0
[1]	RW	SEL_VREF[1]	0: Choose the Current 2*I0 1: Don't choose the current 2*I0	0x0
[0]	RW	SEL_VREF[0]	0: Choose the Current I0 1: Don't choose the current I0	0x0

SEL_IBIAS

SEL_IBIAS is the control register to adjust ADC and DAC bias current Ibias.

Offset Address: 0x0088 Total Reset Value: 0x0000_0077

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	-
[7:4]	RW	SEL_IBIAS_ADC	The signal to adjust ADC bias current I _{bias_ADC} = I0* (BIT[3]*8+BIT[2]*4+BIT[1]*2+BIT[0]+1)	0x7
[3:0]	RW	SEL_IBIAS_DAC	The signal to adjust DAC bias current I _{bias_DAC} = I0* (BIT[3]*8+BIT[2]*4+BIT[1]*2+BIT[0]+1)	0x7

CODEC_008C_CFG

Offset Address: 0x008C Total Reset Value: 0x0000_0077

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	-
[7]	RW	EN_BUF_ADCL	The enable signal of reference voltage buffer of ADCL. 0: Disable 1: Enable	0x0

Bits	Access	Name	Description	Reset
[6]	RW	EN_MICL	The enable signal of MIC module in the ADCL. 0: Disable 1: Enable	0x0
[5]	RW	EN_ALCL	The enable signal of ALCL module 0: Stop Working 1: Work	0x0
[4]	RW	EN_CLK_ADCL	The enable signal of CLOCK for ADCL module 0: Set to logic "1" 1: Work	0x0
[3]	RW	EN_ADCL	The enable signal of ADC module in the ADCL. 0: Disable 1: Enable	0x0
[2]	RW	INITIAL_ADCL	The initial signal of ADC module in the ADCL. 0: Initialization 1: Work	0x0
[1]	RW	INITIAL_ALCL	The initial signal of ALC module in the ADCL. 0: Initialization 1: Work	0x0
[0]	RW	INITIAL_MICL	The initial signal of MIC module in the ADCL. 0: Initialization 1: Work	0x0

CODEC_0090_CFG

Offset Address: 0x0090 Total Reset Value: 0x0000_000C

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	-
[7]	RW	MUTE_MICL	The mute signal of MIC module in the ADCL. 0: Mute 1: Work	0x0
[6:5]	RW	GAIN_MICL [1:0]	The gain signal of MIC module in the ADCL. Gain: 00: 0dB 01: 6dB 10: 20dB 11: 30dB	0x0
[4:0]	RW	GAIN_ALCL	The gain control of ALCL module 00000: -18dB 01100: 0dB 11111: 28.5dB Step: 1.5dB	0x0c

CODEC_0094_CFG

Offset Address: 0x0094 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	-
[7:2]	-	reserved	Reserved	-
[1]	RW	EN_ZERODET_ADCL	The enable signal of the ADCL input zero-crossing detection module: 0: Stop Working 1: Working	0x0
[0]	RW	-	Gain_alcl_det0_bp	0x0

CODEC_0098_CFG

Offset Address: 0x0098 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	-
[7]	RW	EN_BUF_ADCR	The enable signal of reference voltage buffer of ADCR. 0: Disable 1: Enable	0x0
[6]	RW	EN_MICR	The enable signal of MIC module in the ADCR. 0: Disable 1: Enable	0x0
[5]	RW	EN_ALCR	The enable signal of ALCR module 0: Stop Working 1: Work	0x0
[4]	RW	EN_CLK_ADCR	The enable signal of CLOCK for ADCR module 0: Set to logic "1" 1: Work	0x0
[3]	RW	EN_ADCR	The enable signal of ADC module in the ADCR. 0: Disable 1: Enable	0x0
[2]	RW	INITIAL_ADCR	The initial signal of ADC module in the ADCR. 0: Initialization 1: Work	0x0
[1]	RW	INITIAL_ALCR	The initial signal of ALC module in the ADCR. 0: Initialization 1: Work	0x0
[0]	RW	INITIAL_MICR	The initial signal of MIC module in the ADCR. 0: Initialization 1: Work	0x0

CODEC_009C_CFG

Offset Address: 0x009C Total Reset Value: 0x0000_000C

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	-
[7]	RW	MUTE_MICR	The mute signal of MIC module in the ADCR. 0: Mute 1: Work	0x0
[6:5]	RW	GAIN_MICR [1:0]	The gain signal of MIC module in the ADCR. Gain: 00: 0dB 01: 6dB 10: 20dB 11: 30dB	0x0
[4:0]	RW	GAIN_ALCR	The gain control of ALCR module 00000: -18dB 01100: 0dB 11111: 28.5dB Step: 1.5dB	0x0c

CODEC_0100_CFG

Offset Address: 0x0100 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	-
[7:2]	-	reserved	Reserved	-
[1]	RW	EN_ZERODET_ADCR	The enable signal of the ADCR input zero-crossing detection module: 0: Stop Working 1: Working	0x0
[0]	RW	-	Gain_alcr_det0_bp	0x0

CODEC_0104_CFG

Offset Address: 0x0104 Total Reset Value: 0x0000_0010

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	-
[7]	RW	MUTE_HPOUTL	The mute signal of HPOUTL module 0: Mute 1: Work	0x0
[6]	RW	EN_BUF_DACL	The enable signal of reference voltage buffer in the left DAC channel. 0: Disable 1: Enable	0x0
[5:4]	RW	POP_CTRL_DACL [1:0]	The control signal of POP sound in the left DAC channel. 01: Initialization 10: Work	0x1

Bits	Access	Name	Description	Reset
[3]	RW	EN_VREF_DACL	The enable signal of reference voltage of DACL module. 0: Disable 1: Enable	0x0
[2]	RW	EN_CLK_DACL	The enable signal of CLOCK module for DACL 0: Set CLOCK to logic "1" 1: Work	0x0
[1]	RW	EN_DACL	The enable signal of DACL module 0: Stop working 1: Work	0x0
[0]	RW	INITIAL_DACL	The Initial signal of DACL module 0: Initialization 1: Work	0x0

HPOUTL_CFG

HPOUTL_CFG is the HPOUTL module configuration register.

Offset Address: 0x0108 Total Reset Value: 0x0000_0008

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	-
[7:6]	-	reserved	Reserved	-
[5]	RW	EN_HPOUTL	The enable signal of HPOUTL module 0: Initialization 1: Work	0x0
[4]	RW	INITIAL_HPOUTL	The initial signal of HPOUTL module 0: Disable 1: Enable	0x0
[3:0]	RW	SEL_HPOUTL	The signal to adjust HPOUTL driver strength 0001 100% 0010 60.1% 0011 44.2% 0100 35.5% 0101 29.7% 0110 25.7% 0111 22.7% 1000 20.4%	0x8

GAIN_HPOUTL_CFG

GAIN_HPOUTL_CFG is the register selecting gain of HPOUTL module.

Offset Address: 0x010C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	-

Bits	Access	Name	Description	Reset
[7:5]	-	reserved	Reserved	-
[4:0]	RW	GAIN_HPOUTL	The signal to select gain of HPOUTL module: 00000: -39dB 11010: 6dB 11111: 0dB Step: 1.5dB	0x00

CODEC_0110_CFG

Offset Address: 0x0110 Total Reset Value: 0x0000_0010

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	-
[7]	RW	MUTE_HPOUTR	The mute signal of HPOUTR module 0: Mute 1: Work	0x0
[6]	RW	EN_BUF_DACR	The enable signal of reference voltage buffer in the right DAC channel. 0: Disable 1: Enable	0x0
[5:4]	RW	POP_CTRL_DACR [1:0]	The control signal of POP sound in the right DAC channel. 01: Initialization 10: Work	0x1
[3]	RW	EN_VREF_DACR	The enable signal of reference voltage of DACR module. 0: Disable 1: Enable	0x0
[2]	RW	EN_CLK_DACR	The enable signal of CLOCK module for DACR 0: Set CLOCK to logic "1" 1: Work	0x0
[1]	RW	EN_DACR	The enable signal of DACR module 0: Stop working 1: Work	0x0
[0]	RW	INITIAL_DACR	The Initial signal of DACR module 0: Initialization 1: Work	0x0

HPOUTR_CFG

HPOUTR_CFG is the HPOUTR module configuration register.

Offset Address: 0x0114 Total Reset Value: 0x0000_0008

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	-
[7:6]	-	reserved	Reserved	-

Bits	Access	Name	Description	Reset
[5]	RW	EN_HPOUTR	The enable signal of HPOUTR module 0: Initialization 1: Work	0x0
[4]	RW	INITIAL_HPOUTR	The initial signal of HPOUTR module 0: Disable 1: Enable	0x0
[3:0]	RW	SEL_HPOUTR	The signal to adjust HPOUTR driver strength 0001 100% 0010 60.1% 0011 44.2% 0100 35.5% 0101 29.7% 0110 25.7% 0111 22.7% 1000 20.4%	0x8

GAIN_HPOUTR_CFG

GAIN_HPOUTR_CFG is the register Selecting gain of HPOUTR module.

Offset Address: 0x0118 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	-
[7:5]	-	reserved	Reserved	-
[4:0]	RW	GAIN_HPOUTR	The signal to select gain of HPOUTR module: 00000: -39dB 11010: 6dB 11111: 0dB Step: 1.5dB	0x00

NOTE: From the 0x011C to 0x0154 is the description of the agc register for the left channel. The right channel has the same registers but different address from 0x0158 to 0x0190.

CODEC_011C_CFG

Offset Address: 0x011C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	-
[7]	-	reserved	Reserved	-
[6]	RW	-	Choose the method to control the gain attack 0 : Normal way 1 : Jack way	0x0

Bits	Access	Name	Description	Reset
[5:4]	RW	-	There are four methods to generate the control signals 00: Normal way 01: Jack way 1 10: Jack way 2 11: Jack way 3 This register is used to choose the method to generate the control signals according to the actual situation.	0x0
[3:0]	RW	-	AGC hold time before gain is increased in normal mode. 0000: 0ms 0001: 2ms 0010: 4ms 0011: 8ms 0100: 16ms 0101: 32ms 0110: 64ms 0111: 128ms 1000: 256ms 1001: 512ms 1010: 1s 1011~1111 0ms	0x0

Decay_Attack_Time_CFG

Decay_Attack_Time_CFG is the Decay and Attack (gain ramp-up/down) time configuration register.

Offset Address: 0x0120 Total Reset Value: 0x0000_0046

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	-

Bits	Access	Name	Description	Reset
[7:4]	RW	-	In Normal MODE(reg_agc_mode =0), Decay (gain ramp-up) time: 0000: 500us 0001: 1ms 0010: 2ms 0011: 4ms 0100: 8ms 0101: 16ms 0110: 32ms 0111: 64ms 1000: 128ms 1001: 256ms 1010: 512ms 1011~1111: 512ms In Limter MODE(reg_agc_mode =1), Decay (gain ramp-up) time: 0000: 125us 0001: 250us 0010: 500us 0011: 1ms 0100: 2ms 0101: 4ms 0110: 8ms 0111: 16ms 1000: 32ms 1001: 64ms 1010: 128ms 1011~1111: 128ms	0x4

Bits	Access	Name	Description	Reset
[3:0]	RW	-	In Normal MODE(reg_agc_mode =0), Attack (gain ramp-down) time: 0000: 125us 0001: 250us 0010: 500us 0011: 1ms 0100: 2ms 0101: 4ms 0110: 8ms 0111: 16ms 1000: 32ms 1001: 64ms 1010: 128ms 1011~1111: 125us In Limter MODE(reg_agc_mode =1), Decay (gain ramp-up) time: 0000: 32us 0001: 64us 0010: 125us 0011: 250us 0100: 500us 0101: 1ms 0110: 2ms 0111: 4ms 1000: 8ms 1001: 16ms 1010: 32ms 1011~1111: 32us	0x6

CODEC_0124_CFG

Offset Address: 0x0124 Total Reset Value: 0x0000_0041

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	-
[7]	RW	-	Determines the AGC mode of operation: 0 = AGC mode(normal mode) 1 = Limiter mode	0x0
[6]	RW	-	AGC users zero cross enable. 0 = Disabled 1 = Enabled, the AGC gain will update at zero cross enable	0x1
[5]	RW	-	When in the limiter mode, the low amplitude signal will recovery in two modes: 0 = The gain will recovery to the value of the reg_pga_lvol 1 = The gain will recovery to the gain at the moment when the mode changes from AGC to Limiter.	0x0

Bits	Access	Name	Description	Reset
[4]	RW	-	When the amplitude of the signal is more than 87.5% of the Full scale, use this signal to control the fast decrement: 0 = Disabled 1 = Enabled	0x0
[3]	RW	-	AGC noise gate function enable: 0 = Disabled 1 = Enabled	0x0
[2:0]	RW	-	AGC noise gate threshold: 000: -39dB 001: -45dB 010: -51dB 011: -57dB 000: -63dB 001: -69dB 010: -75dB 011: -81dB	0x1

Left_PGA_CFG

Left_PGA_CFG is the left channel input PGA configuration register.

Offset Address: 0x0128 Total Reset Value: 0x0000_002C

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	-
[7:6]	-	reserved	Reserved	-
[5]	RW	-	Left channel input PGA zero cross enable: 0 = Update gain when gain register changes. 1 = Update gain on 1st zero cross after gain register write.	0x1
[4:0]	RW	-	Left channel input PGA gain: 0_0000: -18dB 0_0001: -16.5dB 0_1100: 0dB 0_1101: +1.5dB 0_0110: +3dB 0_1111: 28.5dB	0x0c

CODEC_012C_CFG

Offset Address: 0x012C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	-
[7:4]	-	reserved	Reserved	-

Bits	Access	Name	Description	Reset
[3]	RW	-	Slow clock enabled, used for the zero cross timeout.	0x0
[2:0]	RW	-	Approximate sample rate: 000: 96kHz 001: 48kHz 010: 44.1kHz 011: 32kHz 100: 24kHz 101: 16kHz 110: 12kHz 111: 8kHz Note: According to the sample rate to choose the right configuration.	0x0

CODEC_0130

CODEC_0130 is the low 8 bits of the AGC maximum level configuration register.

Offset Address: 0x0130 Total Reset Value: 0x0000_0026

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	-
[7:0]	RW	-	The low 8 bits of the AGC maximum level	0x26

CODEC_0134

CODEC_0134 is the high 8 bits of the AGC maximum level configuration register.

Offset Address: 0x0134 Total Reset Value: 0x0000_0040

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	-
[7:0]	RW	-	The high 8 bits of the AGC maximum level	0x40

CODEC_0138

CODEC_0138 is the low 8 bits of the AGC minimum level configuration register.

Offset Address: 0x0138 Total Reset Value: 0x0000_0036

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	-
[7:0]	RW	-	The low 8 bits of the AGC minimum level	0x36

CODEC_013C

CODEC_013C is the high 8 bits of the AGC minimum level configuration register.

Offset Address: 0x013C Total Reset Value: 0x0000_0020

Bits	Access	Name	Description	Reset

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	-
[7:0]	RW	-	The high 8 bits of the AGC minimum level	0x20

CODEC_0140

Offset Address: 0x0140 Total Reset Value: 0x0000_0038

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	-
[7]	-	reserved	Reserved	-
[6]	RW	-	AGC function select 0: AGC function off 1: AGC function enable	0x0
[5:3]	RW	-	Set maximum gain of PGA	0x7
[2:0]	RW	-	Set minimum gain of PGA	0x0

CODEC_0144

CODEC_0144 is the reserved Registers.

Offset Address: 0x0144 Total Reset Value: -

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	-
[7:0]	-	reserved	Reserved	-

CODEC_0148

CODEC_0148 is the Peak detect value decrease rate Registers.

Offset Address: 0x0148 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	-
[7:5]	-	reserved	Reserved	-
[4:0]	RW	-	Peak detect value decrease rate	0x01

CODEC_014C

CODEC_014C is the Timeout counter for slow clock when zero cross not occur.

Offset Address: 0x014C Total Reset Value: 0x0000_0010

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	-
[7:0]	RW	-	Timeout counter for slow clock when zero cross not occur	0x10

CODEC_0150

CODEC_0150 is Time delay for noise judge when ALC gain changed registers.

Offset Address: 0x0150 Total Reset Value: 0x0000_0002

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	-
[7:5]	-	reserved	Reserved	-
[4:0]	RW	-	Time delay for noise judge when ALC gain changed	0x02

CODEC_0154

CODEC_0154 is the AGC gain probe signal registers.

Offset Address: 0x0154 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	-
[7:5]	-	reserved	Reserved	-
[4:0]	RO	-	AGC gain probe signal. Only read register. It shows the current value of the agc gain value of left channel. When the agc function is disabled, this value is 0x0c.	0x00

11.2.6 Interface Description

The following table shows the Audio Codec interface description.

Table 11-2 Audio Codec interface description

Module Pin	Direction	Pad Name
MICPL	I	ANA_codec_MICINPL
MICNL	I	ANA_codec_MICINNL
MICPR	I	ANA_codec_MICINPR
MICNR	I	ANA_codec_MICINNR
HPOUTL	O	ANA_codec_HPOUTL
HPOUTR	O	ANA_codec_HPOUTR
MICBIAS	O	ANA_codec_MICBIAS
VCM	O	ANA_codec_VCM

11.2.7 Application Notes

This chapter will give some reference register configurations about how to power up/off CODEC, make the codec in the standby state and how to control the DAC, ADC, AGC and Inner-Bist Wave to integrate the CODEC with you environment.

Note: The every configuration step is better to keep more than 10ms.

11.2.7.1 POWER UP

This section will give the suggest configuration steps to power up the CODEC.

step 0: Supply the power of the digital part and reset the audio codec.

step 1: Configure the register [CODEC_0104_CFG](#) bit[5:4] to 2'b01, to setup the output dc voltage of DAC left channel. Configure the register [CODEC_0110_CFG](#) bit[5:4] to 2'b01, to setup the output dc voltage of DAC right channel.

step 2: Configure the register [SEL_VREF](#) bit[7:0] to 8'b0000_0001.

step 3: Supply the power of the analog part.

step 4: Configure the register [CODEC_0110_CFG](#) bit[6] to 1 to setup reference voltage.

step 5: Change the register [SEL_VREF](#) bit[7:0] from the 8'b0000_0001 to 8'b1111_1111 step by step or configure the [SEL_VREF](#) bit[7:0] to 7'b1111_1111 directly. The suggestion slot time of the step is 20ms.

step 6: Wait until the voltage of VCM keeps stable at the AVDD/2.

step 7: Configure the register [SEL_VREF](#) bit[7:0] to the appropriate value(except 8'b0000_0000) for reducing power.

11.2.7.2 POWER OFF

This section will give the suggest configuration steps to power off the CODEC.

step 0: Keep the power on and disable the DAC and ADC path according to the description of section 11.2.7.4 and 11.2.7.6.

step 1: Configure the register [SEL_VREF](#) bit[7:0] to 8'b0000_0001.

step 2: Configure the register [CODEC_0110_CFG](#) bit[6] to 1'b0 to disable reference voltage.

step 3: Change the register [SEL_VREF](#) bit[7:0] from the 8'b0000_0001 to 8'b111_1111 step by step or configure the [SEL_VREF](#) bit[7:0] to 8'b111_1111 directly. The suggestion slot time of the step is 20ms.

step 4: Wait until the voltage of VCM keep stable at AGND.

step 5: Power off the analog power supply.

step 6: Power off the digital power supply.

11.2.7.3 ENABLE ADC

This section will give the suggest configuration steps to enable ADC.

step 0: Power up the CODEC according to the section 11.2.7.1.

step 1: Configure the register [CODEC_0090_CFG](#) bit[7] to 1, to end the mute station of the left ADC channel. Configure the register [CODEC_009C_CFG](#) bit[7] to 1, to end the mute station of the right ADC channel.

step 2: Configure the register [CODEC_0080_CFG](#) bit[5] to 1, to enable the current source of ADC.

step 3: Configure the register [CODEC_0080_CFG](#) bit[3] to 1, to enable signal of MIC bias voltage(MICBIAS) buffer.

step 4: Configure the register [CODEC_0090_CFG](#) bit[7] to 1, to enable the reference voltage buffer in ADC left channel. Configure the register [CODEC_0098_CFG](#) bit[7] to 1, to enable the reference voltage buffer in ADC right channel.

step 5: Configure the register `CODEC_008C_CFG` bit[6] to 1, to enable the MIC module in ADC left channel. Configure the register `CODEC_0098_CFG` bit[6] to 1, to enable the MIC module in ADC right channel.

step 6: Configure the register `CODEC_008C_CFG` bit[5] to 1, to enable the ALC module in ADC left channel. Configure the register `CODEC_0098_CFG` bit[5] to 1, to enable the ALC module in ADC right channel.

step 7: Configure the register `CODEC_008C_CFG` bit[4] to 1, to enable the clock module in ADC left channel. Configure the register `CODEC_0098_CFG` bit[4] to 1, to enable the clock module in ADC right channel.

step 8: Configure the register `CODEC_008C_CFG` bit[3] to 1, to enable the ADC module in ADC left channel. Configure the register `CODEC_0098_CFG` bit[3] to 1, to enable the ADC module in ADC right channel.

step 9: Configure the register `CODEC_008C_CFG` bit[2] to 1, to end the initialization of the ADCL module. Configure the register `CODEC_0098_CFG` bit[2] to 1, to end the initialization of the ADCR module.

step 10: Configure the register `CODEC_008C_CFG` bit[1] to 1, to end the initialization of the left ALC module. Configure the register `CODEC_0098_CFG` bit[1] to 1, to end the initialization of the right ALC module.

step 11: Configure the register `CODEC_008C_CFG` bit[0] to 1, to end the initialization of the left MIC module. Configure the register `CODEC_0098_CFG` bit[0] to 1, to end the initialization of the right MIC module.

step 12: Configure the register `CODEC_0090_CFG` bit[7] to 1, to end the mute station of the ADC left channel. Configure the register `CODEC_009C_CFG` bit[7] to 1, to end the mute station of the ADC right channel.

step 13: Configure the register `CODEC_0090_CFG` bit[6:5], to select the gain of the left MIC module. Configure the register `CODEC_009C_CFG` bit[6:5], to select the gain of the right MIC module.

step 14: Configure the register `CODEC_0094_CFG` bit[4:0], to select the gain of the left ALC module. Configure the register `CODEC_0100_CFG` bit[4:0], to select the gain of the right ALC module.

step 15: Configure the register `CODEC_0094_CFG` bit[1] to 1, to enable the zerocrossing detection function in ADC left channel. Configure the register `CODEC_0100_CFG` bit[1] to 1, to enable the zerocrossing detection function in ADC right channel.

step 16: Begin recording.

11.2.7.4 DISABLE ADC

This section will give the suggestion configuration steps to disable the ADC channel.

step 0: Keep ADC channel work and stop recording.

step 1: Configure the register `CODEC_0094_CFG` bit[1] to 0, to disable the zerocrossing detection function in ADC left channel. Configure the register `CODEC_0100_CFG` bit[1] to 0, to disable the zerocrossing detection function in ADC right channel

step 2: Configure the register `CODEC_008C_CFG` bit[3] to 0, to disable the ADC module in ADC left channel. Configure the register `CODEC_0098_CFG` bit[3] to 0, to disable the ADC module in ADC right channel.

step 3: Configure the register `CODEC_008C_CFG` bit[4] to 0, to disable the clock module in ADC left channel. Configure the register `CODEC_0098_CFG` bit[4] to 0, to disable the clock module in ADC right channel.

step 4: Configure the register `CODEC_008C_CFG` bit[5] to 0, to disable the ALC module in ADC left channel. Configure the register `CODEC_0098_CFG` bit[5] to 0, to disable the ALC module in ADC right channel.

step 5: Configure the register `CODEC_008C_CFG` bit[6] to 0, to disable the MIC module in ADC left channel. Configure the register `CODEC_0098_CFG` bit[6] to 0, to disable the MIC module in ADC right channel.

step 6: Configure the register `CODEC_008C_CFG` bit[7] to 0, to disable the reference voltage buffer in ADC left channel. Configure the register `CODEC_0098_CFG` bit[7] to 0, to disable the reference voltage buffer in ADC right channel.

step 7: Configure the register `CODEC_0080_CFG` bit[5] to 0, to disable the current source of ADC.

step 8: Configure the register `CODEC_008C_CFG` bit[2] to 0, to begin the initialization of the ADCL module. Configure the register `CODEC_0098_CFG` bit[2] to 0, to begin the initialization of the ADCR module.

step 9: Configure the register `CODEC_008C_CFG` bit[1] to 0, to begin the initialization of the left ALC module. Configure the register `CODEC_0098_CFG` bit[1] to 0, to begin the initialization of the right ALC module.

step 10: Configure the register `CODEC_008C_CFG` bit[4] to 0, to disable the clock module in ADC left channel. Configure the register `CODEC_0098_CFG` bit[4] to 0, to disable the clock module in ADC right channel.

step 11: Configure the register `CODEC_008C_CFG` bit[0] to 0, to begin the initialization of the left MIC module. Configure the register [0] to 0, to begin the initialization of the right MIC module.

11.2.7.5 ENABLE DAC

This section will give the suggest configuration steps to enable DAC.

step 0: Power up the CODEC according to the section 11.2.7.1 and input the mute signal.

step 1: Configure the register `CODEC_0080_CFG` bit[4] to 1, to enable the current source of DAC.

step 2: Configure the register `CODEC_0104_CFG` bit[6] to 1, to enable the reference voltage buffer of the DAC left channel. Configure the register `CODEC_0110_CFG` bit[6] to 1, to enable the reference voltage buffer of the DAC right channel.

step 3: Configure the register `CODEC_0104_CFG` bit[5:4] to 2'b10, to enable POP sound in the DAC left channel. Configure the register `CODEC_0110_CFG` bit[5:4] to 2'b10, to enable POP sound in the DAC right channel.

step 4: Configure the register `HPOUTL_CFG` bit[5] to 1, to enable the HPDRV module in the DAC left channel. Configure the register `HPOUTR_CFG` bit[5] to 1, to enable the HPDRV module in the DAC right channel.

step 5: Configure the register `HPOUTL_CFG` bit[4] to 1, to end the initialization of the HPDRV module in the DAC left channel. Configure the register `HPOUTR_CFG` bit[4] to 1, to end the initialization of the HPDRV module in the DAC right channel.

step 6: Configure the register `CODEC_0104_CFG` bit[3] to 1, to enable the reference voltage of DACL module. Configure the register `CODEC_0110_CFG` bit[3] to 1, to enable the reference voltage of DACR module.

step 7: Configure the register `CODEC_0104_CFG` bit[2] to 1, to enable the clock module of DACL module. Configure the register `CODEC_0110_CFG` bit[2] to 1, to enable the clock module of DACR module.

step 8: Configure the register `CODEC_0104_CFG` bit[1] to 1, to enable the DACL module. Configure the register `CODEC_0110_CFG` bit[1] to 1, to enable the DACR module.

step 9: Configure the register `CODEC_0104_CFG` bit[0] to 1, to end the initialization of the DACL module. Configure the register `CODEC_0110_CFG` bit[0] to 1, to end the initialization of the DACR module.

step 10: Configure the register `CODEC_0104_CFG` bit[7] to 1, to end the mute station of the HPDRV module in the DAC left channel. Configure the register `CODEC_0110_CFG` bit[7] to 1, to end the mute station of the HPDRV module in the DAC right channel.

step 11: Configure the register [GAIN_HPOUTL_CFG](#) bit[4:0], to select the gain of HPDRV module in the DAC left channel. Configure the register [GAIN_HPOUTR_CFG](#) bit[4:0], to select the gain of HPDRV module in the DAC right channel.

step 12: Play the music.

11.2.7.6 DISABLE DAC

This section will give the suggestion configuration steps to disable the ADC channel.

step 0: Keep the DAC channel work and input the mute signal.

step 1: Configure the register [GAIN_HPOUTL_CFG](#) bit[4:0] to 5'b0_0000, to select the gain of the HPDRV in the DAC left channel. Configure the register [GAIN_HPOUTR_CFG](#) bit[4:0] to 5'b0_0000, to select the gain of the HPDRV in the DAC right channel.

step 2: Configure the register [CODEC_0104_CFG](#) bit[7] to 0, to mute the HPDRV module in the DAC left channel. Configure the register [CODEC_0110_CFG](#) bit[7] to 0, to mute the HPDRV module in the DAC right channel.

step 3: Configure the register [HPOUTL_CFG](#) bit[4] to 0, to begin the initialization of the HPDRV module in the DAC left channel. Configure the register [HPOUTR_CFG](#) bit[4] to 0, to begin the initialization of the HPDRV module in the DAC right channel.

step 4: Configure the register [HPOUTL_CFG](#) bit[5] to 0, to disable the HPDRV module in the DAC left channel. Configure the register [HPOUTR_CFG](#) bit[5] to 0, to disable the HPDRV module in the DAC right channel.

step 5: Configure the register [CODEC_0104_CFG](#) bit[1] to 0 to disable the DACL module. Configure the register [CODEC_0110_CFG](#) bit[1] to 0 to disable the DACR module.

step 6: Configure the register [CODEC_0104_CFG](#) bit[2] to 0, to disable the clock module of DACL module. Configure the register [CODEC_0110_CFG](#) bit[2] to 0, to disable the clock module of DACR module.

step 7: Configure the register [CODEC_0104_CFG](#) bit[3] to 0, to disable the reference voltage of DACL module. Configure the register [CODEC_0110_CFG](#) bit[3] to 0, to disable the reference voltage of DACR module.

step 8: Configure the register [CODEC_0104_CFG](#) bit[5:4] to 2'b01 to initialize the POP sound in the DAC left channel. Configure the register [CODEC_0110_CFG](#) bit[5:4] to 2'b01 to initialize the POP sound in the DAC right channel.

step 9: Configure the register [CODEC_0104_CFG](#) bit[6] to 0, to disable the reference voltage buffer of the DAC left channel. Configure the register [CODEC_0110_CFG](#) bit[6] to 0, to disable the reference voltage buffer of the DAC right channel.

step 10: Configure the register [CODEC_0080_CFG](#) bit[4] to 0 to disable the current source of DAC.

step 11: Configure the register [CODEC_0104_CFG](#) bit[0] to 0, to initialize the DAC module in the DAC left channel. Configure the register [CODEC_0110_CFG](#) bit[0] to 0, to initialize the DAC module in the DAC right channel.

11.2.7.7 ENABLE ALC

The CODEC has the ALC function which can adjust the amplitude of the input analog signal automatically. You can use the register to enable or disable this function. We will give you some application reference about how to use this function.

The register's address which related to the ALCL function arrange from 0x011C to 0x0130.

The register's address which related to the ALCR function arrange from 0x0158 to 0x0170.

The steps which can enable the ALC function list below.

Step1: Set the Max level and Min level of the ALC need to control.

The register 0x114 and 0x118 control the max level of the ALC.

$\text{max\_level} = \{\text{reg0x118}, \text{reg0x114}\};$

The relationship between the max_level and the signal amplitude (dB) is equal to:

$\text{amplitude (dB)} = 20\log(\text{max\_level}/16'h7fff)$, 16'7fff means the full scale amplitude.

The register 0x11C and 0x120 control the min level of the ALC.

$\text{min\_level} = \{\text{reg0x120}, \text{reg0x11C}\};$

The relationship between the min_level and the signal amplitude (dB) is equal to:

$\text{amplitude(dB)} = 20\log(\text{min\_level}/16'h7fff)$, 16'7fff means the full scale amplitude.

Step2: According to the sample rate to choose the configuration, related to the register 0x110[2:0].

Note: Before we enable the ALC function, we should accord to the sample rate to choose the right configuration.

Step3: Enable the ALC module.

Set the reg0x124[6] to 1 to enable the ALC module.

Step4: Choose the ALC module to control the gain of the PGA.

Set the reg0x28[4] to 1 to choose the ALCL module to control the gain of the left channel PGA, Set the reg0x28[5] to 1 to choose the ALCL module to control the gain of the right channel PGA.

Step5: Observe the current ALC output gain, related to the register 0x138[4:0].

11.2.7.8 DISABLE ALC

If we want to disable the ALC function, please reference the following steps.

Step1: Disable the ALC module.

Set the reg0x124[6] to 0 to disable the ALC module, then the ALC output gain will keep to the last value.

Step2: Choose to use the register to control the gain of the PGA.

Set the reg0x28[5] to 0 to choose to use the register to control the gain of the PGA, then we can use the register 0x9C.

The ALC has two modes to control the gain of the PGA. If we use the register 0x49 to enable the automatically adjustment function of the ALC, the gain of the PGA will change automatically according to the amplitude of the input signal. If we don't use this function, we can use the register 0x10C to change the gain of the PGA manual. The prerequisite of these two modes is that we use the register 0x28 to choose to use the ALC module to control the gain of the PGA.

12 Peripherals

12.1 I2C Controller

12.1.1 Overview

The inter-integrated circuit (I2C) controller provides the master I2C interface which allows the CPU to read/write data from/to slave devices on the I2C bus through APB interface. The chip provides five I2C controllers. These I2C controllers use the DW_apb_i2c of Synopsys IP.

12.1.2 Feature

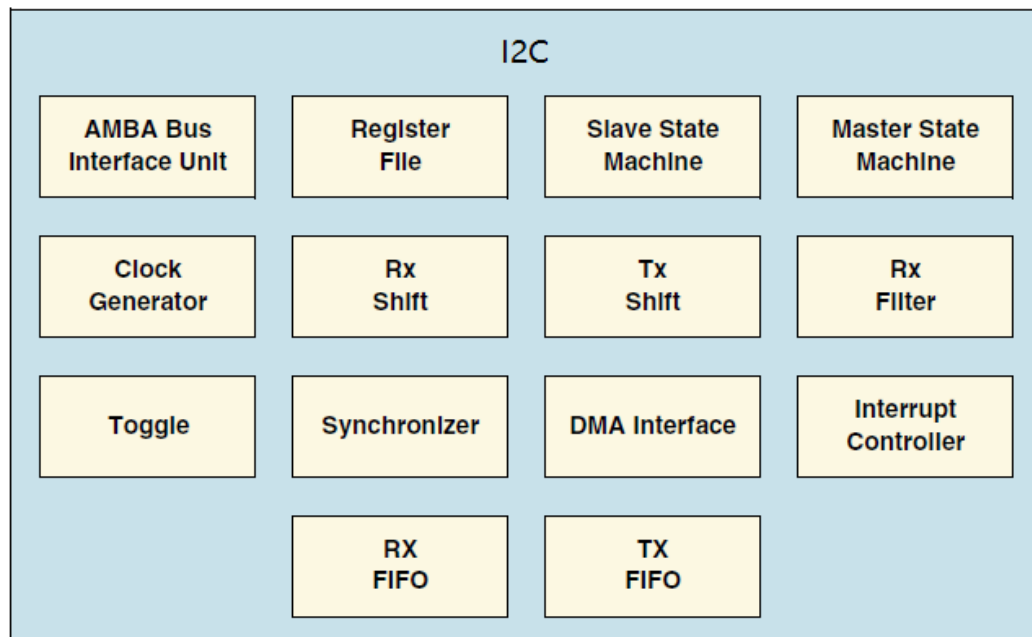
The I2C controller has the following features:

- Two-wire I2C serial interface—consists of a serial data line (SDA) and a serial clock (SCL)
- Three speeds:
 - Standard mode (0 to 100 Kb/s)
 - Fast mode (≤ 400 Kb/s) or fast mode plus (≤ 1000 Kb/s)
 - High-speed mode (≤ 3.4 Mb/s)
- Clock synchronization
- Master I2C
- 7- or 10-bit addressing
- 7- or 10-bit combined format transfers
- Bulk transmit mode
- Ignores CBUS addresses (an older ancestor of I2C that used to share the I2C bus)
- Transmit and receive buffers
 - 32x32 bits TX FIFOs and 64x8 bits RX FIFOs
- Interrupt or polled-mode operation
- Handles Bit and Byte waiting at all bus speeds
- DMA operation.
- Programmable SDA hold time (tHD;DAT)
- Bus clear feature
- Device ID feature
- SMBus/PMBus Support
- SMBus Slave detects and responds to ARP commands
- Support for APB data bus widths of 32 bits

12.1.3 Block Diagram

The I2C bus is a two-wire serial interface, consisting of a serial data line (SDA) and a serial clock (SCL). These wires carry information between the devices connected to the bus. Each device is recognized by a unique address and can operate as either a “transmitter” or “receiver,” depending on the function of the device. A master is a device that initiates a data transfer on the bus and generates the clock signals to permit that transfer. As follows the I2C Controller Block Diagram:

[Figure 12-1-1](#) The I2C Controller Block Diagram



The following define the file names and functions of the blocks in Figure 12-1-1:

- **AMBA Bus Interface Unit:** Takes the APB interface signals and translates them into a common generic interface that allows the register file to be bus protocol-agnostic.
- **Register File:** Contains configuration registers and is the interface with software.
- **Slave State Machine:** Follows the protocol for a slave and monitors bus for address match.
- **Master State Machine:** Generates the I2C protocol for the master transfers.
- **Clock Generator:** Calculates the required timing to do the following:
 - Generate the SCL clock when configured as a master
 - Check for bus idle
 - Generate a START and a STOP
 - Setup the data and hold the data
- **Rx Shift:** Takes data into the design and extracts it in byte format.
- **Tx Shift:** Presents data supplied by CPU for transfer on the I2C bus.
- **Rx Filter:** Detects the events in the bus; for example, start, stop and arbitration lost.
- **Toggle:** Generates pulses on both sides and toggles to transfer signals across clock domains.
- **Synchronizer:** Transfers signals from one clock domain to another.
- **DMA Interface:** Generates the handshaking signals to the central DMA controller in order to automate the data transfer without CPU intervention.
- **Interrupt Controller:** Generates the raw interrupt and interrupt flags, allowing them to be set and cleared.
- **RX FIFO/TX FIFO:** Holds the RX FIFO and TX FIFO register banks and controllers, along with their status levels.

12.1.4 Function Description

The I2C Controller module can operate in standard mode (with data rates 0 to 100 Kb/s), fast mode (with data rates less than or equal to 400 Kb/s), fast mode plus (with data rates less than or equal to 1000 Kb/s), high-speed mode (with data rates less than or equal to 3.4 Mb/s).

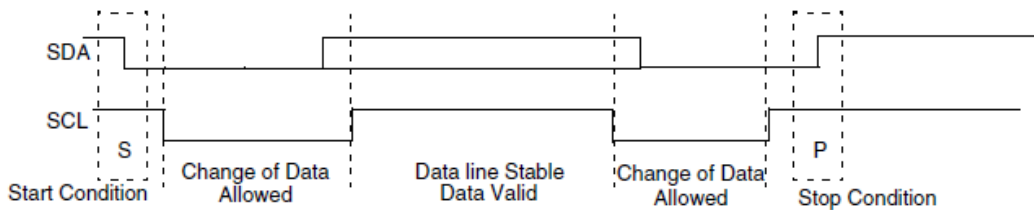
12.1.4.1 I2C Protocols

12.1.4.1.1 START and STOP Conditions

When the bus is idle, both the SCL and SDA signals are pulled high through external pull-up resistors on the bus. When the master wants to start a transmission on the bus, the master issues a START condition. This is defined to be a high-to-low transition of the SDA signal while SCL is 1. When the master wants to terminate the transmission, the

master issues a STOP condition. This is defined to be a low-to-high transition of the SDA line while SCL is 1. Figure 12-1-2 shows the timing of the START and STOP conditions. When data is being transmitted on the bus, the SDA line must be stable when SCL is 1.

Figure 12-1-2 START and STOP Condition

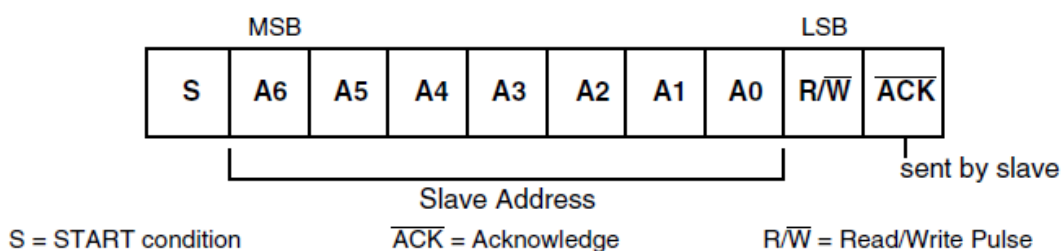


12.1.4.1.2 Addressing Slave Protocol

- 7-bit Address Format

During the 7-bit address format, the first seven bits (bits 7:1) of the first byte set the slave address and the LSB bit (bit 0) is the R/W bit as shown in Figure 12-1-3. When bit 0 (R/W) is set to 0, the master writes to the slave. When bit 0 (R/W) is set to 1, the master reads from the slave.

Figure 12-1-3 7-bit Address Format



- 10-bit Address Format

During 10-bit addressing, two bytes are transferred to set the 10-bit address. The transfer of the first byte contains the following bit definition. The first five bits (bits 7:3) notify the slaves that this is a 10-bit transfer followed by the next two bits (bits 2:1), which set the slaves address bits 9:8, and the LSB bit (bit 0) is the R/W bit. The second byte transferred sets bits 7:0 of the slave address. Figure 12-1-4 shows the 10-bit address format.

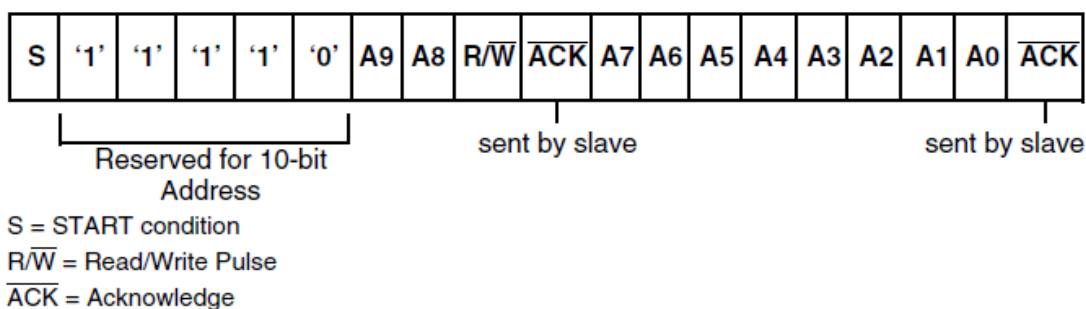


Figure 12-1-4 10-bit Address Format

12.1.4.1.3 Transmitting and Receiving Protocol

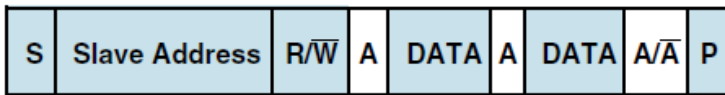
The master can initiate data transmission and reception to/from the bus, acting as either a master-transmitter or master-receiver. A slave responds to requests from the master to either transmit data or receive data to/from the bus, acting as either a slave-transmitter or slave-receiver, respectively.

- Master-Transmitter and Slave-Receiver

All data is transmitted in byte format, with no limit on the number of bytes transferred per data transfer. After the master sends the address and R/W bit or the master transmits a byte of data to the slave, the slave-receiver must respond with the acknowledge signal (ACK). When a slave-receiver does not respond with an ACK pulse, the master aborts the transfer by issuing a STOP condition. The slave must leave the SDA line high so that the master can abort the transfer.

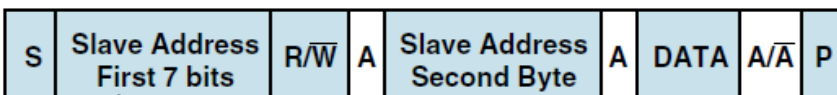
If the master-transmitter is transmitting data as shown in Figure 12-1-5, then the slave-receiver responds to the master-transmitter with an acknowledge pulse after every byte of data is received.

For 7-bit Address

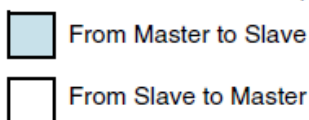


'0' (write)

For 10-bit Address



'11110xxx' '0' (write)



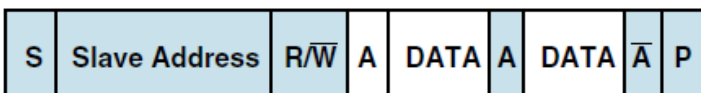
A = Acknowledge (SDA low)
 $\bar{A}$ = No Acknowledge (SDA high)
 S = START Condition
 P = STOP Condition

Figure 12-1-5 Master-Transmitter Protocol

- Master-Receiver and Slave-Transmitter

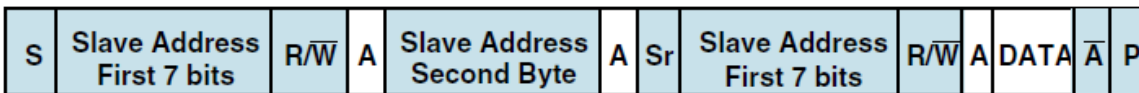
If the master is receiving data as shown in Figure 12-1-6, then the master responds to the slave-transmitter with an acknowledge pulse after a byte of data has been received, except for the last byte. This is the way the master-receiver notifies the slave-transmitter that this is the last byte. The slave-transmitter relinquishes the SDA line after detecting the No Acknowledge (NACK) so that the master can issue a STOP condition.

For 7-bit Address

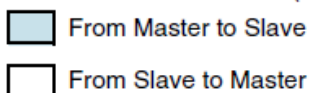


'1' (read)

For 10-bit Address



'11110xxx' '0' (write) '11110xxx' '1' (read)



A = Acknowledge (SDA low) R = RESTART Condition
 $\bar{A}$ = No Acknowledge (SDA high) P = STOP Condition
 S = START Condition

Figure 12-1-6 Master-Receiver Protocol

12.1.4.1.4 START BYTE Transfer Protocol

The START BYTE transfer protocol is set up for systems that do not have an on-board dedicated I2C hardware module. When the I2C controller is a master, it supports the generation of START BYTE transfers at the beginning of every transfer in case a slave device requires it.

This protocol consists of seven zeros being transmitted followed by a 1, as illustrated in Figure 12-1-7. This allows the processor that is polling the bus to under-sample the address phase until 0 is detected. Once the microcontroller detects a 0, it switches from the under sampling rate to the correct rate of the master.

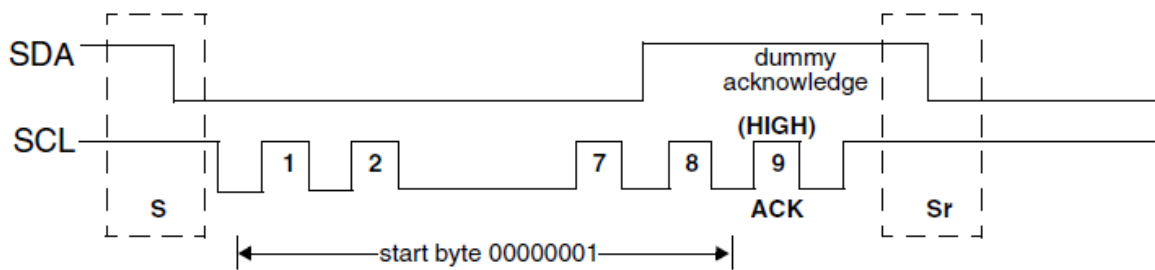


Figure 12-1-7 START BYTE Transfer

The START BYTE procedure is as follows:

1. Master generates a START condition.
2. Master transmits the START byte (0000 0001).
3. Master transmits the ACK clock pulse. (Present only to conform with the byte handling format used on the bus)
4. No slave sets the ACK signal to 0.
5. Master generates a RESTART (R) condition.

A hardware receiver does not respond to the START BYTE because it is a reserved address and resets after the RESTART condition is generated.

12.1.4.1.5 Tx FIFO Management and START, STOP and RESTART Generation

The component generates a STOP on the bus whenever the Tx FIFO becomes empty. If RESTART generation capability is enabled, the component generates a RESTART when the direction of the transfer in the Tx FIFO commands changes from Read to Write or vice-versa; if RESTART is not enabled, a STOP followed by a START is generated in this situation.

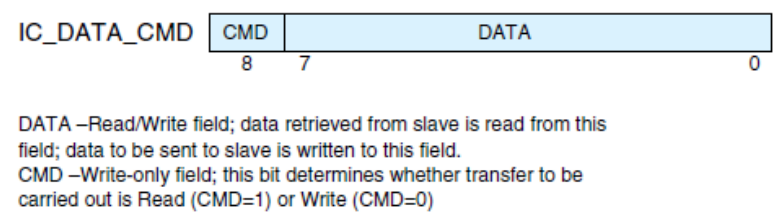


Figure 12-1-8 IC_DATA_CMD Register

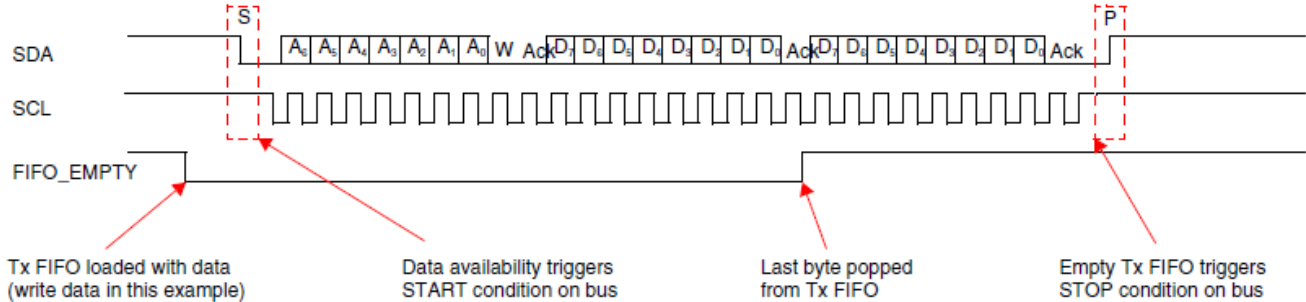


Figure 12-1-9 Master Transmitter — Tx FIFO Becomes Empty

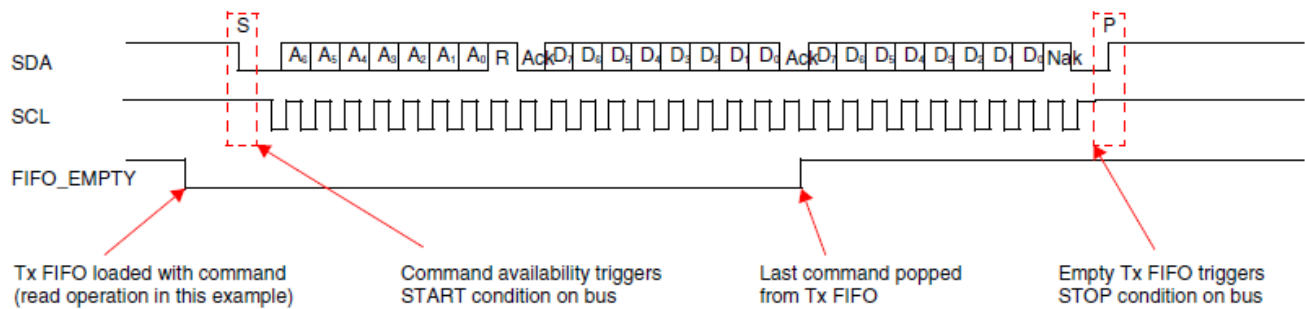


Figure 12-1-10 Master Receiver — Tx FIFO Becomes Empty

12.1.4.2 Clock Synchronization

When two or more masters try to transfer information on the bus at the same time, they must arbitrate and synchronize the SCL clock. All masters generate their own clock to transfer messages. Data is valid only during the high period of SCL clock. Clock synchronization is performed using the wired-AND connection to the SCL signal. When the master transitions the SCL clock to 0, the master starts counting the low time of the SCL clock and transitions the SCL clock signal to 1 at the beginning of the next clock period. However, if another master is holding the SCL line to 0, then the master goes into a HIGH wait state until the SCL clock line transitions to 1.

All masters then count off their high time, and the master with the shortest high time transitions the SCL line to 0. The masters then counts out their low time and the one with the longest low time forces the other master into a HIGH wait state. Therefore, a synchronized SCL clock is generated, which is illustrated in Figure 12-1-11. Optionally, slaves may hold the SCL line low to slow down the timing on the I2C bus.

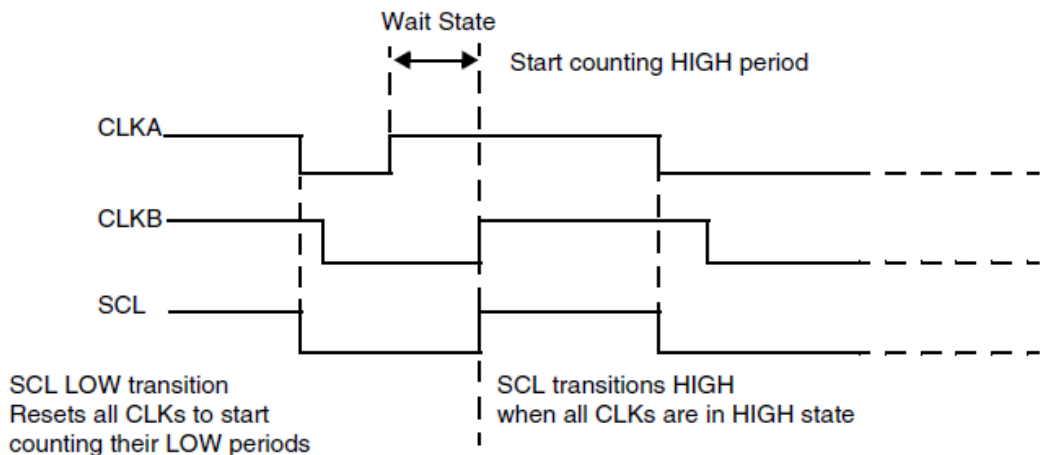


Figure 12-1-11 Multi-Master Clock Synchronization

12.1.4.3 IC_CLK Frequency Configuration

When the I2C controller is configured as a Standard (SS), Fast (FS)/Fast-Mode Plus (FM+), or High Speed (HS) master, the *CNT* registers must be set before any I2C bus transaction can take place in order to ensure proper I/O timing. The *CNT* registers are:

- IC_SS_SCL_HCNT
- IC_SS_SCL_LCNT
- IC_FS_SCL_HCNT
- IC_FS_SCL_LCNT
- IC_HS_SCL_HCNT
- IC_HS_SCL_LCNT

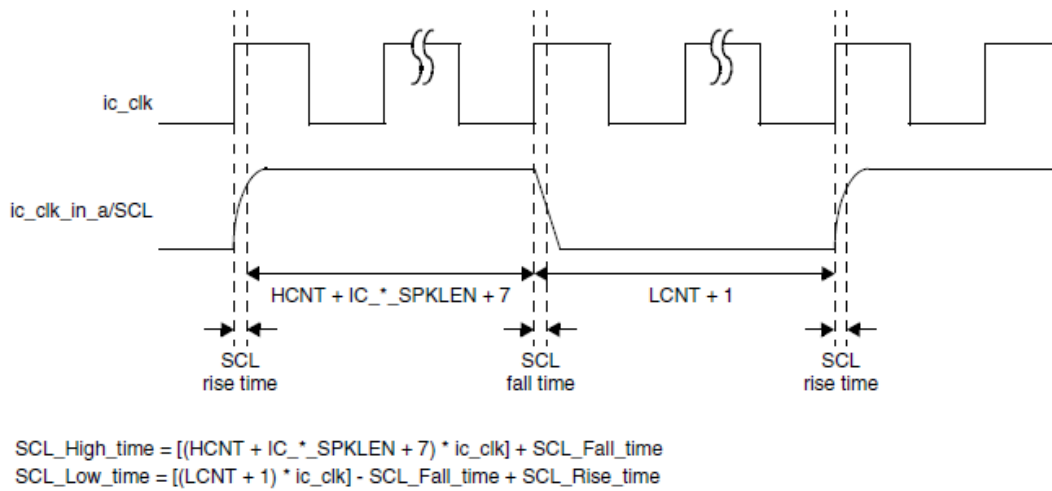


Figure 12-1-12 Impact of SCL Rise Time and Fall Time on Generated SCL

12.1.4.4 APB Interface

The host processor accesses data, control, and status information on the I2C peripheral through the AMBA APB 2.0 interface. This peripheral supports APB data bus widths of 32 bits. Figure 12-1-13 shows the read/write buses:

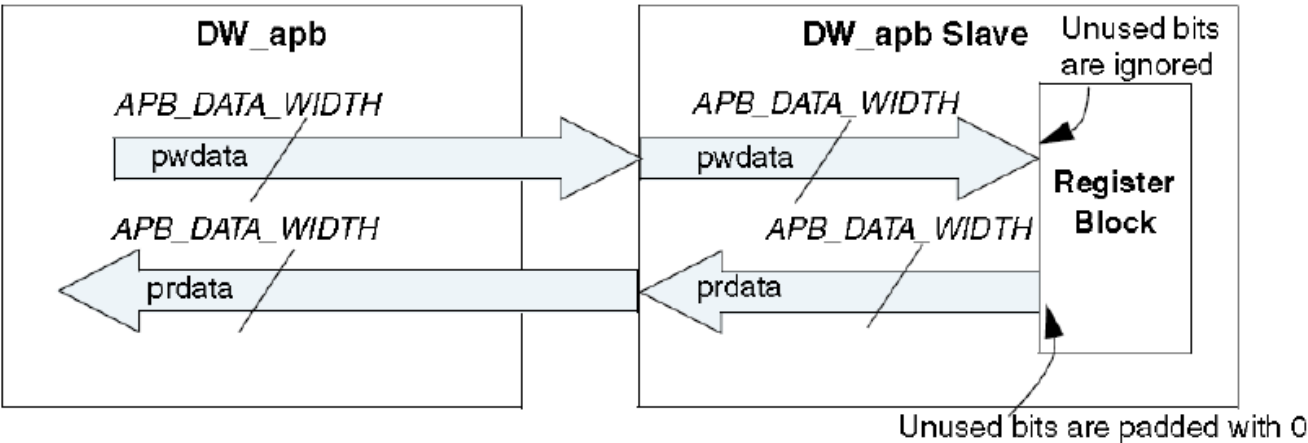
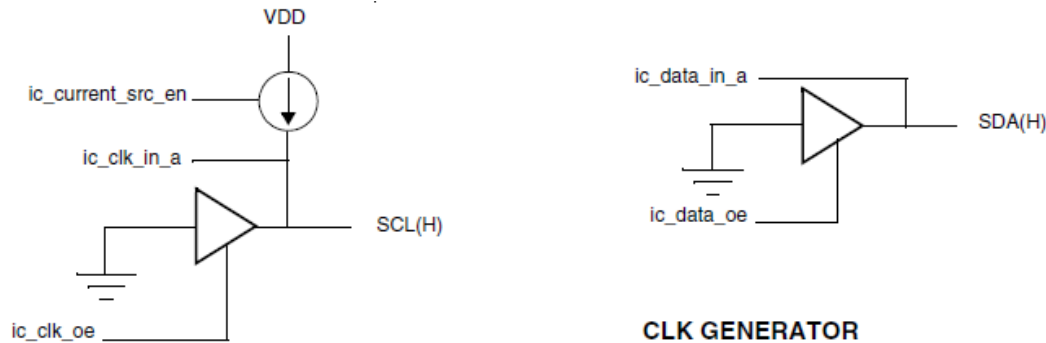


Figure 12-1-13 Read/Write Buses Between the DW_apb and an APB Slave

12.1.4.5 I/O Connections

As illustrated in Figure 12-1-14, the I2C interface consists of two wires, a clock (SCL) and data (SDA). For high-speed systems, the names are SCLH and SDAH. For high-speed mode, a current source pull-up may be used on the SCLH line. It is enabled during some active master transactions. The SDA and SDAH connections are the same at any speed. There are no special connections required for the DesignWare APB slave interface side of the I2C.



12.1.5 Register Description

12.1.5.1 Register Summary

The I2C base addresses are as follows:

- The base address for I2C0 registers is 0x9140_5000.
- The base address for I2C1 registers is 0x9140_6000.
- The base address for I2C2 registers is 0x9140_7000.
- The base address for I2C3 registers is 0x9140_8000.
- The base address for I2C4 registers is 0x9140_9000.

Table 12-1-1 Summary of I2C registers

Offset Address	Register	Description	Page
0x00	IC_CON	I2C Control Register. This register can be written only when the I2C controller is disabled, which corresponds...	12-1-2
0x04	IC_TAR	I2C Target Address Register If the configuration parameter I2C_DYNAMIC_TAR_UPDATE is set to 'No'...	12-1-3
0x08	IC_SAR	I2C Slave Address Register	12-1-4
0x0c	IC_HS_MADDR	I2C High Speed Master Mode Code Address Register	12-1-5
0x10	IC_DATA_CMD	I2C Rx/Tx Data Buffer and Command Register; this is the register the CPU writes to when filling...	12-1-6
0x14	IC_SS_SCL_HCNT	Standard Speed I2C Clock SCL High Count Register	12-1-7
0x14	IC_UFM_SCL_HCNT	Ultra-Fast Speed I2C Clock SCL High Count Register	12-1-8
0x18	IC_SS_SCL_LCNT	Standard Speed I2C Clock SCL Low Count Register	12-1-9
0x18	IC_UFM_SCL_LCNT	Ultra-Fast Speed I2C Clock SCL Low Count Register	12-1-10
0x1c	IC_FS_SCL_HCNT	Fast Mode or Fast Mode Plus I2C Clock SCL High Count Register	12-1-11
0x1c	IC_UFM_TBUF_CNT	Ultra-Fast Speed mode TBuf Idle Count Register	12-1-12
0x20	IC_FS_SCL_LCNT	Fast Mode or Fast Mode Plus I2C Clock SCL Low Count Register	12-1-13
0x24	IC_HS_SCL_HCNT	High Speed I2C Clock SCL High Count Register	12-1-14

Offset Address	Register	Description	Page
0x28	IC_HS_SCL_LCNT	High Speed I2C Clock SCL Low Count Register	12-1-15
0x2c	IC_INTR_STAT	I2C Interrupt Status Register Each bit in this register has a corresponding mask bit in the IC_INTR_MASK...	12-1-16
0x30	IC_INTR_MASK	I2C Interrupt Mask Register. These bits mask their corresponding interrupt status bits. This register...	12-1-17
0x34	IC_RAW_INTR_STAT	I2C Raw Interrupt Status Register Unlike the IC_INTR_STAT register, these bits are not masked so...	12-1-18
0x38	IC_RX_TL	I2C Receive FIFO Threshold Register	12-1-19
0x3c	IC_TX_TL	I2C Transmit FIFO Threshold Register	12-1-20
0x40	IC_CLR_INTR	Clear Combined and Individual Interrupt Register	12-1-21
0x44	IC_CLR_RX_UNDER	Clear RX_UNDER Interrupt Register	12-1-22
0x48	IC_CLR_RX_OVER	Clear RX_OVER Interrupt Register	12-1-23
0x4c	IC_CLR_TX_OVER	Clear TX_OVER Interrupt Register	12-1-24
0x50	IC_CLR_RD_REQ	Clear RD_REQ Interrupt Register	12-1-25
0x54	IC_CLR_TX_ABRT	Clear TX_ABRT Interrupt Register	12-1-26
0x58	IC_CLR_RX_DONE	Clear RX_DONE Interrupt Register	12-1-27
0x5c	IC_CLR_ACTIVITY	Clear ACTIVITY Interrupt Register	12-1-28
0x60	IC_CLR_STOP_DET	Clear STOP_DET Interrupt Register	12-1-29
0x64	IC_CLR_START_DET	Clear START_DET Interrupt Register	12-1-30
0x68	IC_CLR_GEN_CALL	Clear GEN_CALL Interrupt Register	12-1-31
0x6c	IC_ENABLE	I2C Enable Register	12-1-32

Offset Address	Register	Description	Page
0x70	IC_STATUS	I2C Status Register This is a read-only register used to indicate the current transfer status...	12-1-33
0x74	IC_TXFLR	I2C Transmit FIFO Level Register This register contains the number of valid data entries in the...	12-1-34
0x78	IC_RXFLR	I2C Receive FIFO Level Register This register contains the number of valid data entries in the receive...	12-1-35
0x7c	IC_SDA_HOLD	I2C SDA Hold Time Length Register The bits [15:0] of this register are used to control the hold...	12-1-36
0x80	IC_TX_ABRT_SOURCE	I2C Transmit Abort Source Register This register has 32 bits that indicate the source of the TX_ABRT...	12-1-37
0x84	IC_SLV_DATA_NACK_ONLY	Generate Slave Data NACK Register The register is used to generate a NACK for the data part of...	12-1-38
0x88	IC_DMA_CR	DMA Control Register This register is only valid when I2C controller is configured with a set of DMA...	12-1-39
0x8c	IC_DMA_TDLR	DMA Transmit Data Level Register This register is only valid when the I2C controller is configured...	12-1-40
0x90	IC_DMA_RDLR	I2C Receive Data Level Register This register is only valid when I2C controller is configured with...	12-1-41
0x94	IC_SDA_SETUP	I2C SDA Setup Register This register controls the amount of time delay (in terms of number of ic_clk...	12-1-42
0x98	IC_ACK_GENERAL_CALL	I2C ACK General Call Register The register controls whether I2C controller responds with a ACK or NACK...	12-1-43
0x9c	IC_ENABLE_STATUS	I2C Enable Status Register The register is used to report the I2C controller hardware status when the...	12-1-44
0xa0	IC_FS_SPKLEN	I2C SS, FS or FM+ spike suppression limit This register is used to store the duration, measured...	12-1-45
0xa0	IC_UFM_SPKLEN	I2C UFM spike suppression limit This register is used to store the duration, measured in ic_clk...	12-1-46
0xa4	IC_HS_SPKLEN	I2C HS spike suppression limit register This register is used to store the duration, measured in...	12-1-47
0xa8	IC_CLR_RESTART_DET	Clear RESTART_DET Interrupt Register	12-1-48

Offset Address	Register	Description	Page
0xac	IC_SCL_STUCK_AT_LOW_TIMEOUT	I2C SCL Stuck at Low Timeout This register is used to store the duration, measured in ic_clk cycles,...	12-1-49
0xb0	IC_SDA_STUCK_AT_LOW_TIMEOUT	I2C SDA Stuck at Low Timeout This register is used to store the duration, measured in ic_clk cycles,...	12-1-50
0xb4	IC_CLR_SCL_STUCK_DET	Clear SCL Stuck at Low Detect Interrupt Register	12-1-51
0xb8	IC_DEVICE_ID	I2C Device-ID Register This Register contains the Device-ID of the component which includes 12-bits...	12-1-52
0xbc	IC_SMBUS_CLK_LOW_SEXT	SMBus Slave Clock Extend Timeout Register This Register contains the Timeout value used to determine...	12-1-53
0xc0	IC_SMBUS_CLK_LOW_MEXT	SMBus Master Clock Extend Timeout Register This Register contains the Timeout value used to determine...	12-1-54
0xc4	IC_SMBUS_THIGH_MAX_IDLE_COUNT	SMBus Master THigh MAX Bus-idle count Register This register programs the Bus-idle time period...	12-1-55
0xc8	IC_SMBUS_INTR_STAT	SMBUS Interrupt Status Register Each bit in this register has a corresponding mask bit in the IC_SMBUS_INTR_MASK...	12-1-56
0xcc	IC_SMBUS_INTR_MASK	SMBus Interrupt Mask Register	12-1-57
0xd0	IC_SMBUS_RAW_INTR_STAT	SMBus Raw Interrupt Status Register Unlike the IC_SMBUS_INTR_STAT register, these bits are not...	12-1-58
0xd4	IC_CLR_SMBUS_INTR	SMBus Clear Interrupt Register	12-1-59
0xd8	IC_OPTIONAL_SAR	I2C Optional Slave Address Register Optional Slave address for I2C in SMBus Mode. A same restriction...	12-1-60
0xdc	IC_SMBUS_UDID_LSB	SMBUS ARP UDID LSB Register This Register can be written only when the I2C controller is disabled,...	12-1-61
0xf4	IC_COMP_PARAM_1	Component Parameter Register 1 Note This is a constant read-only register that contains encoded...	12-1-62
0xf8	IC_COMP_VERSION	I2C Component Version Register	12-1-63

Offset Address	Register	Description	Page
0xfc	IC_COMP_TYPE	I2C Component Type Register	12-1-64

12.1.5.2 Register Description

12.1.5.2.1 IC_CON

- Name: I2C Control Register.
- Description: I2C Control Register. This register can be written only when the I2C controller is disabled, which corresponds to the IC_ENABLE[0] register being set to 0. Writes at other times have no effect.
- Offset Address: 0x00

Table 12-1-2 Fields for Register: IC_CON

Bits	Access	Name	Description & Reset Value
31:20	R	RSVD_IC_CON_2	IC_CON_2 Reserved bits - Read Only Exists: Always
19	R/W	SMBUS_PERSISTENT_SLV_ADDR_EN	The bit controls to enable DW_apb_i2c slave as persistent or non persistent slave. If the slave is non-PSA then DW_apb_i2c slave device clears the Address valid flag for both General and Directed Reset ARP command else the address valid flag will always set to 1. This bit is applicable only in Slave mode. Reset Value :IC_PERSISTANT_SLV_ADDR_DEFAULT. Values: 0x1 (ENABLED): SMBus Persistent Slave address control is enabled. 0x0 (DISABLED): SMBus Persistent Slave address control is disabled. Exists: IC_SMBUS_ARP==1
18	R/W	SMBUS_ARP_EN	This bit controls whether DW_apb_i2c should enable Address Resolution Logic in SMBus Mode. The Slave mode will decode the Address Resolution Protocol commands and respond to it. The DW_apb_i2c slave also includes the generation/validity of PEC byte for Address Resolution Protocol commands. This bit is applicable only in Slave mode. Reset Value: 0x0. Values: 0x1 (ENABLED): SMBus ARP control is enabled. 0x0 (DISABLED): SMBus ARP control is disabled. Exists: IC_SMBUS_ARP==1

Bits	Access	Name	Description & Reset Value
17	R/W	SMBUS_SLAVE_QUICK_EN	If this bit is set to 1, DW_apb_i2c slave only receives Quick commands in SMBus Mode. If this bit is set to 0, DW_apb_i2c slave receives all bus protocols but not Quick commands. This bit is applicable only in slave mode. Reset value: 0x0. Values: 0x1 (ENABLED): SMBus SLave is enabled to receive Quick command. 0x0 (DISABLED): SMBus SLave is disabled to receive Quick command. Exists: IC_SMBUS==1
16	R/W	OPTIONAL_SAR_CTRL	Enables the usage of IC_OPTIONAL_SAR register. If IC_OPTIONAL_SAR =1, IC_OPTIONAL_SAR value is used as additional slave address. User must program a valid address in IC_OPTIONAL_SAR before writing 1 to this field. If IC_OPTIONAL_SAR =0, IC_OPTIONAL_SAR value is not used as additional slave address. In this mode only one I2C slave address is used. Reset value: IC_OPTIONAL_SAR_DEFAULT. Values: 0x1 (ENABLED): Optional SAR Address Register is enabled. 0x0 (DISABLED): Optional SAR Address Register is disabled. Exists: IC_OPTIONAL_SAR==1
15:12	R	RSVD_IC_CON_1	IC_CON_1 Reserved bits - Read Only Exists: Always
11	R/W	BUS_CLEAR_FEATURE_CTRL	In Master mode: 1'b1: Bus Clear Feature is enabled. 1'b0: Bus Clear Feature is Disabled. In Slave mode, this register bit is not applicable. Reset value: 0x0. Values: 0x1 (ENABLED): Bus Clear Feature ois enabled. 0x0 (DISABLED): Bus Clear Feature is disabled. Exists: IC_BUS_CLEAR_FEATURE==1

Bits	Access	Name	Description & Reset Value
10	*Varies	STOP_DET_IF_MASTER_ACTIVE	In Master mode: 1'b1: issues the STOP_DET interrupt only when master is active. 1'b0: issues the STOP_DET irrespective of whether master is active or not. Reset value: 0x0. Values: 0x1 (ENABLED): Master issues the STOP_DET interrupt only when master is active 0x0 (DISABLED): Master issues the STOP_DET interrupt irrespective of whether master is active or not Exists: Always Memory Access: (IC_STOP_DET_IF_MASTER_ACTIVE==1) ? read-write : read-only
9	*Varies	RX_FIFO_FULL_HLD_CTRL	This bit controls whether DW_apb_i2c should hold the bus when the Rx FIFO is physically full to its RX_BUFFER_DEPTH, as described in the IC_RX_FULL_HLD_BUS_EN parameter. Reset value: 0x0. Values: 0x1 (ENABLED): Hold bus when RX_FIFO is full 0x0 (DISABLED): Overflow when RX_FIFO is full Exists: Always Memory Access: (IC_RX_FULL_HLD_BUS_EN==1) ? read-write : read-only
8	R/W	TX_EMPTY_CTRL	This bit controls the generation of the TX_EMPTY interrupt, as described in the IC_RAW_INTR_STAT register. Reset value: 0x0. Values: 0x1 (ENABLED): Controlled generation of TX_EMPTY interrupt 0x0 (DISABLED): Default behaviour of TX_EMPTY interrupt Exists: Always

Bits	Access	Name	Description & Reset Value
7	R/W	STOP_DET_IFADDRESSED	In slave mode: 1'b1: issues the STOP_DET interrupt only when it is addressed. 0'b0: issues the STOP_DET irrespective of whether it's addressed or not. Reset value: 0x0 NOTE: During a general call address, this slave does not issue the STOP_DET interrupt if STOP_DET_IF_ADDRESSED = 1'b1, even if the slave responds to the general call address by generating ACK. The STOP_DET interrupt is generated only when the transmitted address matches the slave address (SAR). Values: 0x1 (ENABLED): slave issues STOP_DET intr only if addressed 0x0 (DISABLED): slave issues STOP_DET intr always Exists: Always
6	R/W	IC_SLAVE_DISABLE	This bit controls whether I2C has its slave disabled, which means once the presetrn signal is applied, then this bit takes on the value of the configuration parameter IC_SLAVE_DISABLE. You have the choice of having the slave enabled or disabled after reset is applied, which means software does not have to configure the slave. By default, the slave is always enabled (in reset state as well). If you need to disable it after reset, set this bit to 1. If this bit is set (slave is disabled), DW_apb_i2c functions only as a master and does not perform any action that requires a slave. Reset value: IC_SLAVE_DISABLE configuration parameter NOTE: Software should ensure that if this bit is written with 0, then bit 0 should also be written with a 0. Values: 0x1 (SLAVE_DISABLED): Slave mode is disabled 0x0 (SLAVE_ENABLED): Slave mode is enabled Exists: Always

Bits	Access	Name	Description & Reset Value
5	R/W	IC_RESTART_EN	Determines whether RESTART conditions may be sent when acting as a master. Some older slaves do not support handling RESTART conditions; however, RESTART conditions are used in several DW_apb_i2c operations. When RESTART is disabled, the master is prohibited from performing the following functions: - Sending a START BYTE - Performing any high-speed mode operation - High-speed mode operation - Performing direction changes in combined format mode - Performing a read operation with a 10-bit address By replacing RESTART condition followed by a STOP and a subsequent START condition, split operations are broken down into multiple DW_apb_i2c transfers. If the above operations are performed, it will result in setting bit 6 (TX_ABRT) of the IC_RAW_INTR_STAT register. Reset value: IC_RESTART_EN configuration parameter Values: 0x1 (ENABLED): Master restart enabled 0x0 (DISABLED): Master restart disabled Exists: Always
4	R/W	IC_10BITADDR_MASTER	If the I2C_DYNAMIC_TAR_UPDATE configuration parameter is set to 'No' (0), this bit is named IC_10BITADDR_MASTER and controls whether the DW_apb_i2c starts its transfers in 7- or 10-bit addressing mode when acting as a master. If I2C_DYNAMIC_TAR_UPDATE is set to 'Yes' (1), the function of this bit is handled by bit 12 of IC_TAR register, and becomes a read-only copy called IC_10BITADDR_MASTER_rd_only. 0: 7-bit addressing 1: 10-bit addressing Reset value: IC_10BITADDR_MASTER configuration parameter Values: 0x1 (ADDR_10BITS): Master 10Bit addressing mode 0x0 (ADDR_7BITS): Master 7Bit addressing mode Exists: I2C_DYNAMIC_TAR_UPDATE == 0

Bits	Access	Name	Description & Reset Value
3	R/W	IC_10BITADDR_SLAVE	When acting as a slave, this bit controls whether the DW_apb_i2c responds to 7- or 10-bit addresses. 0: 7-bit addressing. The DW_apb_i2c ignores transactions that involve 10-bit addressing; for 7-bit addressing, only the lower 7 bits of the IC_SAR register are compared. 1: 10-bit addressing. The DW_apb_i2c responds to only 10-bit addressing transfers that match the full 10 bits of the IC_SAR register. Reset value: IC_10BITADDR_SLAVE configuration parameter Values: 0x1 (ADDR_10BITS): Slave 10Bit addressing 0x0 (ADDR_7BITS): Slave 7Bit addressing Exists: Always
2:1	R/W	SPEED	These bits control at which speed the DW_apb_i2c operates; its setting is relevant only if one is operating the DW_apb_i2c in master mode. Hardware protects against illegal values being programmed by software. These bits must be programmed appropriately for slave mode also, as it is used to capture correct value of spike filter as per the speed mode. This register should be programmed only with a value in the range of 1 to IC_MAX_SPEED_MODE; otherwise, hardware updates this register with the value of IC_MAX_SPEED_MODE. 1: standard mode (100 kbit/s) 2: fast mode (<=400 kbit/s) or fast mode plus (<=1000Kbit/s) 3: high speed mode (3.4 Mbit/s) Note: This field is not applicable when IC_ULTRA_FAST_MODE=1 Reset value: IC_MAX_SPEED_MODE configuration Values: 0x1 (STANDARD): Standard Speed mode of operation 0x2 (FAST): Fast or Fast Plus mode of operation 0x3 (HIGH): High Speed mode of operation Exists: Always
0	R/W	MASTER_MODE	This bit controls whether the DW_apb_i2c master is enabled. Reset value: IC_MASTER_MODE configuration parameter NOTE: Software should ensure that if this bit is written with '1' then bit 6 should also be written with a '1'. Values: 0x1 (ENABLED): Master mode is enabled 0x0 (DISABLED): Master mode is disabled Exists: Always

12.1.5.2.2 IC_TAR

- Name: I2C Target Address Register.

- Description: I2C Target Address Register

If the configuration parameter I2C_DYNAMIC_TAR_UPDATE is set to 'No'(0), this register is 12 bits wide, and bits 31:12 are reserved. This register can be written to only when IC_ENABLE[0] is set to 0.

However, if I2C_DYNAMIC_TAR_UPDATE = 1, then the register becomes 13 bits wide. In this case, writes to IC_TAR succeed when one of the following conditions are true:

- I2C controller is NOT enabled (IC_ENABLE[0] is set to 0); or
- I2C controller is enabled (IC_ENABLE[0]=1); AND I2C controller is NOT engaged in any Master (tx, rx) operation (IC_STATUS[5]=0); AND I2C controller is enabled to operate in Master mode (IC_CON[0]=1); AND there are NO entries in the TX FIFO (IC_STATUS[2]=1)

You can change the TAR address dynamically without losing the bus, only if the following conditions are met.

- I2C controller is enabled (IC_ENABLE[0]=1); AND IC_EMPTYFIFO_HOLD_MASTER_EN configuration parameter is set to 1; AND I2C controller is enabled to operate in Master mode (IC_CON[0]=1); AND there are NO entries in the Tx FIFO and the master is in HOLD state (IC_INTR_STAT[13]=1).

Note: If the software or application is aware that the I2C controller is not using the TAR address for the pending commands in the Tx FIFO, then it is possible to update the TAR address even while the Tx FIFO has entries (IC_STATUS[2]= 0).

- It is not necessary to perform any write to this register if I2C controller is enabled as an I2C slave only.

- Offset Address: 0x04

Table 12-1-3 Fields for Register: IC_TAR

Bits	Access	Name	Description & Reset Value
31:17	R	RSVD_IC_TAR_2	IC_TAR_2 Reserved bits - Read Only Exists: Always
16	R/W	SMBUS_QUICK_CMD	If bit 11 (SPECIAL) is set to 1, then this bit indicates whether a Quick command is to be performed by the DW_apb_i2c. Reset value: 0x0 Values: 0x1 (ENABLED): Enables programming of QUICK-CMD transmission 0x0 (DISABLED): Disables programming of QUICK-CMD transmission Exists: IC_SMBUS == 1
15:14	R	RSVD_IC_TAR_1	IC_TAR_1 Reserved bits - Read Only Exists: Always

Bits	Access	Name	Description & Reset Value
13	R/W	DEVICE_ID	If bit 11 (SPECIAL) is set to 1, then this bit indicates whether a Device-ID of a particular slave mentioned in IC_TAR[9:0] is to be performed by the DW_apb_i2c Master. 0: Device-ID is not performed and checks ic_tar[10] to perform either general call or START byte command 1: Device-ID transfer is performed and bytes based on the number of read commands in the Tx-FIFO are received from the targeted slave and put in the Rx-FIFO. Reset value: 0x0 Values: 0x1 (ENABLED): Enables programming of DEVICE-ID transmission 0x0 (DISABLED): Disables programming of DEVICE-ID transmission Exists: IC_DEVICE_ID == 1
12	R/W	IC_10BITADDR_MASTER	This bit controls whether the DW_apb_i2c starts its transfers in 7- or 10-bit addressing mode when acting as a master. 0: 7-bit addressing 1: 10-bit addressing Reset value: IC_10BITADDR_MASTER configuration parameter Values: 0x1 (ADDR_10BITS): Address 10Bit transmission format 0x0 (ADDR_7BITS): Address 7Bit transmission format Exists: I2C_DYNAMIC_TAR_UPDATE
11	R/W	SPECIAL	This bit indicates whether software performs a Device-ID or General Call or START BYTE command. 0: ignore bit 10 GC_OR_START and use IC_TAR normally 1: perform special I2C command as specified in Device_ID or GC_OR_START bit Reset value: 0x0 Values: 0x1 (ENABLED): Enables programming of GENERAL_CALL or START_BYTE transmission 0x0 (DISABLED): Disables programming of GENERAL_CALL or START_BYTE transmission Exists: Always

Bits	Access	Name	Description & Reset Value
10	R/W	GC_OR_START	If bit 11 (SPECIAL) is set to 1 and bit 13(Device-ID) is set to 0, then this bit indicates whether a General Call or START byte command is to be performed by the DW_apb_i2c. 0: General Call Address - after issuing a General Call, only writes may be performed. Attempting to issue a read command results in setting bit 6 (TX_ABRT) of the IC_RAW_INTR_STAT register. The DW_apb_i2c remains in General Call mode until the SPECIAL bit value (bit 11) is cleared. 1: START BYTE Reset value: 0x0 Values: 0x1 (START_BYTE): START byte transmission 0x0 (GENERAL_CALL): GENERAL_CALL byte transmission Exists: Always
9:0	R/W	IC_TAR	This is the target address for any master transaction. When transmitting a General Call, these bits are ignored. To generate a START BYTE, the CPU needs to write only once into these bits. If the IC_TAR and IC_SAR are the same, loopback exists but the FIFOs are shared between master and slave, so full loopback is not feasible. Only one direction loopback mode is supported (simplex), not duplex. A master cannot transmit to itself; it can transmit to only a slave. Reset value: IC_DEFAULT_TAR_SLAVE_ADDR configuration parameter Exists: Always

12.1.5.2.3 IC_SAR

- Name: I2C Slave Address Register.
- Description: I2C Slave Address Register
- Offset Address: 0x08

Table 12-1-4 Fields for Register: IC_SAR

Bits	Access	Name	Description & Reset Value
31:10	R	RSVD_IC_SAR	IC_SAR Reserved bits - Read Only Exists: Always
9:0	R/W	IC_SAR	The IC_SAR holds the slave address when the I2C is operating as a slave. For 7-bit addressing, only IC_SAR[6:0] is used. This register can be written only when the I2C interface is disabled, which corresponds to the IC_ENABLE[0] register being set to 0. Writes at other times have no effect. Note: The default values cannot be any of the reserved address locations: that is, 0x00 to 0x07, or 0x78 to 0x7f. The correct operation of the device is not guaranteed if you program the IC_SAR or IC_TAR to a reserved value. Refer to Table I2C/SMBus Definition of Bits in First Byte for a complete list of these reserved values. Reset value: IC_DEFAULT_SLAVE_ADDR configuration parameter Exists: Always

12.1.5.2.4 IC_HS_MADDR

- Name: I2C High Speed Master Mode Code Address Register.
- Description: I2C High Speed Master Mode Code Address Register
- Offset Address: 0x0c

Table 12-1-5 Fields for Register: IC_HS_MADDR

Bits	Access	Name	Description & Reset Value
31:3	R	RSVD_IC_HS_MAR	IC_HS_MAR Reserved bits - Read Only Exists: Always
2:0	R/W	IC_HS_MAR	This bit field holds the value of the I2C HS mode master code. HS-mode master codes are reserved 8-bit codes (00001xxx) that are not used for slave addressing or other purposes. Each master has its unique master code; up to eight high-speed mode masters can be present on the same I2C bus system. Valid values are from 0 to 7. This register goes away and becomes read-only returning 0's if the IC_MAX_SPEED_MODE configuration parameter is set to either Standard (1) or Fast (2). This register can be written only when the I2C interface is disabled, which corresponds to the IC_ENABLE[0] register being set to 0. Writes at other times have no effect. Reset value: IC_HS_MASTER_CODE configuration parameter Exists: Always

12.1.5.2.5 IC_DATA_CMD

- Name: I2C Rx/Tx Data Buffer and Command Register.
- Description: I2C Rx/Tx Data Buffer and Command Register; this is the register the CPU writes to when filling the TX FIFO and the CPU reads from when retrieving bytes from RX FIFO.

The size of the register changes as follows:

Write:

- 11 bits when IC_EMPTYFIFO_HOLD_MASTER_EN=1
- 9 bits when IC_EMPTYFIFO_HOLD_MASTER_EN=0

Read:

- 12 bits when IC_FIRST_DATA_BYTE_STATUS = 1
- bits when IC_FIRST_DATA_BYTE_STATUS = 0

Note: In order for the I2C controller to continue acknowledging reads, a read command should be written for every byte that is to be received; otherwise the I2C controller will stop acknowledging.

- Offset Address: 0x10

Table 12-1-6 Fields for Register: IC_DATA_CMD

Bits	Access	Name	Description & Reset Value
31:12	R	RSVD_IC_DATA_CMD	IC_DATA_CMD Reserved bits - Read Only Exists: Always Volatile: true

Bits	Access	Name	Description & Reset Value
11	R	FIRST_DATA_BYTE	Indicates the first data byte received after the address phase for receive transfer in Master receiver or Slave receiver mode. Reset value : 0x0 NOTE: In case of APB_DATA_WIDTH=8, 1. The user has to perform two APB Reads to IC_DATA_CMD in order to get status on 11 bit. 2. Inorder to read the 11 bit, the user has to perform the first data byte read [7:0] (offset 0x10) and then perform the second read[15:8](offset 0x11) in order to know the status of 11 bit (whether the data received in previous read is a first data byte or not). 3. The 11th bit is an optional read field, user can ignore 2nd byte read [15:8] (offset 0x11) if not interested in FIRST_DATA_BYTE status. Values: 0x1 (ACTIVE): Non sequential data byte received 0x0 (INACTIVE): Sequential data byte received Exists: IC_FIRST_DATA_BYTE_STATUS == 1 Volatile: true
10	W	RESTART	This bit controls whether a RESTART is issued before the byte is sent or received. This bit is available only if IC_EMPTYFIFO_HOLD_MASTER_EN is configured to 1. 1 - If IC_RESTART_EN is 1, a RESTART is issued before the data is sent/received (according to the value of CMD), regardless of whether or not the transfer direction is changing from the previous command; if IC_RESTART_EN is 0, a STOP followed by a START is issued instead. 0 - If IC_RESTART_EN is 1, a RESTART is issued only if the transfer direction is changing from the previous command; if IC_RESTART_EN is 0, a STOP followed by a START is issued instead. Reset value: 0x0 Values: 0x1 (ENABLE): Issue RESTART before this command 0x0 (DISABLE): Donot Issue RESTART before this command Exists: IC_EMPTYFIFO_HOLD_MASTER_EN Volatile: true

Bits	Access	Name	Description & Reset Value
9	W	STOP	This bit controls whether a STOP is issued after the byte is sent or received. This bit is available only if IC_EMPTYFIFO_HOLD_MASTER_EN is configured to 1. 1 - STOP is issued after this byte, regardless of whether or not the Tx FIFO is empty. If the Tx FIFO is not empty, the master immediately tries to start a new transfer by issuing a START and arbitrating for the bus. 0 - STOP is not issued after this byte, regardless of whether or not the Tx FIFO is empty. If the Tx FIFO is not empty, the master continues the current transfer by sending/receiving data bytes according to the value of the CMD bit. If the Tx FIFO is empty, the master holds the SCL line low and stalls the bus until a new command is available in the Tx FIFO. Reset value: 0x0 Values: 0x1 (ENABLE): Issue STOP after this command 0x0 (DISABLE): Donot Issue STOP after this command Exists: IC_EMPTYFIFO_HOLD_MASTER_EN Volatile: true
8	W	CMD	This bit controls whether a read or a write is performed. This bit does not control the direction when the DW_apb_i2c acts as a slave. It controls only the direction when it acts as a master. When a command is entered in the TX FIFO, this bit distinguishes the write and read commands. In slave- receiver mode, this bit is a don't care because writes to this register are not required. In slave-transmitter mode, a 0 indicates that the data in IC_DATA_CMD is to be transmitted. When programming this bit, you should remember the following: attempting to perform a read operation after a General Call command has been sent results in a TX_ABRT interrupt (bit 6 of the IC_RAW_INTR_STAT register), unless bit 11 (SPECIAL) in the IC_TAR register has been cleared. If a 1 is written to this bit after receiving a RD_REQ interrupt, then a TX_ABRT interrupt occurs. Reset value: 0x0 Values: 0x1 (READ): Master Read Command 0x0 (WRITE): Master Write Command Exists: Always Volatile: true
7:0	R/W	DAT	This register contains the data to be transmitted or received on the I2C bus. If you are writing to this register and want to perform a read, bits 7:0 (DAT) are ignored by the DW_apb_i2c. However, when you read this register, these bits return the value of data received on the DW_apb_i2c interface. Reset value: 0x0 Exists: Always Volatile: true

12.1.5.2.6 IC_SS_SCL_HCNT

- Name: Standard Speed I2C Clock SCL High Count Register.
- Description: Standard Speed I2C Clock SCL High Count Register
- Offset Address: 0x14

Table 12-1-7 Fields for Register: IC_SS_SCL_HCNT

Bits	Access	Name	Description & Reset Value
31:16	R	RSVD_IC_SS_SCL_HIGH_COUNT	IC_SS_SCL_HCNT Reserved bits - Read Only Exists: Always
15:0	*Varies	IC_SS_SCL_HCNT	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock high-period count for standard speed. For more information, refer to IC_CLK Frequency Configuration . This register can be written only when the I2C interface is disabled which corresponds to the IC_ENABLE[0] register being set to 0. Writes at other times have no effect. The minimum valid value is 6; hardware prevents values less than this being written, and if attempted results in 6 being set. For designs with APB_DATA_WIDTH = 8, the order of programming is important to ensure the correct operation of the DW_apb_i2c. The lower byte must be programmed first. Then the upper byte is programmed. When the configuration parameter IC_HC_COUNT_VALUES is set to 1, this register is read only. NOTE: This register must not be programmed to a value higher than 65525, because DW_apb_i2c uses a 16-bit counter to flag an I2C bus idle condition when this counter reaches a value of IC_SS_SCL_HCNT + 10. Reset value: IC_SS_SCL_HIGH_COUNT configuration parameter Exists: Always Memory Access: (IC_HC_COUNT_VALUES==1) ? read-only : read-write

12.1.5.2.7 IC_UFM_SCL_HCNT

- Name: Ultra-Fast Speed I2C Clock SCL High Count Register.
- Description: Ultra-Fast Speed I2C Clock SCL High Count Register
- Offset Address: 0x14

Table 12-1-8 Fields for Register: IC_UFM_SCL_HCNT

Bits	Access	Name	Description & Reset Value
31:16	R	RSVD_IC_UFM_SCL_HCNT	IC_UFM_SCL_HCNT Reserved bits - Read Only Exists: Always

Bits	Access	Name	Description & Reset Value
15:0	*Varies	IC_UFM_SCL_HCNT	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock high-period count for Ultra-Fast speed. For more information, refer to IC_CLK Frequency Configuration . This register can be written only when the I2C interface is disabled which corresponds to the IC_ENABLE[0] register being set to 0. Writes at other times have no effect. The minimum valid value is 3; hardware prevents values less than this being written, and if attempted results in 3 being set. For designs with APB_DATA_WIDTH = 8, the order of programming is important to ensure the correct operation of the DW_apb_i2c. The lower byte must be programmed first. Then the upper byte is programmed. When the configuration parameter IC_HC_COUNT_VALUES is set to 1, this register is read only. Reset value: IC_UFM_SCL_HIGH_COUNT configuration parameter Exists: Always Memory Access: (IC_HC_COUNT_VALUES==1) ? read- only : read-write

12.1.5.2.8 IC_SS_SCL_LCNT

- Name: Standard Speed I2C Clock SCL Low Count Register.
- Description: Standard Speed I2C Clock SCL Low Count Register
- Offset Address: 0x18

Table 12-1-9 Fields for Register: IC_SS_SCL_LCNT

Bits	Access	Name	Description & Reset Value
31:16	R	RSVD_IC_SS_SCL_LOW_COUNT	RSVD_IC_SS_SCL_LOW_COUNT Reserved bits - Read Only Exists: Always

Bits	Access	Name	Description & Reset Value
15:0	*Varies	IC_SS_SCL_LCNT	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock low period count for standard speed. For more information, refer to IC_CLK Frequency Configuration This register can be written only when the I2C interface is disabled which corresponds to the IC_ENABLE[0] register being set to 0. Writes at other times have no effect. The minimum valid value is 8; hardware prevents values less than this being written, and if attempted, results in 8 being set. For designs with APB_DATA_WIDTH = 8, the order of programming is important to ensure the correct operation of DW_apb_i2c. The lower byte must be programmed first, and then the upper byte is programmed. When the configuration parameter IC_HC_COUNT_VALUES is set to 1, this register is read only. Reset value: IC_SS_SCL_LOW_COUNT configuration parameter Exists: Always Memory Access: (IC_HC_COUNT_VALUES==1) ? read-only : read-write

12.1.5.2.9 IC_UFM_SCL_LCNT

- Name: Ultra-Fast Speed I2C Clock SCL Low Count Register.
- Description: Ultra-Fast Speed I2C Clock SCL Low Count Register
- Offset Address: 0x18

Table 12-1-10 Fields for Register: IC_UFM_SCL_LCNT

Bits	Access	Name	Description & Reset Value
31:16	R	RSVD_IC_UFM_SCL_LCNT	IC_UFM_SCL_LCNT Reserved bits - Read Only Exists: Always

Bits	Access	Name	Description & Reset Value
15:0	*Varies	IC_UFM_SCL_LCNT	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock low period count for Ultra-Fast speed. This register can be written only when the I2C interface is disabled which corresponds to the IC_ENABLE[0] register being set to 0. Writes at other times have no effect. The minimum valid value is 5; hardware prevents values less than this being written, and if attempted, results in 5 being set. For designs with APB_DATA_WIDTH = 8, the order of programming is important to ensure the correct operation of DW_apb_i2c. The lower byte must be programmed first, and then the upper byte is programmed. When the configuration parameter IC_HC_COUNT_VALUES is set to 1, this register is read only. Reset value: IC_UFM_SCL_LOW_COUNT configuration parameter Exists: Always Memory Access: (IC_HC_COUNT_VALUES==1) ? read- only : read-write

12.1.5.2.10 IC_FS_SCL_HCNT

- Name: Fast Mode or Fast Mode Plus I2C Clock SCL High Count Register.
- Description: Fast Mode or Fast Mode Plus I2C Clock SCL High Count Register
- Offset Address: 0x1c

Table 12-1-11 Fields for Register: IC_FS_SCL_HCNT

Bits	Access	Name	Description & Reset Value
31:16	R	RSVD_IC_FS_SCL_HCNT	IC_FS_SCL_HCNT Reserved bits - Read Only Exists: Always

Bits	Access	Name	Description & Reset Value
15:0	*Varies	IC_FS_SCL_HCNT	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock high-period count for fast mode or fast mode plus. It is used in high-speed mode to send the Master Code and START BYTE or General CALL. For more information, refer to IC_CLK Frequency Configuration . This register goes away and becomes read-only returning 0s if IC_MAX_SPEED_MODE = standard. This register can be written only when the I2C interface is disabled, which corresponds to the IC_ENABLE[0] register being set to 0. Writes at other times have no effect. The minimum valid value is 6; hardware prevents values less than this being written, and if attempted results in 6 being set. For designs with APB_DATA_WIDTH == 8 the order of programming is important to ensure the correct operation of the DW_apb_i2c. The lower byte must be programmed first. Then the upper byte is programmed. Reset value: IC_FS_SCL_HIGH_COUNT configuration parameter Exists: Always Memory Access: (IC_HC_COUNT_VALUES==1) ? read- only : read-write

12.1.5.2.11 IC_UFM_TBUF_CNT

- Name: Ultra-Fast Speed mode TBuf Idle Count Register.
- Description: Ultra-Fast Speed mode TBuf Idle Count Register
- Offset Address: 0x1c

Table 12-1-12 Fields for Register: IC_UFM_TBUF_CNT

Bits	Access	Name	Description & Reset Value
31:16	R	RSVD_IC_UFM_TBUF_CNT	IC_UFM_TBUF_CNT Reserved bits - Read Only Exists: Always

Bits	Access	Name	Description & Reset Value
15:0	R/W	IC_UFM_TBUF_CNT	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the Bus-Free time between a STOP and STOP condition count for Ultra-Fast speed. This register can be written only when the I2C interface is disabled which corresponds to the IC_ENABLE[0] register being set to 0. Writes at other times have no effect. For designs with APB_DATA_WIDTH = 8, the order of programming is important to ensure the correct operation of the DW_apb_i2c. The lower byte must be programmed first and then the upper byte is programmed. When the configuration parameter. NOTE: The DW_apb_i2c will add 9 ic_clks after tBuf time is expired to generate START on the Bus. Reset value: IC_UFM_TBUF_CNT_DEFAULT configuration parameter Exists: Always

12.1.5.2.12 IC_FS_SCL_LCNT

- Name: Fast Mode or Fast Mode Plus I2C Clock SCL Low Count Register.
- Description: Fast Mode or Fast Mode Plus I2C Clock SCL Low Count Register
- Offset Address: 0x20

Table 12-1-13 Fields for Register: IC_FS_SCL_LCNT

Bits	Access	Name	Description & Reset Value
31:16	R	RSVD_IC_FS_SCL_LCNT	IC_FS_SCL_LCNT Reserved bits - Read Only Exists: Always

Bits	Access	Name	Description & Reset Value
15:0	*Varies	IC_FS_SCL_LCNT	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock low period count for fast speed. It is used in high- speed mode to send the Master Code and START BYTE or General CALL. For more information, refer to IC_CLK Frequency Configuration . This register goes away and becomes read-only returning 0s if IC_MAX_SPEED_MODE = standard. This register can be written only when the I2C interface is disabled, which corresponds to the IC_ENABLE[0] register being set to 0. Writes at other times have no effect. The minimum valid value is 8; hardware prevents values less than this being written, and if attempted results in 8 being set. For designs with APB_DATA_WIDTH = 8 the order of programming is important to ensure the correct operation of the DW_apb_i2c. The lower byte must be programmed first. Then the upper byte is programmed. If the value is less than 8 then the count value gets changed to 8. When the configuration parameter IC_HC_COUNT_VALUES is set to 1, this register is read only. Reset value: IC_FS_SCL_LOW_COUNT configuration parameter Exists: Always Memory Access: (IC_HC_COUNT_VALUES==1) ? read- only : read-write

12.1.5.2.13 IC_HS_SCL_HCNT

- Name: High Speed I2C Clock SCL High Count Register.
- Description: High Speed I2C Clock SCL High Count Register
- Offset Address: 0x24

Table 12-1-14 Fields for Register: IC_HS_SCL_HCNT

Bits	Access	Name	Description & Reset Value
31:16	R	RSVD_IC_HS_SCL_HCNT	IC_HS_SCL_HCNT Reserved bits - Read Only Exists: Always

Bits	Access	Name	Description & Reset Value
15:0	*Varies	IC_HS_SCL_HCNT	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock high period count for high speed.refer to IC_CLK Frequency Configuration . The SCL High time depends on the loading of the bus. For 100pF loading, the SCL High time is 60ns; for 400pF loading, the SCL High time is 120ns. This register goes away and becomes read-only returning 0s if IC_MAX_SPEED_MODE != high. This register can be written only when the I2C interface is disabled, which corresponds to the IC_ENABLE[0] register being set to 0. Writes at other times have no effect. The minimum valid value is 6; hardware prevents values less than this being written, and if attempted results in 6 being set. For designs with APB_DATA_WIDTH = 8 the order of programming is important to ensure the correct operation of the DW_apb_i2c. The lower byte must be programmed first. Then the upper byte is programmed. Reset value: IC_HS_SCL_HIGH_COUNT configuration parameter Exists: Always Memory Access: (IC_HC_COUNT_VALUES==1) ? read- only : read-write

12.1.5.2.14 IC_HS_SCL_LCNT

- Name: High Speed I2C Clock SCL Low Count Register.
- Description: High Speed I2C Clock SCL Low Count Register
- Offset Address: 0x28

Table 12-1-15 Fields for Register: IC_HS_SCL_LCNT

Bits	Access	Name	Description & Reset Value
31:16	R	RSVD_IC_HS_SCL_LOW_CNT	IC_HS_SCL_LCNT Reserved bits - Read Only Exists: Always

Bits	Access	Name	Description & Reset Value
15:0	*Varies	IC_HS_SCL_LCNT	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock low period count for high speed. For more information, refer to IC_CLK Frequency Configuration . The SCL low time depends on the loading of the bus. For 100pF loading, the SCL low time is 160ns; for 400pF loading, the SCL low time is 320ns. This register goes away and becomes read-only returning 0s if IC_MAX_SPEED_MODE != high. This register can be written only when the I2C interface is disabled, which corresponds to the IC_ENABLE[0] register being set to 0. Writes at other times have no effect. The minimum valid value is 8; hardware prevents values less than this being written, and if attempted results in 8 being set. For designs with APB_DATA_WIDTH == 8 the order of programming is important to ensure the correct operation of the DW_apb_i2c. The lower byte must be programmed first. Then the upper byte is programmed. If the value is less than 8 then the count value gets changed to 8. Reset value: IC_HS_SCL_LOW_COUNT configuration parameter Exists: Always Memory Access: (IC_HC_COUNT_VALUES==1) ? read-only : read-write

12.1.5.2.15 IC_INTR_STAT

- Name: I2C Interrupt Status Register.
- Description: I2C Interrupt Status Register Each bit in this register has a corresponding mask bit in the IC_INTR_MASK register. These bits are cleared by reading the matching interrupt clear register. The unmasked raw versions of these bits are available in the IC_RAW_INTR_STAT register.
- Offset Address: 0x2c

Table 12-1-16 Fields for Register: IC_INTR_STAT

Bits	Access	Name	Description & Reset Value
31:15	R	RSVD_IC_INTR_STAT	IC_INTR_STAT Reserved bits - Read Only Exists: Always Volatile: true
14	R	R_SCL_STUCK_AT_LOW	See IC_RAW_INTR_STAT for a detailed description of R_SCL_STUCK_AT_LOW bit. Reset Value: 0x0 Values: 0x1 (ACTIVE): R_SCL_STUCK_AT_LOW interrupt is active 0x0 (INACTIVE): R_SCL_STUCK_AT_LOW interrupt is inactive Exists: IC_BUS_CLEAR_FEATURE==1 Volatile: true

Bits	Access	Name	Description & Reset Value
13	R	R_MASTER_ON_HOLD	See IC_RAW_INTR_STAT for a detailed description of R_MASTER_ON_HOLD bit. Reset value: 0x0 Values: 0x1 (ACTIVE): R_MASTER_ON_HOLD interrupt is active 0x0 (INACTIVE): R_MASTER_ON_HOLD interrupt is inactive Exists: Always Volatile: true
12	R	R_RESTART_DET	See IC_RAW_INTR_STAT for a detailed description of R_RESTART_DET bit. Reset value: 0x0 Values: 0x1 (ACTIVE): R_RESTART_DET interrupt is active 0x0 (INACTIVE): R_RESTART_DET interrupt is inactive Exists: Always Volatile: true
11	R	R_GEN_CALL	See IC_RAW_INTR_STAT for a detailed description of R_GEN_CALL bit. Reset value: 0x0 Values: 0x1 (ACTIVE): R_GEN_CALL interrupt is active 0x0 (INACTIVE): R_GEN_CALL interrupt is inactive Exists: Always Volatile: true
10	R	R_START_DET	See IC_RAW_INTR_STAT for a detailed description of R_START_DET bit. Reset value: 0x0 Values: 0x1 (ACTIVE): R_START_DET interrupt is active 0x0 (INACTIVE): R_START_DET interrupt is inactive Exists: Always Volatile: true
9	R	R_STOP_DET	See IC_RAW_INTR_STAT for a detailed description of R_STOP_DET bit. Reset value: 0x0 Values: 0x1 (ACTIVE): R_STOP_DET interrupt is active 0x0 (INACTIVE): R_STOP_DET interrupt is inactive Exists: Always Volatile: true

Bits	Access	Name	Description & Reset Value
8	R	R_ACTIVITY	See IC_RAW_INTR_STAT for a detailed description of R_ACTIVITY bit. Reset value: 0x0 Values: 0x1 (ACTIVE): R_ACTIVITY interrupt is active 0x0 (INACTIVE): R_ACTIVITY interrupt is inactive Exists: Always Volatile: true
7	R	R_RX_DONE	See IC_RAW_INTR_STAT for a detailed description of R_RX_DONE bit. Reset value: 0x0 Values: 0x1 (ACTIVE): R_RX_DONE interrupt is active 0x0 (INACTIVE): R_RX_DONE interrupt is inactive Exists: IC_ULTRA_FAST_MODE==0 Volatile: true
6	R	R_TX_ABRT	See IC_RAW_INTR_STAT for a detailed description of R_TX_ABRT bit. Reset value: 0x0 Values: 0x1 (ACTIVE): R_TX_ABRT interrupt is active 0x0 (INACTIVE): R_TX_ABRT interrupt is inactive Exists: Always Volatile: true
5	R	R_RD_REQ	See IC_RAW_INTR_STAT for a detailed description of R_RD_REQ bit. Reset value: 0x0 Values: 0x1 (ACTIVE): R_RD_REQ interrupt is active 0x0 (INACTIVE): R_RD_REQ interrupt is inactive Exists: IC_ULTRA_FAST_MODE==0 Volatile: true
4	R	R_TX_EMPTY	See IC_RAW_INTR_STAT for a detailed description of R_TX_EMPTY bit. Reset value: 0x0 Values: 0x1 (ACTIVE): R_TX_EMPTY interrupt is active 0x0 (INACTIVE): R_TX_EMPTY interrupt is inactive Exists: Always Volatile: true

Bits	Access	Name	Description & Reset Value
3	R	R_TX_OVER	See IC_RAW_INTR_STAT for a detailed description of R_TX_OVER bit. Reset value: 0x0 Values: 0x1 (ACTIVE): R_TX_OVER interrupt is active 0x0 (INACTIVE): R_TX_OVER interrupt is inactive Exists: Always Volatile: true
2	R	R_RX_FULL	See IC_RAW_INTR_STAT for a detailed description of R_RX_FULL bit. Reset value: 0x0 Values: 0x1 (ACTIVE): R_RX_FULL interrupt is active 0x0 (INACTIVE): R_RX_FULL interrupt is inactive Exists: Always Volatile: true
1	R	R_RX_OVER	See IC_RAW_INTR_STAT for a detailed description of R_RX_OVER bit. Reset value: 0x0 Values: 0x1 (ACTIVE): R_RX_OVER interrupt is active 0x0 (INACTIVE): R_RX_OVER interrupt is inactive Exists: Always Volatile: true
0	R	R_RX_UNDER	See IC_RAW_INTR_STAT for a detailed description of R_RX_UNDER bit. Reset value: 0x0 Values: 0x1 (ACTIVE): RX_UNDER interrupt is active 0x0 (INACTIVE): RX_UNDER interrupt is inactive Exists: Always Volatile: true

12.1.5.2.16 IC_INTR_MASK

- Name: I2C Interrupt Mask Register.
- Description: I2C Interrupt Mask Register. These bits mask their corresponding interrupt status bits. This register is active low; a value of 0 masks the interrupt, whereas a value of 1 unmasks the interrupt.
- Offset Address: 0x30

Table 12-1-17 Fields for Register: IC_INTR_MASK

Bits	Access	Name	Description & Reset Value
31:15	R	RSVD_IC_INTR_STAT	IC_INTR_STAT Reserved bits - Read Only Exists: Always

Bits	Access	Name	Description & Reset Value
14	R/W	M_SCL_STUCK_AT_LOW	This bit masks the R_SCL_STUCK_AT_LOW interrupt in IC_INTR_STAT register. Reset Value: 0x0 Values: 0x1 (DISABLED): SCL_STUCK_AT_LOW interrupt is unmasked 0x0 (ENABLED): SCL_STUCK_AT_LOW interrupt is masked Exists: IC_BUS_CLEAR_FEATURE==1
13	R/W	M_MASTER_ON_HOLD	This bit masks the R_MASTER_ON_HOLD interrupt in IC_INTR_STAT register. Reset value: 0x0 Values: 0x1 (DISABLED): MASTER_ON_HOLD interrupt is unmasked 0x0 (ENABLED): MASTER_ON_HOLD interrupt is masked Exists: I2C_DYNAMIC_TAR_UPDATE == 1 && IC_EMPTYFIFO_HOLD_MASTER_EN == 1
12	R/W	M_RESTART_DET	This bit masks the R_RESTART_DET interrupt in IC_INTR_STAT register. Reset value: 0x0 Values: 0x1 (DISABLED): RESTART_DET interrupt is unmasked 0x0 (ENABLED): RESTART_DET interrupt is masked Exists: IC_SLV_RESTART_DET_EN == 1
11	R/W	M_GEN_CALL	This bit masks the R_GEN_CALL interrupt in IC_INTR_STAT register. Reset value: 0x1 Values: 0x1 (DISABLED): GEN_CALL interrupt is unmasked 0x0 (ENABLED): GEN_CALL interrupt is masked Exists: Always
10	R/W	M_START_DET	This bit masks the R_START_DET interrupt in IC_INTR_STAT register. Reset value: 0x0 Values: 0x1 (DISABLED): START_DET interrupt is unmasked 0x0 (ENABLED): START_DET interrupt is masked Exists: Always
9	R/W	M_STOP_DET	This bit masks the R_STOP_DET interrupt in IC_INTR_STAT register. Reset value: 0x0 Values: 0x1 (DISABLED): STOP_DET interrupt is unmasked 0x0 (ENABLED): STOP_DET interrupt is masked Exists: Always

Bits	Access	Name	Description & Reset Value
8	R/W	M_ACTIVITY	This bit masks the R_ACTIVITY interrupt in IC_INTR_STAT register. Reset value: 0x0 Values: 0x1 (DISABLED): ACTIVITY interrupt is unmasked 0x0 (ENABLED): ACTIVITY interrupt is masked Exists: Always
7	R/W	M_RX_DONE	This bit masks the R_RX_DONE interrupt in IC_INTR_STAT register. Reset value: 0x1 Values: 0x1 (DISABLED): RX_DONE interrupt is unmasked 0x0 (ENABLED): RX_DONE interrupt is masked Exists: IC_ULTRA_FAST_MODE==0
6	R/W	M_TX_ABRT	This bit masks the R_TX_ABRT interrupt in IC_INTR_STAT register. Reset value: 0x1 Values: 0x1 (DISABLED): TX_ABORT interrupt is unmasked 0x0 (ENABLED): TX_ABORT interrupt is masked Exists: Always
5	R/W	M_RD_REQ	This bit masks the R_RD_REQ interrupt in IC_INTR_STAT register. Reset value: 0x1 Values: 0x1 (DISABLED): RD_REQ interrupt is unmasked 0x0 (ENABLED): RD_REQ interrupt is masked Exists: IC_ULTRA_FAST_MODE==0
4	R/W	M_TX_EMPTY	This bit masks the R_TX_EMPTY interrupt in IC_INTR_STAT register. Reset value: 0x1 Values: 0x1 (DISABLED): TX_EMPTY interrupt is unmasked 0x0 (ENABLED): TX_EMPTY interrupt is masked Exists: Always
3	R/W	M_TX_OVER	This bit masks the R_TX_OVER interrupt in IC_INTR_STAT register. Reset value: 0x1 Values: 0x1 (DISABLED): TX_OVER interrupt is unmasked 0x0 (ENABLED): TX_OVER interrupt is masked Exists: Always

Bits	Access	Name	Description & Reset Value
2	R/W	M_RX_FULL	This bit masks the R_RX_FULL interrupt in IC_INTR_STAT register. Reset value: 0x1 Values: 0x1 (DISABLED): RX_FULL interrupt is unmasked 0x0 (ENABLED): RX_FULL interrupt is masked Exists: Always
1	R/W	M_RX_OVER	This bit masks the R_RX_OVER interrupt in IC_INTR_STAT register. Reset value: 0x1 Values: 0x1 (DISABLED): RX_OVER interrupt is unmasked 0x0 (ENABLED): RX_OVER interrupt is masked Exists: Always
0	R/W	M_RX_UNDER	This bit masks the R_RX_UNDER interrupt in IC_INTR_STAT register. Reset value: 0x1 Values: 0x1 (DISABLED): RX_UNDER interrupt is unmasked 0x0 (ENABLED): RX_UNDER interrupt is masked Exists: Always

12.1.5.2.17 IC_RAW_INTR_STAT

- Name: I2C Raw Interrupt Status Register.
- Description: I2C Raw Interrupt Status Register Unlike the IC_INTR_STAT register, these bits are not masked so they always show the true status of the I2C controller.
- Offset Address: 0x34

Table 12-1-18 Fields for Register: IC_RAW_INTR_STAT

Bits	Access	Name	Description & Reset Value
31:15	R	RSVD_IC_RAW_INTR_STAT	IC_RAW_INTR_STAT Reserved bits - Read Only Exists: Always Volatile: true
14	R	SCL_STUCK_AT_LOW	Indicates whether the SCL Line is stuck at low for the IC_SCL_STUCK_LOW_TIMEOUT number of ic_clk periods. Enabled only when IC_BUS_CLEAR_FEATURE=1 and IC_ULTRA_FAST_MODE=0. Reset Value: 0x0 Values: 0x1 (ACTIVE): SCL_STUCK_AT_LOW interrupt is active 0x0 (INACTIVE): SCL_STUCK_AT_LOW interrupt is inactive. Exists: IC_BUS_CLEAR_FEATURE==1 Volatile: true

Bits	Access	Name	Description & Reset Value
13	R	MASTER_ON_HOLD	Indicates whether master is holding the bus and TX FIFO is empty. Enabled only when I2C_DYNAMIC_TAR_UPDATE=1 and IC_EMPTYFIFO_HOLD_MASTER_EN=1. Reset value: 0x0 Values: 0x1 (ACTIVE): MASTER_ON_HOLD interrupt is active 0x0 (INACTIVE): MASTER_ON_HOLD interrupt is inactive Exists: Always Volatile: true
12	R	RESTART_DET	Indicates whether a RESTART condition has occurred on the I2C interface when DW_apb_i2c is operating in Slave mode and the slave is being addressed. Enabled only when IC_SLV_RESTART_DET_EN=1. Note: However, in high-speed mode or during a START BYTE transfer, the RESTART comes before the address field as per the I2C protocol. In this case, the slave is not the addressed slave when the RESTART is issued, therefore DW_apb_i2c does not generate the RESTART_DET interrupt. Reset value: 0x0 Values: 0x1 (ACTIVE): RESTART_DET interrupt is active 0x0 (INACTIVE): RESTART_DET interrupt is inactive Exists: Always Volatile: true
11	R	GEN_CALL	Set only when a General Call address is received and it is acknowledged. It stays set until it is cleared either by disabling DW_apb_i2c or when the CPU reads bit 0 of the IC_CLR_GEN_CALL register. DW_apb_i2c stores the received data in the Rx buffer. Reset value: 0x0 Values: 0x1 (ACTIVE): GEN_CALL interrupt is active 0x0 (INACTIVE): GEN_CALL interrupt is inactive Exists: Always Volatile: true
10	R	START_DET	Indicates whether a START or RESTART condition has occurred on the I2C interface regardless of whether DW_apb_i2c is operating in slave or master mode. Reset value: 0x0 Values: 0x1 (ACTIVE): START_DET interrupt is active 0x0 (INACTIVE): START_DET interrupt is inactive Exists: Always Volatile: true

Bits	Access	Name	Description & Reset Value
9	R	STOP_DET	Indicates whether a STOP condition has occurred on the I2C interface regardless of whether DW_apb_i2c is operating in slave or master mode. In Slave Mode: If IC_CON[7]=1'b1 (STOP_DET_IFADDRESSED), the STOP_DET interrupt will be issued only if slave is addressed. Note: During a general call address, this slave does not issue a STOP_DET interrupt if STOP_DET_IF_ADDRESSED=1'b1, even if the slave responds to the general call address by generating ACK. The STOP_DET interrupt is generated only when the transmitted address matches the slave address (SAR). If IC_CON[7]=1'b0 (STOP_DET_IFADDRESSED), the STOP_DET interrupt is issued irrespective of whether it is being addressed. In Master Mode: If IC_CON[10]=1'b1 (STOP_DET_IF_MASTER_ACTIVE), the STOP_DET interrupt will be issued only if Master is active. If IC_CON[10]=1'b0 (STOP_DET_IFADDRESSED), the STOP_DET interrupt will be issued irrespective of whether master is active or not. Reset value: 0x0 Values: 0x1 (ACTIVE): STOP_DET interrupt is active 0x0 (INACTIVE): STOP_DET interrupt is inactive Exists: Always Volatile: true
8	R	ACTIVITY	This bit captures DW_apb_i2c activity and stays set until it is cleared. There are four ways to clear it: Disabling the DW_apb_i2c Reading the IC_CLR_ACTIVITY register Reading the IC_CLR_INTR register System reset Once this bit is set, it stays set unless one of the four methods is used to clear it. Even if the DW_apb_i2c module is idle, this bit remains set until cleared, indicating that there was activity on the bus. Reset value: 0x0 Values: 0x1 (ACTIVE): RAW_INTR_ACTIVITY interrupt is active 0x0 (INACTIVE): RAW_INTR_ACTIVITY interrupt is inactive Exists: Always Volatile: true

Bits	Access	Name	Description & Reset Value
7	R	RX_DONE	When the DW_apb_i2c is acting as a slave-transmitter, this bit is set to 1 if the master does not acknowledge a transmitted byte. This occurs on the last byte of the transmission, indicating that the transmission is done. Reset value: 0x0 Values: 0x1 (ACTIVE): RX_DONE interrupt is active 0x0 (INACTIVE): RX_DONE interrupt is inactive Exists: IC_ULTRA_FAST_MODE==0 Volatile: true
6	R	TX_ABRT	This bit indicates if DW_apb_i2c, as an I2C transmitter, is unable to complete the intended actions on the contents of the transmit FIFO. This situation can occur both as an I2C master or an I2C slave, and is referred to as a 'transmit abort'. When this bit is set to 1, the IC_TX_ABRT_SOURCE register indicates the reason why the transmit abort takes places. Note: The DW_apb_i2c flushes/resets/empties only the TX_FIFO whenever there is a transmit abort caused by any of the events tracked by the IC_TX_ABRT_SOURCE register. The Tx FIFO remains in this flushed state until the register IC_CLR_TX_ABRT is read. Once this read is performed, the Tx FIFO is then ready to accept more data bytes from the APB interface. RX FIFO flush because of TX_ABRT is controlled by the coreConsultant parameter IC_AVOID_RX_FIFO_FLUSH_ON_TX_ABRT. Reset value: 0x0 Values: 0x1 (ACTIVE): TX_ABRT interrupt is active 0x0 (INACTIVE): TX_ABRT interrupt is inactive Exists: Always Volatile: true
5	R	RD_REQ	This bit is set to 1 when DW_apb_i2c is acting as a slave and another I2C master is attempting to read data from DW_apb_i2c. The DW_apb_i2c holds the I2C bus in a wait state (SCL=0) until this interrupt is serviced, which means that the slave has been addressed by a remote master that is asking for data to be transferred. The processor must respond to this interrupt and then write the requested data to the IC_DATA_CMD register. This bit is set to 0 just after the processor reads the IC_CLR_RD_REQ register. Reset value: 0x0 Values: 0x1 (ACTIVE): RD_REQ interrupt is active 0x0 (INACTIVE): RD_REQ interrupt is inactive Exists: IC_ULTRA_FAST_MODE==0 Volatile: true

Bits	Access	Name	Description & Reset Value
4	R	TX_EMPTY	The behavior of the TX_EMPTY interrupt status differs based on the TX_EMPTY_CTRL selection in the IC_CON register. When TX_EMPTY_CTRL = 0: This bit is set to 1 when the transmit buffer is at or below the threshold value set in the IC_TX_TL register. When TX_EMPTY_CTRL = 1: This bit is set to 1 when the transmit buffer is at or below the threshold value set in the IC_TX_TL register and the transmission of the address/data from the internal shift register for the most recently popped command is completed. It is automatically cleared by hardware when the buffer level goes above the threshold. When IC_ENABLE[0] is set to 0, the TX FIFO is flushed and held in reset. There the TX FIFO looks like it has no data within it, so this bit is set to 1, provided there is activity in the master or slave state machines. When there is no longer any activity, then with ic_en=0, this bit is set to 0. Reset value: 0x0. Values: 0x1 (ACTIVE): TX_EMPTY interrupt is active 0x0 (INACTIVE): TX_EMPTY interrupt is inactive Exists: Always Volatile: true
3	R	TX_OVER	Set during transmit if the transmit buffer is filled to IC_TX_BUFFER_DEPTH and the processor attempts to issue another I2C command by writing to the IC_DATA_CMD register. When the module is disabled, this bit keeps its level until the master or slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared. Reset value: 0x0 Values: 0x1 (ACTIVE): TX_OVER interrupt is active 0x0 (INACTIVE): TX_OVER interrupt is inactive Exists: Always Volatile: true
2	R	RX_FULL	Set when the receive buffer reaches or goes above the RX_TL threshold in the IC_RX_TL register. It is automatically cleared by hardware when buffer level goes below the threshold. If the module is disabled (IC_ENABLE[0]=0), the RX FIFO is flushed and held in reset; therefore the RX FIFO is not full. So this bit is cleared once the IC_ENABLE bit 0 is programmed with a 0, regardless of the activity that continues. Reset value: 0x0 Values: 0x1 (ACTIVE): RX_FULL interrupt is active 0x0 (INACTIVE): RX_FULL interrupt is inactive Exists: Always Volatile: true

Bits	Access	Name	Description & Reset Value
1	R	RX_OVER	Set if the receive buffer is completely filled to IC_RX_BUFFER_DEPTH and an additional byte is received from an external I2C device. The DW_apb_i2c acknowledges this, but any data bytes received after the FIFO is full are lost. If the module is disabled (IC_ENABLE[0]=0), this bit keeps its level until the master or slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared. Note: If the configuration parameter IC_RX_FULL_HLD_BUS_EN is enabled and bit 9 of the IC_CON register (RX_FIFO_FULL_HLD_CTRL) is programmed to HIGH, then the RX_OVER interrupt never occurs, because the Rx FIFO never overflows. Reset value: 0x0 Values: 0x1 (ACTIVE): RX_OVER interrupt is active 0x0 (INACTIVE): RX_OVER interrupt is inactive Exists: Always Volatile: true
0	R	RX_UNDER	Set if the processor attempts to read the receive buffer when it is empty by reading from the IC_DATA_CMD register. If the module is disabled (IC_ENABLE[0]=0), this bit keeps its level until the master or slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared. Reset value: 0x0 Values: 0x1 (ACTIVE): RX_UNDER interrupt is active 0x0 (INACTIVE): RX_UNDER interrupt is inactive Exists: Always Volatile: true

12.1.5.2.18 IC_RX_TL

- Name: I2C Receive FIFO Threshold Register.
- Description: I2C Receive FIFO Threshold Register
- Offset Address: 0x38

Table 12-1-19 Fields for Register: IC_RX_TL

Bits	Access	Name	Description & Reset Value
31:8	R	RSVD_IC_RX_TL	IC_RX_TL Reserved bits - Read Only Exists: Always

Bits	Access	Name	Description & Reset Value
7:0	R/W	RX_TL	Receive FIFO Threshold Level. Controls the level of entries (or above) that triggers the RX_FULL interrupt (bit 2 in IC_RAW_INTR_STAT register). The valid range is 0-255, with the additional restriction that hardware does not allow this value to be set to a value larger than the depth of the buffer. If an attempt is made to do that, the actual value set will be the maximum depth of the buffer. A value of 0 sets the threshold for 1 entry, and a value of 255 sets the threshold for 256 entries. Reset value: IC_RX_TL configuration parameter Exists: Always

12.1.5.2.19 IC_TX_TL

- Name: I2C Transmit FIFO Threshold Register.
- Description: I2C Transmit FIFO Threshold Register
- Offset Address: 0x3c

Table 12-1-20 Fields for Register: IC_TX_TL

Bits	Access	Name	Description & Reset Value
31:8	R	RSVD_IC_TX_TL	IC_TX_TL Reserved bits - Read Only Exists: Always
7:0	R/W	TX_TL	Transmit FIFO Threshold Level. Controls the level of entries (or below) that trigger the TX_EMPTY interrupt (bit 4 in IC_RAW_INTR_STAT register). The valid range is 0-255, with the additional restriction that it may not be set to value larger than the depth of the buffer. If an attempt is made to do that, the actual value set will be the maximum depth of the buffer. A value of 0 sets the threshold for 0 entries, and a value of 255 sets the threshold for 255 entries. Reset value: IC_TX_TL configuration parameter Exists: Always

12.1.5.2.20 IC_CLR_INTR

- Name: Clear Combined and Individual Interrupt Register.
- Description: Clear Combined and Individual Interrupt Register
- Offset Address: 0x40

Table 12-1-21 Fields for Register: IC_CLR_INTR

Bits	Access	Name	Description & Reset Value
31:1	R	RSVD_IC_CLR_INTR	CLR_INTR Reserved bits - Read Only Exists: Always Volatile: true

Bits	Access	Name	Description & Reset Value
0	R	CLR_INTR	Read this register to clear the combined interrupt, all individual interrupts, and the IC_TX_ABRT_SOURCE register. This bit does not clear hardware clearable interrupts but software clearable interrupts. Refer to Bit 9 of the IC_TX_ABRT_SOURCE register for an exception to clearing IC_TX_ABRT_SOURCE. Reset value: 0x0 Exists: Always Volatile: true

12.1.5.2.21 IC_CLR_RX_UNDER

- Name: Clear RX_UNDER Interrupt Register.
- Description: Clear RX_UNDER Interrupt Register
- Offset Address: 0x44

Table 12-1-22 Fields for Register: IC_CLR_RX_UNDER

Bits	Access	Name	Description & Reset Value
31:1	R	RSVD_IC_CLR_RX_UNDER	IC_CLR_RX_UNDER Reserved bits - Read Only Exists: Always Volatile: true
0	R	CLR_RX_UNDER	Read this register to clear the RX_UNDER interrupt (bit 0) of the IC_RAW_INTR_STAT register. Reset value: 0x0 Exists: Always Volatile: true

12.1.5.2.22 IC_CLR_RX_OVER

- Name: Clear RX_OVER Interrupt Register.
- Description: Clear RX_OVER Interrupt Register
- Offset Address: 0x48

Table 12-1-23 Fields for Register: IC_CLR_RX_OVER

Bits	Access	Name	Description & Reset Value
31:1	R	RSVD_IC_CLR_RX_OVER	IC_CLR_RX_OVER Reserved bits - Read Only Exists: Always Volatile: true
0	R	CLR_RX_OVER	Read this register to clear the RX_OVER interrupt (bit 1) of the IC_RAW_INTR_STAT register. Reset value: 0x0 Exists: Always Volatile: true

12.1.5.2.23 IC_CLR_TX_OVER

- Name: Clear TX_OVER Interrupt Register.
- Description: Clear TX_OVER Interrupt Register
- Offset Address: 0x4c

Table 12-1-24 Fields for Register: IC_CLR_TX_OVER

Bits	Access	Name	Description & Reset Value

Bits	Access	Name	Description & Reset Value
31:1	R	RSVD_IC_CLR_TX_OVER	IC_CLR_TX_OVER Reserved bits - Read Only Exists: Always Volatile: true
0	R	CLR_TX_OVER	Read this register to clear the TX_OVER interrupt (bit 3) of the IC_RAW_INTR_STAT register. Reset value: 0x0 Exists: Always Volatile: true

12.1.5.2.24 IC_CLR_RD_REQ

- Name: Clear RD_REQ Interrupt Register.
- Description: Clear RD_REQ Interrupt Register
- Offset Address: 0x50

Table 12-1-25 Fields for Register: IC_CLR_RD_REQ

Bits	Access	Name	Description & Reset Value
31:1	R	RSVD_IC_CLR_RD_REQ	IC_CLR_RD_REQ Reserved bits - Read Only Exists: Always Volatile: true
0	R	CLR_RD_REQ	Read this register to clear the RD_REQ interrupt (bit 5) of the IC_RAW_INTR_STAT register. Reset value: 0x0 Exists: Always Volatile: true

12.1.5.2.25 IC_CLR_TX_ABRT

- Name: Clear TX_ABRT Interrupt Register.
- Description: Clear TX_ABRT Interrupt Register
- Offset Address: 0x54

Table 12-1-26 Fields for Register: IC_CLR_TX_ABRT

Bits	Access	Name	Description & Reset Value
31:1	R	RSVD_IC_CLR_TX_ABRT	IC_CLR_TX_ABRT Reserved bits - Read Only Exists: Always Volatile: true
0	R	CLR_TX_ABRT	Read this register to clear the TX_ABRT interrupt (bit 6) of the IC_RAW_INTR_STAT register, and the IC_TX_ABRT_SOURCE register. This also releases the TX FIFO from the flushed/reset state, allowing more writes to the TX FIFO. Refer to Bit 9 of the IC_TX_ABRT_SOURCE register for an exception to clearing IC_TX_ABRT_SOURCE. Reset value: 0x0 Exists: Always Volatile: true

12.1.5.2.26 IC_CLR_RX_DONE

- Name: Clear RX_DONE Interrupt Register.
- Description: Clear RX_DONE Interrupt Register
- Offset Address: 0x58

Table 12-1-27 Fields for Register: IC_CLR_RX_DONE

Bits	Access	Name	Description & Reset Value
31:1	R	RSVD_IC_CLR_RX_DONE	IC_CLR_RX_DONE Reserved bits - Read Only Exists: Always Volatile: true
0	R	CLR_RX_DONE	Read this register to clear the RX_DONE interrupt (bit 7) of the IC_RAW_INTR_STAT register. Reset value: 0x0 Exists: Always Volatile: true

12.1.5.2.27 IC_CLR_ACTIVITY

- Name: Clear ACTIVITY Interrupt Register.
- Description: Clear ACTIVITY Interrupt Register
- Offset Address: 0x5c

Table 12-1-28 Fields for Register: IC_CLR_ACTIVITY

Bits	Access	Name	Description & Reset Value
31:1	R	RSVD_IC_CLR_ACTIVITY	IC_CLR_ACTIVITY Reserved bits - Read Only Exists: Always Volatile: true
0	R	CLR_ACTIVITY	Reading this register clears the ACTIVITY interrupt if the I2C is not active anymore. If the I2C module is still active on the bus, the ACTIVITY interrupt bit continues to be set. It is automatically cleared by hardware if the module is disabled and if there is no further activity on the bus. The value read from this register to get status of the ACTIVITY interrupt (bit 8) of the IC_RAW_INTR_STAT register. Reset value: 0x0 Exists: Always Volatile: true

12.1.5.2.28 IC_CLR_STOP_DET

- Name: Clear STOP_DET Interrupt Register.
- Description: Clear STOP_DET Interrupt Register
- Offset Address: 0x60

Table 12-1-29 Fields for Register: IC_CLR_STOP_DET

Bits	Access	Name	Description & Reset Value
31:1	R	RSVD_IC_CLR_STOP_DET	IC_CLR_STOP_DET Reserved bits - Read Only Exists: Always Volatile: true
0	R	CLR_STOP_DET	Read this register to clear the STOP_DET interrupt (bit 9) of the IC_RAW_INTR_STAT register. Reset value: 0x0 Exists: Always Volatile: true

12.1.5.2.29 IC_CLR_START_DET

- Name: Clear START_DET Interrupt Register.

- Description: Clear START_DET Interrupt Register
- Offset Address: 0x64

Table 12-1-30 Fields for Register: IC_CLR_START_DET

Bits	Access	Name	Description & Reset Value
31:1	R	RSVD_IC_CLR_START_DET	IC_CLR_START_DET Reserved bits - Read Only Exists: Always Volatile: true
0	R	CLR_START_DET	Read this register to clear the START_DET interrupt (bit 10) of the IC_RAW_INTR_STAT register. Reset value: 0x0 Exists: Always Volatile: true

12.1.5.2.30 IC_CLR_GEN_CALL

- Name: Clear GEN_CALL Interrupt Register.
- Description: Clear GEN_CALL Interrupt Register
- Offset Address: 0x68

Table 12-1-31 Fields for Register: IC_CLR_GEN_CALL

Bits	Access	Name	Description & Reset Value
31:1	R	RSVD_IC_CLR_GEN_CALL	IC_CLR_GEN_CALL Reserved bits - Read Only Exists: Always Volatile: true
0	R	CLR_GEN_CALL	Read this register to clear the GEN_CALL interrupt (bit 11) of IC_RAW_INTR_STAT register. Reset value: 0x0 Exists: Always Volatile: true

12.1.5.2.31 IC_ENABLE

- Name: I2C Enable Register.
- Description: I2C Enable Register
- Offset Address: 0x6c

Table 12-1-32 Fields for Register: IC_ENABLE

Bits	Access	Name	Description & Reset Value
31:19	R	RSVD_IC_ENABLE_2	IC_ENABLE Reserved bits - Read Only Exists: Always

Bits	Access	Name	Description & Reset Value
18	R/W	SMBUS_ALERT_EN	The SMBUS_ALERT_CTRL register bit is used to control assertion of SMBALERT signal. - 1: Assert SMBALERT signal This register bit is auto-cleared after detection of Acknowledgement from master for Alert Response address. Reset value: 0x0. Values: 0x1 (ALERT_ENABLED): Slave initiates the Alert signal to indicate SMBus Host 0x0 (SUSPEND_DISABLED): Slave will not initiate the Alert signal to indicate SMBus Host. Exists: IC_SMBUS_SUSPEND_ALERT==1
17	R/W	SMBUS_SUSPEND_EN	The SMBUS_SUSPEND_EN register bit is used to control assertion and de-assertion of SMBSUS signal. 0: De-assert SMBSUS signal 1: Assert SMBSUS signal Reset value: 0x0. Values: 0x1 (ENABLED): Host/Master initiates the SMBUS system to enter Suspend Mode. 0x0 (DISABLED): Host/Master will not initiate the SMBUS system to enter Suspend Mode. Exists: IC_SMBUS_SUSPEND_ALERT==1
16	R/W	SMBUS_CLK_RESET	This bit is used in SMBus Host mode to initiate the SMBus Master Clock Reset. This bit should be enabled only when Master is in idle. Whenever this bit is enabled, the SMBCLK is held low for the IC_SCL_STUCK_TIMEOUT ic_clk cycles to reset the SMBus slave devices. Reset value: 0x0. Values: 0x1 (ENABLED): Master initiates the SMBUS Clock Reset Mechanism. 0x0 (DISABLED): Master will not initiate SMBUS Clock Reset Mechanism. Exists: IC_SMBUS==1
15:4	R	RSVD_IC_ENABLE_1	RSVD_IC_ENABLE_1 Reserved bits - Read Only Exists: Always

Bits	Access	Name	Description & Reset Value
3	R/W	SDA_STUCK_RECOVERY_ENABLE	If SDA is stuck at low indicated through the TX_ABORT interrupt (IC_TX_ABRT_SOURCE[17]), then this bit is used as a control knob to initiate the SDA Recovery Mechanism (that is, send at most 9 SCL clocks and STOP to release the SDA line) and then this bit gets auto clear. Reset value: 0x0. Values: 0x1 (SDA_STUCK_RECOVERY_ENABLED): Master initiates the SDA stuck at low recovery mechanism. 0x0 (SDA_STUCK_RECOVERY_DISABLED): Master disabled the SDA stuck at low recovery mechanism. Exists: IC_BUS_CLEAR_FEATURE==1
2	R/W	TX_CMD_BLOCK	In Master mode: 1'b1: Blocks the transmission of data on I2C bus even if Tx FIFO has data to transmit. 1'b0: The transmission of data starts on I2C bus automatically, as soon as the first data is available in the Tx FIFO. Note: To block the execution of Master commands, set the TX_CMD_BLOCK bit only when Tx FIFO is empty (IC_STATUS[2]==1) and Master is in Idle state (IC_STATUS[5] == 0). Any further commands put in the Tx FIFO are not executed until TX_CMD_BLOCK bit is unset. Reset value: IC_TX_CMD_BLOCK_DEFAULT Values: 0x1 (BLOCKED): Tx Command execution blocked 0x0 (NOT_BLOCKED): Tx Command execution not blocked Exists: Always
1	R/W	ABORT	When set, the controller initiates the transfer abort. 0: ABORT not initiated or ABORT done 1: ABORT operation in progress The software can abort the I2C transfer in master mode by setting this bit. The software can set this bit only when ENABLE is already set; otherwise, the controller ignores any write to ABORT bit. The software cannot clear the ABORT bit once set. In response to an ABORT, the controller issues a STOP and flushes the Tx FIFO after completing the current transfer, then sets the TX_ABORT interrupt after the abort operation. The ABORT bit is cleared automatically after the abort operation. For a detailed description on how to abort I2C transfers, refer to Aborting I2C Transfers . Reset value: 0x0 Values: 0x1 (ENABLED): ABORT operation in progress 0x0 (DISABLE): ABORT operation not in progress Exists: Always

Bits	Access	Name	Description & Reset Value
0	R/W	ENABLE	Controls whether the DW_apb_i2c is enabled. 0: Disables DW_apb_i2c (TX and RX FIFOs are held in an erased state) 1: Enables DW_apb_i2c Software can disable DW_apb_i2c while it is active. However, it is important that care be taken to ensure that DW_apb_i2c is disabled properly. A recommended procedure is described in Disabling DW_apb_i2c . When DW_apb_i2c is disabled, the following occurs: The TX FIFO and RX FIFO get flushed. Status bits in the IC_INTR_STAT register are still active until DW_apb_i2c goes into IDLE state. If the module is transmitting, it stops as well as deletes the contents of the transmit buffer after the current transfer is complete. If the module is receiving, the DW_apb_i2c stops the current transfer at the end of the current byte and does not acknowledge the transfer. In systems with asynchronous pclk and ic_clk when IC_CLK_TYPE parameter set to asynchronous (1), there is a two ic_clk delay when enabling or disabling the DW_apb_i2c. For a detailed description on how to disable DW_apb_i2c, refer to Disabling DW_apb_i2c Reset value: 0x0 Values: 0x1 (ENABLED): I2C is enabled 0x0 (DISABLED): I2C is disabled Exists: Always

12.1.5.2.32 IC_STATUS

- Name: I2C Status Register.
- Description: I2C Status Register This is a read-only register used to indicate the current transfer status and FIFO status. The status register may be read at any time. None of the bits in this register request an interrupt.
When the I2C is disabled by writing 0 in bit 0 of the IC_ENABLE register:
 - Bits 1 and 2 are set to 1
 - Bits 3 and 10 are set to 0
When the master or slave state machines goes to idle and ic_en=0:
 - Bits 5 and 6 are set to 0
- Offset Address: 0x70

Table 12-1-33 Fields for Register: IC_STATUS

Bits	Access	Name	Description & Reset Value
31:21	R	RSVD_IC_STATUS_2	IC_STATUS Reserved bits - Read Only Exists: Always Volatile: true

Bits	Access	Name	Description & Reset Value
20	R	SMBUS_ALERT_STATUS	This bit indicates the status of the SMBus Alert signal (ic_smbalert_in_n). This signal is asserted when the SMBus Alert signal is asserted by the SMBus Device. Reset value: 0x0. Values: 0x1 (ACTIVE): SMBUS Alert is asserted. 0x0 (INACTIVE): SMBUS Alert is not asserted. Exists: IC_SMBUS_SUSPEND_ALERT==1 Volatile: true
19	R	SMBUS_SUSPEND_STATUS	This bit indicates the status of the SMBus Suspend signal (ic_smbsus_in_n). This signal is asserted when the SMBus Suspend signal is asserted by the SMBus Host. Reset value: 0x0. Values: 0x1 (ACTIVE): SMBUS System is in Suspended mode. 0x0 (INACTIVE): SMBUS System is not in Suspended mode. Exists: IC_SMBUS_SUSPEND_ALERT==1 Volatile: true
18	R	SMBUS_SLAVE_ADDR_RESOLVED	This bit indicates whether the slave address (ic_sar) is resolved by the ARP Master. Reset value: 0x0. Values: 0x1 (ACTIVE): SMBUS Slave Address is Resolved. 0x0 (INACTIVE): SMBUS Slave Address is not Resolved. Exists: IC_SMBUS_ARP==1 Volatile: true
17	R	SMBUS_SLAVE_ADDR_VALID	This bit indicates whether the slave address (ic_sar) is valid or not. Reset value: IC_PERSISTANT_SLV_ADDR_DEFAULT Values: 0x1 (ACTIVE): SMBUS Slave Address is Valid. 0x0 (INACTIVE): SMBUS SLave Address is not valid. Exists: IC_SMBUS_ARP==1 Volatile: true

Bits	Access	Name	Description & Reset Value
16	R	SMBUS_QUICK_CMD_BIT	This bit indicates the R/W bit of the Quick command received. This bit will be cleared after the user has read this bit. Reset value: 0x0. Values: 0x1 (ACTIVE): SMBUS QUICK CMD Read/write is set to 1. 0x0 (INACTIVE): SMBUS QUICK CMD Read/write is set to 0. Exists: IC_SMBUS==1 Volatile: true
15:12	R	RSVD_IC_STATUS_1	RSVD_IC_STATUS_1 Reserved bits - Read Only Exists: Always Volatile: true
11	R	SDA_STUCK_NOT_RECOVERED	This bit indicates that SDA stuck at low is not recovered after the recovery mechanism. In Slave mode, this register bit is not applicable. Reset value: 0x0. Values: 0x1 (ACTIVE): SDA Stuck at low is recovered after recovery mechanism. 0x0 (INACTIVE): SDA Stuck at low is not recovered after recovery mechanism. Exists: IC_BUS_CLEAR_FEATURE==1 Volatile: true
10	R	SLV_HOLD_RX_FIFO_FULL	This bit indicates the BUS Hold in Slave mode due to Rx FIFO is Full and an additional byte has been received (This kind of Bus hold is applicable if IC_RX_FULL_HLD_BUS_EN is set to 1). Reset value: 0x0 Values: 0x1 (ACTIVE): Slave holds the bus due to Rx FIFO is full 0x0 (INACTIVE): Slave is not holding the bus or Bus hold is not due to Rx FIFO is full Exists: IC_STAT_FOR_CLK_STRETCH == 1 Volatile: true

Bits	Access	Name	Description & Reset Value
9	R	SLV_HOLD_TX_FIFO_EMPTY	This bit indicates the BUS Hold in Slave mode for the Read request when the Tx FIFO is empty. The Bus is in hold until the Tx FIFO has data to Transmit for the read request. Reset value: 0x0 Values: 0x1 (ACTIVE): Slave holds the bus due to Tx FIFO is empty 0x0 (INACTIVE): Slave is not holding the bus or Bus hold is not due to Tx FIFO is empty Exists: IC_STAT_FOR_CLK_STRETCH == 1 Volatile: true
8	R	MST_HOLD_RX_FIFO_FULL	This bit indicates the BUS Hold in Master mode due to Rx FIFO is Full and additional byte has been received (This kind of Bus hold is applicable if IC_RX_FULL_HLD_BUS_EN is set to 1). Reset value: 0x0 Values: 0x1 (ACTIVE): Master holds the bus due to Rx FIFO is full 0x0 (INACTIVE): Master is not holding the bus or Bus hold is not due to Rx FIFO is full Exists: IC_STAT_FOR_CLK_STRETCH == 1 Volatile: true
7	R	MST_HOLD_TX_FIFO_EMPTY	If the IC_EMPTYFIFO_HOLD_MASTER_EN parameter is set to 1, the DW_apb_i2c master stalls the write transfer when Tx FIFO is empty, and the the last byte does not have the Stop bit set. This bit indicates the BUS hold when the master holds the bus because of the Tx FIFO being empty, and the the previous transferred command does not have the Stop bit set. (This kind of Bus hold is applicable if IC_EMPTYFIFO_HOLD_MASTER_EN is set to 1). Reset value: 0x0 Values: 0x1 (ACTIVE): Master holds the bus due to Tx FIFO is empty 0x0 (INACTIVE): Master is not holding the bus or Bus hold is not due to Tx FIFO is empty Exists: IC_STAT_FOR_CLK_STRETCH == 1 Volatile: true

Bits	Access	Name	Description & Reset Value
6	R	SLV_ACTIVITY	Slave FSM Activity Status. When the Slave Finite State Machine (FSM) is not in the IDLE state, this bit is set. 0: Slave FSM is in IDLE state so the Slave part of DW_apb_i2c is not Active 1: Slave FSM is not in IDLE state so the Slave part of DW_apb_i2c is Active Reset value: 0x0 Values: 0x1 (ACTIVE): Slave not idle 0x0 (IDLE): Slave is idle Exists: Always Volatile: true
5	R	MST_ACTIVITY	Master FSM Activity Status. When the Master Finite State Machine (FSM) is not in the IDLE state, this bit is set. 0: Master FSM is in IDLE state so the Master part of DW_apb_i2c is not Active 1: Master FSM is not in IDLE state so the Master part of DW_apb_i2c is Active Note: IC_STATUS[0]-that is, ACTIVITY bit-is the OR of SLV_ACTIVITY and MST_ACTIVITY bits. Reset value: 0x0 Values: 0x1 (ACTIVE): Master not idle 0x0 (IDLE): Master is idle Exists: Always Volatile: true
4	R	RFF	Receive FIFO Completely 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 Reset value: 0x0 Values: 0x1 (FULL): Rx FIFO is full 0x0 (NOT_FULL): Rx FIFO not full Exists: Always Volatile: true
3	R	RFNE	Receive FIFO Not Empty. This bit is set when the receive FIFO contains one or more entries; it is cleared when the receive FIFO is empty. 0: Receive FIFO is empty 1: Receive FIFO is not empty Reset value: 0x0 Values: 0x1 (NOT_EMPTY): Rx FIFO not empty 0x0 (EMPTY): Rx FIFO is empty Exists: Always Volatile: true

Bits	Access	Name	Description & Reset Value
2	R	TFE	Transmit FIFO Completely Empty. When the transmit FIFO is completely empty, this bit is set. When it 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 Reset value: 0x1 Values: 0x1 (EMPTY): Tx FIFO is empty 0x0 (NON_EMPTY): Tx FIFO not empty Exists: Always Volatile: true
1	R	TFNF	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 Reset value: 0x1 Values: 0x1 (NOT_FULL): Tx FIFO not full 0x0 (FULL): Tx FIFO is full Exists: Always Volatile: true
0	R	ACTIVITY	I2C Activity Status. Reset value: 0x0 Values: 0x1 (ACTIVE): I2C is active 0x0 (INACTIVE): I2C is idle Exists: Always Volatile: true

12.1.5.2.33 IC_TXFLR

- Name: I2C Transmit FIFO Level Register.
- Description: I2C Transmit FIFO Level Register This register contains the number of valid data entries in the transmit FIFO buffer. It is cleared whenever:
 - The I2C is disabled
 - There is a transmit abort - that is, TX_ABRT bit is set in the IC_RAW_INTR_STAT register
 - The slave bulk transmit mode is aborted

The register increments whenever data is placed into the transmit FIFO and decrements when data is taken from the transmit FIFO.
- Offset Address: 0x74

Table 12-1-34 Fields for Register: IC_TXFLR

Bits	Access	Name	Description & Reset Value

Bits	Access	Name	Description & Reset Value
31:y	R	RSVD_TXFLR	TXFLR Register field Reserved bits - Read Only Exists: Always Volatile: true Range Variable[y]: TX_ABW_P1
x:0	R	TXFLR	Transmit FIFO Level. Contains the number of valid data entries in the transmit FIFO. Reset value: 0x0 Exists: Always Volatile: true Range Variable[x]: TX_ABW_P1 - 1

12.1.5.2.34 IC_RXFLR

- Name: I2C Receive FIFO Level Register.
- Description: I2C Receive FIFO Level Register This register contains the number of valid data entries in the receive FIFO buffer. It is cleared whenever:
 - The I2C is disabled
 - Whenever there is a transmit abort caused by any of the events tracked in IC_TX_ABRT_SOURCE
 The register increments whenever data is placed into the receive FIFO and decrements when data is taken from the receive FIFO.
- Offset Address: 0x78

Table 12-1-35 Fields for Register: IC_RXFLR

Bits	Access	Name	Description & Reset Value
31:y	R	RSVD_RXFLR	RXFLR Reserved bits - Read Only Exists: Always Volatile: true Range Variable[y]: RX_ABW_P1
x:0	R	RXFLR	Receive FIFO Level. Contains the number of valid data entries in the receive FIFO. Reset value: 0x0 Exists: Always Volatile: true Range Variable[x]: RX_ABW_P1 - 1

12.1.5.2.35 IC_SDA_HOLD

- Name: I2C SDA Hold Time Length Register.
- Description: I2C SDA Hold Time Length Register The bits [15:0] of this register are used to control the hold time of SDA during transmit in both slave and master mode (after SCL goes from HIGH to LOW).
The bits [23:16] of this register are used to extend the SDA transition (if any) whenever SCL is HIGH in the receiver in either master or slave mode.
Writes to this register succeed only when IC_ENABLE[0]=0.
The values in this register are in units of ic_clk period. The value programmed in IC_SDA_TX_HOLD must be greater than the minimum hold time in each mode one cycle in master mode, seven cycles in slave mode for the value to be implemented.
The programmed SDA hold time during transmit (IC_SDA_TX_HOLD) cannot exceed at any time the duration of the low part of scl. Therefore the programmed value cannot be larger than N_SCL_LOW-2, where N_SCL_LOW is the duration of the low part of the scl period measured in ic_clk cycles.
- Offset Address: 0x7c

Table 12-1-36 Fields for Register: IC_SDA_HOLD

Bits	Access	Name	Description & Reset Value
31:24	R	RSVD_IC_SDA_HOLD	IC_SDA_HOLD Reserved bits - Read Only Exists: Always
23:16	R/W	IC_SDA_RX_HOLD	Sets the required SDA hold time in units of ic_clk period, when DW_apb_i2c acts as a receiver. Reset value: IC_DEFAULT_SDA_HOLD[23:16]. Exists: Always
15:0	R/W	IC_SDA_TX_HOLD	Sets the required SDA hold time in units of ic_clk period, when DW_apb_i2c acts as a transmitter. Reset value: IC_DEFAULT_SDA_HOLD[15:0]. Exists: Always

12.1.5.2.36 IC_TX_ABRT_SOURCE

- Name: I2C Transmit Abort Source Register.
- Description: I2C Transmit Abort Source Register This register has 32 bits that indicate the source of the TX_ABRT bit. Except for Bit 9, this register is cleared whenever the IC_CLR_TX_ABRT register or the IC_CLR_INTR register is read. To clear Bit 9, the source of the ABRT_SBYTE_NORSTRT must be fixed first; RESTART must be enabled (IC_CON[5]=1), the SPECIAL bit must be cleared (IC_TAR[11]), or the GC_OR_START bit must be cleared (IC_TAR[10]).
Once the source of the ABRT_SBYTE_NORSTRT is fixed, then this bit can be cleared in the same manner as other bits in this register. If the source of the ABRT_SBYTE_NORSTRT is not fixed before attempting to clear this bit, Bit 9 clears for one cycle and is then re-asserted.
- Offset Address: 0x80

Table 12-1-37 Fields for Register: IC_TX_ABRT_SOURCE

Bits	Access	Name	Description & Reset Value
31:23	R	TX_FLUSH_CNT	This field indicates the number of Tx FIFO Data Commands which are flushed due to TX_ABRT interrupt. It is cleared whenever I2C is disabled. Reset value: 0x0 Role of DW_apb_i2c: Master-Transmitter or Slave-Transmitter Exists: Always Volatile: true
22:21	R	RSVD_IC_TX_ABRT_SOURCE	IC_TX_ABRT_SOURCE Reserved bits - Read Only Exists: Always Volatile: true

Bits	Access	Name	Description & Reset Value
20	R	ABRT_DEVICE_WRITE	This is a master-mode-only bit. Master is initiating the DEVICE_ID transfer and the Tx-FIFO consists of write commands. Reset value: 0x0 Role of DW_apb_i2c: Master Values: 0x1 (ACTIVE): This abort is generated because of NOACK for Slave address 0x0 (INACTIVE): This abort is not generated Exists: IC_DEVICE_ID == 1 Volatile: true
19	R	ABRT_DEVICE_SLVADDR_NOACK	This is a master-mode-only bit. Master is initiating the DEVICE_ID transfer and the slave address sent was not acknowledged by any slave. Reset value: 0x0 Role of DW_apb_i2c: Master Values: 0x1 (ACTIVE): This abort is generated because of NOACK for Slave address 0x0 (INACTIVE): This abort is not generated Exists: IC_DEVICE_ID == 1 Volatile: true
18	R	ABRT_DEVICE_NOACK	This is a master-mode-only bit. Master is initiating the DEVICE_ID transfer and the device id sent was not acknowledged by any slave. Reset value: 0x0 Role of DW_apb_i2c: Master Values: 0x1 (ACTIVE): This abort is generated because of NOACK for DEVICE-ID 0x0 (INACTIVE): This abort is not generated Exists: IC_DEVICE_ID == 1 Volatile: true
17	R	ABRT_SDA_STUCK_AT_LOW	This is a master-mode-only bit. Master detects the SDA Stuck at low for the IC_SDA_STUCK_AT_LOW_TIMEOUT value of ic_clks. Reset value: 0x0 Role of DW_apb_i2c: Master Values: 0x1 (ACTIVE): This abort is generated because of Sda stuck at low for IC_SDA_STUCK_AT_LOW_TIMEOUT value of ic_clks 0x0 (INACTIVE): This abort is not generated Exists: IC_BUS_CLEAR_FEATURE == 1 Volatile: true

Bits	Access	Name	Description & Reset Value
16	R	ABRT_USER_ABRT	This is a master-mode-only bit. Master has detected the transfer abort (IC_ENABLE[1]) Reset value: 0x0 Role of DW_apb_i2c: Master-Transmitter Values: 0x1 (ABRT_USER_ABRT_GENERATED): Transfer abort detected by master 0x0 (ABRT_USER_ABRT_VOID): Transfer abort detected by master- scenario not present Exists: Always Volatile: true
15	R	ABRT_SLVRD_INTX	1: When the processor side responds to a slave mode request for data to be transmitted to a remote master and user writes a 1 in CMD (bit 8) of IC_DATA_CMD register. Reset value: 0x0 Role of DW_apb_i2c: Slave-Transmitter Values: 0x1 (ABRT_SLVRD_INTX_GENERATED): Slave trying to transmit to remote master in read mode 0x0 (ABRT_SLVRD_INTX_VOID): Slave trying to transmit to remote master in read mode- scenario not present Exists: IC_ULTRA_FAST_MODE==0 Volatile: true
14	R	ABRT_SLV_ARBLOST	This field indicates that a Slave has lost the bus while transmitting data to a remote master. IC_TX_ABRT_SOURCE[12] is set at the same time. Note: Even though the slave never 'owns' the bus, something could go wrong on the bus. This is a fail safe check. For instance, during a data transmission at the low-to- high transition of SCL, if what is on the data bus is not what is supposed to be transmitted, then DW_apb_i2c no longer own the bus. Reset value: 0x0 Role of DW_apb_i2c: Slave-Transmitter Values: 0x1 (ABRT_SLV_ARBLOST_GENERATED): Slave lost arbitration to remote master 0x0 (ABRT_SLV_ARBLOST_VOID): Slave lost arbitration to remote master- scenario not present Exists: IC_ULTRA_FAST_MODE==0 Volatile: true

Bits	Access	Name	Description & Reset Value
13	R	ABRT_SLVFLUSH_TXFIFO	This field specifies that the Slave has received a read command and some data exists in the TX FIFO, so the slave issues a TX_ABRT interrupt to flush old data in TX FIFO. Reset value: 0x0 Role of DW_apb_i2c: Slave-Transmitter Values: 0x1 (ABRT_SLVFLUSH_TXFIFO_GENERATED): Slave flushes existing data in TX-FIFO upon getting read command 0x0 (ABRT_SLVFLUSH_TXFIFO_VOID): Slave flushes existing data in TX-FIFO upon getting read command- scenario not present Exists: IC_ULTRA_FAST_MODE==0 Volatile: true
12	R	ARB_LOST	This field specifies that the Master has lost arbitration, or if IC_TX_ABRT_SOURCE[14] is also set, then the slave transmitter has lost arbitration. Reset value: 0x0 Role of DW_apb_i2c: Master-Transmitter or Slave-Transmitter Values: 0x1 (ABRT_LOST_GENERATED): Master or Slave-Transmitter lost arbitration 0x0 (ABRT_LOST_VOID): Master or Slave-Transmitter lost arbitration- scenario not present Exists: IC_ULTRA_FAST_MODE==0 Volatile: true
11	R	ABRT_MASTER_DIS	This field indicates that the User tries to initiate a Master operation with the Master mode disabled. Reset value: 0x0 Role of DW_apb_i2c: Master-Transmitter or Master-Receiver Values: 0x1 (ABRT_MASTER_DIS_GENERATED): User initiating master operation when MASTER disabled 0x0 (ABRT_MASTER_DIS_VOID): User initiating master operation when MASTER disabled- scenario not present Exists: Always Volatile: true

Bits	Access	Name	Description & Reset Value
10	R	ABRT_10B_RD_NORSTRT	This field indicates that the restart is disabled (IC_RESTART_EN bit (IC_CON[5]) =0) and the master sends a read command in 10-bit addressing mode. Reset value: 0x0 Role of DW_apb_i2c: Master-Receiver Values: 0x1 (ABRT_10B_RD_GENERATED): Master trying to read in 10Bit addressing mode when RESTART disabled 0x0 (ABRT_10B_RD_VOID): Master not trying to read in 10Bit addressing mode when RESTART disabled Exists: IC_ULTRA_FAST_MODE==0 Volatile: true
9	R	ABRT_SBYTE_NORSTRT	To clear Bit 9, the source of the ABRT_SBYTE_NORSTRT must be fixed first; restart must be enabled (IC_CON[5]=1), the SPECIAL bit must be cleared (IC_TAR[11]), or the GC_OR_START bit must be cleared (IC_TAR[10]). Once the source of the ABRT_SBYTE_NORSTRT is fixed, then this bit can be cleared in the same manner as other bits in this register. If the source of the ABRT_SBYTE_NORSTRT is not fixed before attempting to clear this bit, bit 9 clears for one cycle and then gets reasserted. When this field is set to 1, the restart is disabled (IC_RESTART_EN bit (IC_CON[5]) =0) and the user is trying to send a START Byte. Reset value: 0x0 Role of DW_apb_i2c: Master Values: 0x1 (ABRT_SBYTE_NORSTRT_GENERATED): User trying to send START byte when RESTART disabled 0x0 (ABRT_SBYTE_NORSTRT_VOID): User trying to send START byte when RESTART disabled- scenario not present Exists: Always Volatile: true

Bits	Access	Name	Description & Reset Value
8	R	ABRT_HS_NORSTRT	This field indicates that the restart is disabled (IC_RESTART_EN bit (IC_CON[5]) =0) and the user is trying to use the master to transfer data in High Speed mode. Reset value: 0x0 Role of DW_apb_i2c: Master-Transmitter or Master-Receiver Values: 0x1 (ABRT_HS_NORSTRT_GENERATED): User trying to switch Master to HS mode when RESTART disabled 0x0 (ABRT_HS_NORSTRT_VOID): User trying to switch Master to HS mode when RESTART disabled- scenario not present Exists: IC_ULTRA_FAST_MODE==0 Volatile: true
7	R	ABRT_SBYTE_ACKDET	This field indicates that the Master has sent a START Byte and the START Byte was acknowledged (wrong behavior). Reset value: 0x0 Role of DW_apb_i2c: Master Values: 0x1 (ABRT_SBYTE_ACKDET_GENERATED): ACK detected for START byte 0x0 (ABRT_SBYTE_ACKDET_VOID): ACK detected for START byte- scenario not present Exists: IC_ULTRA_FAST_MODE==0 Volatile: true
6	R	ABRT_HS_ACKDET	This field indicates that the Master is in High Speed mode and the High Speed Master code was acknowledged (wrong behavior). Reset value: 0x0 Role of DW_apb_i2c: Master Values: 0x1 (ABRT_HS_ACK_GENERATED): HS Master code ACKed in HS Mode 0x0 (ABRT_HS_ACK_VOID): HS Master code ACKed in HS Mode- scenario not present Exists: IC_ULTRA_FAST_MODE==0 Volatile: true

Bits	Access	Name	Description & Reset Value
5	R	ABRT_GCALL_READ	This field indicates that DW_apb_i2c in the master mode has sent a General Call but the user programmed the byte following the General Call to be a read from the bus (IC_DATA_CMD[9] is set to 1). Reset value: 0x0 Role of DW_apb_i2c: Master-Transmitter Values: 0x1 (ABRT_GCALL_READ_GENERATED): GCALL is followed by read from bus 0x0 (ABRT_GCALL_READ_VOID): GCALL is followed by read from bus-scenario not present Exists: IC_ULTRA_FAST_MODE==0 Volatile: true
4	R	ABRT_GCALL_NOACK	This field indicates that DW_apb_i2c in master mode has sent a General Call and no slave on the bus acknowledged the General Call. Reset value: 0x0 Role of DW_apb_i2c: Master-Transmitter Values: 0x1 (ABRT_GCALL_NOACK_GENERATED): GCALL not ACKed by any slave 0x0 (ABRT_GCALL_NOACK_VOID): GCALL not ACKed by any slave-scenario not present Exists: IC_ULTRA_FAST_MODE==0 Volatile: true
3	R	ABRT_TXDATA_NOACK	This field indicates the master-mode only bit. When the master receives an acknowledgement for the address, but when it sends data byte(s) following the address, it did not receive an acknowledge from the remote slave(s). Reset value: 0x0 Role of DW_apb_i2c: Master-Transmitter Values: 0x1 (ABRT_TXDATA_NOACK_GENERATED): Transmitted data not ACKed by addressed slave 0x0 (ABRT_TXDATA_NOACK_VOID): Transmitted data non-ACKed by addressed slave-scenario not present Exists: IC_ULTRA_FAST_MODE==0 Volatile: true

Bits	Access	Name	Description & Reset Value
2	R	ABRT_10ADDR2_NOACK	This field indicates that the Master is in 10-bit address mode and that the second address byte of the 10-bit address was not acknowledged by any slave. Reset value: 0x0 Role of DW_apb_i2c: Master-Transmitter or Master-Receiver Values: 0x1 (ACTIVE): Byte 2 of 10Bit Address not ACKed by any slave 0x0 (INACTIVE): This abort is not generated Exists: IC_ULTRA_FAST_MODE==0 Volatile: true
1	R	ABRT_10ADDR1_NOACK	This field indicates that the Master is in 10-bit address mode and the first 10-bit address byte was not acknowledged by any slave. Reset value: 0x0 Role of DW_apb_i2c: Master-Transmitter or Master-Receiver Values: 0x1 (ACTIVE): Byte 1 of 10Bit Address not ACKed by any slave 0x0 (INACTIVE): This abort is not generated Exists: IC_ULTRA_FAST_MODE==0 Volatile: true
0	R	ABRT_7B_ADDR_NOACK	This field indicates that the Master is in 7-bit addressing mode and the address sent was not acknowledged by any slave. Reset value: 0x0 Role of DW_apb_i2c: Master-Transmitter or Master-Receiver Values: 0x1 (ACTIVE): This abort is generated because of NOACK for 7-bit address 0x0 (INACTIVE): This abort is not generated Exists: IC_ULTRA_FAST_MODE==0 Volatile: true

12.1.5.2.37 IC_SLV_DATA_NACK_ONLY

- Name: Generate Slave Data NACK Register.
- Description: Generate Slave Data NACK Register The register is used to generate a NACK for the data part of a transfer when I2C controller is acting as a slave-receiver. This register only exists when the IC_SLV_DATA_NACK_ONLY parameter is set to 1. When this parameter disabled, this register does not exist and writing to the register's address has no effect.
A write can occur on this register if both of the following conditions are met:
 - I2C controller is disabled (IC_ENABLE[0] = 0)
 - Slave part is inactive (IC_STATUS[6] = 0)

- Note: The IC_STATUS[6] is a register read-back location for the internal slv_activity signal; the user should poll this before writing the ic_slv_data_nack_only bit.

- Offset Address: 0x84

Table 12-1-38 Fields for Register: IC_SLV_DATA_NACK_ONLY

Bits	Access	Name	Description & Reset Value
31:1	R	RSVD_IC_SLV_DATA_NACK_ONLY	IC_SLV_DATA_NACK_ONLY Reserved bits - Read Only Exists: Always
0	R/W	NACK	Generate NACK. This NACK generation only occurs when DW_apb_i2c is a slave-receiver. If this register is set to a value of 1, it can only generate a NACK after a data byte is received; hence, the data transfer is aborted and the data received is not pushed to the receive buffer. When the register is set to a value of 0, it generates NACK/ACK, depending on normal criteria. 1: generate NACK after data byte received 0: generate NACK/ACK normally Reset value: 0x0 Values: 0x1 (ENABLED): Slave reciever generates NACK upon data reception only 0x0 (DISABLED): Slave reciever generates NACK normally Exists: Always

12.1.5.2.38 IC_DMA_CR

- Name: DMA Control Register.
- Description: DMA Control Register This register is only valid when I2C controller is configured with a set of DMA Controller interface signals (IC_HAS_DMA = 1). When I2C controller is not configured for DMA operation, this register does not exist and writing to the register's address has no effect and reading from this register address will return zero. The register is used to enable the DMA Controller interface operation. There is a separate bit for transmit and receive. This can be programmed regardless of the state of IC_ENABLE.
- Offset Address: 0x88

Table 12-1-39 Fields for Register: IC_DMA_CR

Bits	Access	Name	Description & Reset Value
31:2	R	RSVD_IC_DMA_CR_2_31	RSVD_IC_DMA_CR_2_31 Reserved bits - Read Only Exists: Always
1	R/W	TDMAE	Transmit DMA Enable. This bit enables/disables the transmit FIFO DMA channel. Reset value: 0x0 Values: 0x1 (ENABLED): Transmit FIFO DMA channel enabled 0x0 (DISABLED): transmit FIFO DMA channel disabled Exists: Always

Bits	Access	Name	Description & Reset Value
0	R/W	RDMAE	Receive DMA Enable. This bit enables/disables the receive FIFO DMA channel. Reset value: 0x0 Values: 0x1 (ENABLED): Receive FIFO DMA channel enabled 0x0 (DISABLED): Receive FIFO DMA channel disabled Exists: Always

12.1.5.2.39 IC_DMA_TDLR

- Name: DMA Transmit Data Level Register.
- Description: DMA Transmit Data Level Register This register is only valid when the I2C controller is configured with a set of DMA interface signals (IC_HAS_DMA = 1). When I2C controller is not configured for DMA operation, this register does not exist; writing to its address has no effect; reading from its address returns zero.
- Offset Address: 0x8c

Table 12-1-40 Fields for Register: IC_DMA_TDLR

Bits	Access	Name	Description & Reset Value
31:y	R	RSVD_DMA_TDLR	DMA_TDLR Reserved bits - Read Only Exists: Always Range Variable[y]: TX_ABW
x:0	R/W	DMATDL	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. Reset value: 0x0 Exists: Always Range Variable[x]: TX_ABW - 1

12.1.5.2.40 IC_DMA_RDLR

- Name: I2C Receive Data Level Register.
- Description: I2C Receive Data Level Register This register is only valid when I2C controller is configured with a set of DMA interface signals (IC_HAS_DMA = 1). When I2C controller is not configured for DMA operation, this register does not exist; writing to its address has no effect; reading from its address returns zero.
- Offset Address: 0x90

Table 12-1-41 Fields for Register: IC_DMA_RDLR

Bits	Access	Name	Description & Reset Value
31:y	R	RSVD_DMA_RDLR	DMA_RDLR Reserved bits - Read Only Exists: Always Range Variable[y]: RX_ABW

Bits	Access	Name	Description & Reset Value
x:0	R/W	DMARDL	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 more than this field value + 1, and RDMAE =1. For instance, when DMARDL is 0, then dma_rx_req is asserted when 1 or more data entries are present in the receive FIFO. Reset value: 0x0 Exists: Always Range Variable[x]: RX_ABW - 1

12.1.5.2.41 IC_SDA_SETUP

- Name: I2C SDA Setup Register.
- Description: I2C SDA Setup Register This register controls the amount of time delay (in terms of number of ic_clk clock periods) introduced in the rising edge of SCL - relative to SDA changing - when I2C controller services a read request in a slave-transmitter operation. The relevant I2C requirement is tSU:DAT (note 4) as detailed in the I2C Bus Specification. This register must be programmed with a value equal to or greater than 2.
Writes to this register succeed only when IC_ENABLE[0] = 0.
Note: The length of setup time is calculated using $[(IC\_SDA\_SETUP - 1) * (ic\_clk\_period)]$, so if the user requires 10 ic_clk periods of setup time, they should program a value of 11. The IC_SDA_SETUP register is only used by the I2C controller when operating as a slave transmitter.
- Offset Address: 0x94

Table 12-1-42 Fields for Register: IC_SDA_SETUP

Bits	Access	Name	Description & Reset Value
31:8	R	RSVD_IC_SDA_SETUP	IC_SDA_SETUP Reserved bits - Read Only Exists: Always
7:0	R/W	SDA_SETUP	SDA Setup. It is recommended that if the required delay is 1000ns, then for an ic_clk frequency of 10 MHz, IC_SDA_SETUP should be programmed to a value of 11. IC_SDA_SETUP must be programmed with a minimum value of 2. Reset value: 0x64, but can be hardcoded by setting the IC_DEFAULT_SDA_SETUP configuration parameter. Exists: Always

12.1.5.2.42 IC_ACK_GENERAL_CALL

- Name: I2C ACK General Call Register.
- Description: I2C ACK General Call Register The register controls whether I2C controller responds with a ACK or NACK when it receives an I2C General Call address.
This register is applicable only when the I2C controller is in slave mode.
- Offset Address: 0x98

Table 12-1-43 Fields for Register: IC_ACK_GENERAL_CALL

Bits	Access	Name	Description & Reset Value
31:1	R	RSVD_IC_ACK_GEN_1_31	RSVD_IC_ACK_GEN_1_31 Reserved bits - Read Only Exists: Always

Bits	Access	Name	Description & Reset Value
0	R/W	ACK_GEN_CALL	ACK General Call. When set to 1, DW_apb_i2c responds with a ACK (by asserting ic_data_oe) when it receives a General Call. Otherwise, DW_apb_i2c responds with a NACK (by negating ic_data_oe). Reset value: 0x1, but can be hardcoded by setting the IC_DEFAULT_ACK_GENERAL_CALL configuration parameter. Values: 0x1 (ENABLED): Generate ACK for a General Call 0x0 (DISABLED): Generate NACK for General Call Exists: Always

12.1.5.2.43 IC_ENABLE_STATUS

- Name: I2C Enable Status Register.
- Description: I2C Enable Status Register The register is used to report the I2C controller hardware status when the IC_ENABLE[0] register is set from 1 to 0; that is, when I2C controller is disabled.
If IC_ENABLE[0] has been set to 1, bits 2:1 are forced to 0, and bit 0 is forced to 1.
If IC_ENABLE[0] has been set to 0, bits 2:1 is only be valid as soon as bit 0 is read as '0'.
Note: When IC_ENABLE[0] has been set to 0, a delay occurs for bit 0 to be read as 0 because disabling the I2C controller depends on I2C bus activities.
- Offset Address: 0x9c

Table 12-1-44 Fields for Register: IC_ENABLE_STATUS

Bits	Access	Name	Description & Reset Value
31:3	R	RSVD_IC_ENABLE_STATUS	IC_ENABLE_STATUS Reserved bits - Read Only Exists: Always Volatile: true

Bits	Access	Name	Description & Reset Value
2	R	SLV_RX_DATA_LOST	Slave Received Data Lost. This bit indicates if a Slave-Receiver operation has been aborted with at least one data byte received from an I2C transfer due to the setting bit 0 of IC_ENABLE from 1 to 0. When read as 1, DW_apb_i2c is deemed to have been actively engaged in an aborted I2C transfer (with matching address) and the data phase of the I2C transfer has been entered, even though a data byte has been responded with a NACK. Note: If the remote I2C master terminates the transfer with a STOP condition before the DW_apb_i2c has a chance to NACK a transfer, and IC_ENABLE[0] has been set to 0, then this bit is also set to 1. When read as 0, DW_apb_i2c is deemed to have been disabled without being actively involved in the data phase of a Slave-Receiver transfer. Note: The CPU can safely read this bit when IC_EN (bit 0) is read as 0. Reset value: 0x0 Values: 0x1 (ACTIVE): Slave RX Data is lost 0x0 (INACTIVE): Slave RX Data is not lost Exists: Always Volatile: true

Bits	Access	Name	Description & Reset Value
1	R	SLV_DISABLED_WHILE_BUSY	Slave Disabled While Busy (Transmit, Receive). This bit indicates if a potential or active Slave operation has been aborted due to the setting bit 0 of the IC_ENABLE register from 1 to 0. This bit is set when the CPU writes a 0 to the IC_ENABLE register while: (a) DW_apb_i2c is receiving the address byte of the Slave- Transmitter operation from a remote master; OR, (b) address and data bytes of the Slave- Receiver operation from a remote master. When read as 1, DW_apb_i2c is deemed to have forced a NACK during any part of an I2C transfer, irrespective of whether the I2C address matches the slave address set in DW_apb_i2c (IC_SAR register) OR if the transfer is completed before IC_ENABLE is set to 0 but has not taken effect. Note: If the remote I2C master terminates the transfer with a STOP condition before the DW_apb_i2c has a chance to NACK a transfer, and IC_ENABLE[0] has been set to 0, then this bit will also be set to 1. When read as 0, DW_apb_i2c is deemed to have been disabled when there is master activity, or when the I2C bus is idle. Note: The CPU can safely read this bit when IC_EN (bit 0) is read as 0. Reset value: 0x0 Values: 0x1 (ACTIVE): Slave is disabled when it is active 0x0 (INACTIVE): Slave is disabled when it is idle Exists: Always Volatile: true
0	R	IC_EN	ic_en Status. This bit always reflects the value driven on the output port ic_en. When read as 1, DW_apb_i2c is deemed to be in an enabled state. When read as 0, DW_apb_i2c is deemed completely inactive. Note: The CPU can safely read this bit anytime. When this bit is read as 0, the CPU can safely read SLV_RX_DATA_LOST (bit 2) and SLV_DISABLED_WHILE_BUSY (bit 1). Reset value: 0x0 Values: 0x1 (ENABLED): I2C enabled 0x0 (DISABLED): I2C disabled Exists: Always Volatile: true

12.1.5.2.44 IC_FS_SPKLEN

- Name: I2C SS, FS or FM+ spike suppression limit register.
- Description: I2C SS, FS or FM+ spike suppression limit This register is used to store the duration, measured in ic_clk cycles, of the longest spike that is filtered out by the spike suppression logic w hen the component is

operating in SS, FS or FM+ modes. The relevant I2C requirement is tSP (table 4) as detailed in the I2C Bus Specification. This register must be programmed with a minimum value of 1.

- Offset Address: 0xa0

Table 12-1-45 Fields for Register: IC_FS_SPKLEN

Bits	Access	Name	Description & Reset Value
31:8	R	RSVD_IC_FS_SPKLEN	IC_FS_SPKLEN Reserved bits - Read Only Exists: Always
7:0	R/W	IC_FS_SPKLEN	This register must be set before any I2C bus transaction can take place to ensure stable operation. This register sets the duration, measured in ic_clk cycles, of the longest spike in the SCL or SDA lines that will be filtered out by the spike suppression logic. This register can be written only when the I2C interface is disabled which corresponds to the IC_ENABLE[0] register being set to 0. Writes at other times have no effect. The minimum valid value is 1; hardware prevents values less than this being written, and if attempted results in 1 being set. or more information, refer to Spike Suppression . Reset value: IC_DEFAULT_FS_SPKLEN configuration parameter. Exists: Always

12.1.5.2.45 IC_UFM_SPKLEN

- Name: I2C UFM spike suppression limit register.
- Description: I2C UFM spike suppression limit This register is used to store the duration, measured in ic_clk cycles, of the longest spike that is filtered out by the spike suppression logic when the component is operating in Ultra-Fast mode. The relevant I2C requirement is tSP (table 13) as detailed in the I2C Bus Specification. This register must be programmed with a minimum value of 1.
- Offset Address: 0xa0

Table 12-1-46 Fields for Register: IC_UFM_SPKLEN

Bits	Access	Name	Description & Reset Value
31:8	R	RSVD_IC_UFM_SPKLEN	IC_UFM_SPKLEN Reserved bits - Read Only Exists: Always
7:0	R/W	IC_UFM_SPKLEN	This register must be set before any I2C bus transaction can take place to ensure stable operation. This register sets the duration, measured in ic_clk cycles, of the longest spike in the SCL or SDA lines that will be filtered out by the spike suppression logic. This register can be written only when the I2C interface is disabled which corresponds to the IC_ENABLE[0] register being set to 0. Writes at other times have no effect. The minimum valid value is 1; hardware prevents values less than this being written, and if attempted results in 1 being set. Reset value: IC_DEFAULT_UFM_SPKLEN configuration parameter. Exists: Always

12.1.5.2.46 IC_HS_SPKLEN

- Name: I2C HS spike suppression limit register.

- Description: I2C HS spike suppression limit register This register is used to store the duration, measured in ic_clk cycles, of the longest spike that is filtered out by the spike suppression logic when the component is operating in HS modes. The relevant I2C requirement is tSP (table 6) as detailed in the I2C Bus Specification. This register must be programmed with a minimum value of 1 and is implemented only if the component is configured to support HS mode; that is, if the IC_MAX_SPEED_MODE parameter is set to 3.
- Offset Address: 0xa4

Table 12-1-47 Fields for Register: IC_HS_SPKLEN

Bits	Access	Name	Description & Reset Value
31:8	R	RSVD_IC_HS_SPKLEN	IC_HS_SPKLEN Reserved bits - Read Only Exists: Always
7:0	R/W	IC_HS_SPKLEN	This register must be set before any I2C bus transaction can take place to ensure stable operation. This register sets the duration, measured in ic_clk cycles, of the longest spike in the SCL or SDA lines that will be filtered out by the spike suppression logic; for more information, refer to Spike Suppression This register can be written only when the I2C interface is disabled which corresponds to the IC_ENABLE[0] register being set to 0. Writes at other times have no effect. The minimum valid value is 1; hardware prevents values less than this being written, and if attempted results in 1 being set. Reset value: IC_DEFAULT_HS_SPKLEN configuration parameter. Exists: Always

12.1.5.2.47 IC_CLR_RESTART_DET

- Name: Clear RESTART_DET Interrupt Register
- Description: Clear RESTART_DET Interrupt Register
- Offset Address: 0xa8

Table 12-1-48 Fields for Register: IC_CLR_RESTART_DET

Bits	Access	Name	Description & Reset Value
31:1	R	RSVD_IC_CLR_RESTART_DET	IC_CLR_RESTART_DET Reserved bits - Read Only Exists: Always Volatile: true
0	R	CLR_RESTART_DET	Read this register to clear the RESTART_DET interrupt (bit 12) of IC_RAW_INTR_STAT register. Reset value: 0x0 Exists: Always Volatile: true

12.1.5.2.48 IC_SCL_STUCK_AT_LOW_TIMEOUT

- Name: I2C SCL Stuck at Low Timeout Register.
- Description: I2C SCL Stuck at Low Timeout This register is used to store the duration, measured in ic_clk cycles, used to Generate an Interrupt (SCL_STUCK_AT_LOW) if SCL is held low for the IC_SCL_STUCK_LOW_TIMEOUT duration.
- Offset Address: 0xac

Table 12-1-49 Fields for Register: IC_SCL_STUCK_AT_LOW_TIMEOUT

Bits	Access	Name	Description & Reset Value
31:0	R/W	IC_SCL_STUCK_LOW_TIMEOUT	DW_apb_i2c generate the interrupt to indicate SCL stuck at low (SCL_STUCK_AT_LOW) if it detects the SCL stuck at low for the IC_SCL_STUCK_LOW_TIMEOUT in units of ic_clk period. This register can be written only when the I2C interface is disabled which corresponds to the IC_ENABLE[0] register being set to 0. Writes at other times have no effect. Reset value: IC_SCL_STUCK_TIMEOUT_DEFAULT Parameter Exists: Always

12.1.5.2.49 IC_SDA_STUCK_AT_LOW_TIMEOUT

- Name: I2C SDA Stuck at Low Timeout register.
- Description: I2C SDA Stuck at Low Timeout This register is used to store the duration, measured in ic_clk cycles, used to Recover the Data (SDA) line through sending SCL pulses if SDA is held low for the mentioned duration.
- Offset Address: 0xb0

Table 12-1-50 Fields for Register: IC_SDA_STUCK_AT_LOW_TIMEOUT

Bits	Access	Name	Description & Reset Value
31:0	R/W	IC_SDA_STUCK_LOW_TIMEOUT	DW_apb_i2c initiates the recovery of SDA line through enabling the SDA_STUCK_RECOVERY_EN (IC_ENABLE[3]) register bit, if it detects the SDA stuck at low for the IC_SDA_STUCK_LOW_TIMEOUT in units of ic_clk period. Reset value: IC_SDA_STUCK_TIMEOUT_DEFAULT Parameter Exists: Always

12.1.5.2.50 IC_CLR_SCL_STUCK_DET

- Name: Clear SCL Stuck at Low Detect interrupt Register.
- Description: Clear SCL Stuck at Low Detect Interrupt Register
- Offset Address: 0xb4

Table 12-1-51 Fields for Register: IC_CLR_SCL_STUCK_DET

Bits	Access	Name	Description & Reset Value
31:1	R	RSVD_CLR_SCL_STUCK_DET	CLR_SCL_STUCK_DET Reserved bits - Read Only Exists: Always Volatile: true
0	R	CLR_SCL_STUCK_DET	Read this register to clear the SCL_STUCT_AT_LOW interrupt (bit 15) of the IC_RAW_INTR_STAT register. Reset value: 0x0 Exists: Always Volatile: true

12.1.5.2.51 IC_DEVICE_ID

- Name: I2C Device-ID Register.

- Description: I2C Device-ID Register This Register contains the Device-ID of the component which includes 12-bits of Manufacturer name and 9-bits of part identification and 3 bits of die-version.
- Offset Address: 0xb8

Table 12-1-52 Fields for Register: IC_DEVICE_ID

Bits	Access	Name	Description & Reset Value
31:24	R	RSVD_IC_DEVICE_ID	IC_DEVICE_ID Reserved bits - Read Only Exists: Always
23:0	R	DEVICE-ID	Contains the Device-ID of the component assigned through the configuration parameter 'IC_DEVICE_ID_VALUE' Reset value: IC_DEVICE_ID_VALUE Exists: Always

12.1.5.2.52 IC_SMBUS_CLK_LOW_SEXT

- Name: SMBus Slave Clock Extend Timeout Register.
- Description: SMBus Slave Clock Extend Timeout Register This Register contains the Timeout value used to determine the Slave Clock Extend Timeout in one transfer (from START to STOP). This Register can be written only when the I2C controller is disabled, which corresponds to IC_ENABLE[0] being set to 0. This register is present only if configuration parameter IC_SMBUS is set to 1. This register is used to store the duration, measured in ic_clk cycles, used to detect the slave clock extend timeout if slave extends the clock (SCL) for the mentioned duration.
- Offset Address: 0xbc

Table 12-1-53 Fields for Register: IC_SMBUS_CLK_LOW_SEXT

Bits	Access	Name	Description & Reset Value
31:0	R/W	SMBUS_CLK_LOW_SEXT_TIMEOUT	This field is used to detect the Slave Clock Extend timeout (tLOW:SEXT) in master mode extended by the slave device in one message from the initial START to the STOP. The values in this register are in units of ic_clk period. Reset value: IC_SMBUS_CLK_LOW_SEXT_DEFAULT Exists: Always

12.1.5.2.53 IC_SMBUS_CLK_LOW_MEXT

- Name: SMBus Master Clock Extend Timeout Register.
- Description: SMBus Master Clock Extend Timeout Register This Register contains the Timeout value used to determine the Master Clock Extend Timeout in one byte of transfer. This Register can be written only when the I2C controller is disabled, which corresponds to IC_ENABLE[0] being set to 0. This register is present only if configuration parameter IC_SMBUS is set to 1. This register is used to store the duration, measured in ic_clk cycles, used to detect the Master clock extend timeout if Master extends the clock (SCL) for the mentioned duration.
- Offset Address: 0xc0

Table 12-1-54 Fields for Register: IC_SMBUS_CLK_LOW_MEXT

Bits	Access	Name	Description & Reset Value

Bits	Access	Name	Description & Reset Value
31:0	R/W	SMBUS_CLK_LOW_MEXT_TIME OUT	This field is used to detect the Master extend SMBus clock (SCLK) timeout defined from START-to-ACK, ACK-to-ACK, or ACK-to-STOP in Master mode. The values in this register are in units of ic_clk period. Reset value: IC_SMBUS_CLK_LOW_MEXT_DEFAULT Exists: Always

12.1.5.2.54 IC_SMBUS_THIGH_MAX_IDLE_COUNT

- Name: SMBus Master THigh MAX Bus-idle count Register.
- Description: SMBus Master THigh MAX Bus-idle count Register This register programs the Bus-idle time period used when a master has been dynamically added to the bus or when a master has generated a clock reset on the bus. This register is used to store the duration, measured in ic_clk cycles, used to detect the Bus Idle condition if SCL and SDA are held high for the mentioned duration. This register can be written only when the I2C controller is disabled, which corresponds to IC_ENABLE[0] being set to 0. This register is present only if configuration parameter IC_SMBUS is set to 1.
- Offset Address: 0xc4

Table 12-1-55 Fields for Register: IC_SMBUS_THIGH_MAX_IDLE_COUNT

Bits	Access	Name	Description & Reset Value
31:16	R	RSVD_SMBUS_THIGH_MAX_BU S_IDLE_CNT	SMBUS_THIGH_MAX_BUS_IDLE_CNT Reserved bits - Read Only Exists: Always
15:0	R/W	SMBUS_THIGH_MAX_BUS_IDL E_CNT	This field is used to set the required Bus-Idle time period used when a master has been dynamically added to the bus and may not have detected a state transition on the SMBCLK or SMBDAT lines. In this case, the master must wait long enough to ensure that a transfer is not currently in progress The values in this register are in units of ic_clk period. Reset value: IC_SMBUS_RST_IDLE_CNT_DEFAULT Exists: Always

12.1.5.2.55 IC_SMBUS_INTR_STAT

- Name: SMBUS Interrupt Status Register.
- Description: SMBUS Interrupt Status Register Each bit in this register has a corresponding mask bit in the IC_SMBUS_INTR_MASK register. These bits are cleared by writing the matching SMBus interrupt clear register(IC_CLR_SMBUS_INTR) bits. The unmasked raw versions of these bits are available in the IC_SMBUS_RAW_INTR_STAT register.
- Offset Address: 0xc8

Table 12-1-56 Fields for Register: IC_SMBUS_INTR_STAT

Bits	Access	Name	Description & Reset Value
31:11	R	RSVD_IC_SMBUS_INTR_STAT	IC_SMBUS_INTR_STAT Reserved bits - Read Only Exists: Always Volatile: true

Bits	Access	Name	Description & Reset Value
10	R	R_SMBUS_ALERT_DET	See IC_SMBUS_INTR_RAW_STATUS for a detailed description of R_SMBUS_ALERT_DET bit. Reset value: 0x0 Values: 0x1 (ACTIVE): SMBUS_ALERT_DET interrupt is active 0x0 (INACTIVE): SMBUS_ALERT_DET interrupt is inactive Exists: IC_SMBUS_SUSPEND_ALERT==1 Volatile: true
9	R	R_SMBUS_SUSPEND_DET	See IC_SMBUS_INTR_RAW_STATUS for a detailed description of R_SMBUS_SUSPEND_DET bit. Reset value: 0x0 Values: 0x1 (ACTIVE): SMBUS_SUSPEND_DET interrupt is active 0x0 (INACTIVE): SMBUS_SUSPEND_DET interrupt is inactive Exists: IC_SMBUS_SUSPEND_ALERT==1 Volatile: true
8	R	R_SLV_RX_PEC_NACK	See IC_SMBUS_INTR_RAW_STATUS for a detailed description of R_SLV_RX_PEC_NACK bit. Reset value: 0x0 Values: 0x1 (ACTIVE): SLV_RX_PEC_NACK interrupt is active 0x0 (INACTIVE): SLV_RX_PEC_NACK interrupt is inactive Exists: IC_SMBUS_ARP==1 Volatile: true
7	R	R_ARP_ASSGN_ADDR_CMD_DET	See IC_SMBUS_INTR_RAW_STATUS for a detailed description of R_ARP_ASSGN_ADDR_CMD_DET bit. Reset value: 0x0 Values: 0x1 (ACTIVE): ARP_ASSGN_ADDR_CMD_DET interrupt is active 0x0 (INACTIVE): ARP_ASSGN_ADDR_CMD_DET interrupt is inactive Exists: IC_SMBUS_ARP==1 Volatile: true

Bits	Access	Name	Description & Reset Value
6	R	R_ARP_GET_UDID_CMD_DET	See IC_SMBUS_INTR_RAW_STATUS for a detailed description of R_ARP_GET_UDID_CMD_DET bit. Reset value: 0x0 Values: 0x1 (ACTIVE): ARP_GET_UDID_CMD_DET interrupt is active 0x0 (INACTIVE): ARP_GET_UDID_CMD_DET interrupt is inactive Exists: IC_SMBUS_ARP==1 Volatile: true
5	R	R_ARP_RST_CMD_DET	See IC_SMBUS_INTR_RAW_STATUS for a detailed description of R_ARP_RST_CMD_DET bit. Reset value: 0x0 Values: 0x1 (ACTIVE): ARP_RST_CMD_DET interrupt is active 0x0 (INACTIVE): ARP_RST_CMD_DET interrupt is inactive Exists: IC_SMBUS_ARP==1 Volatile: true
4	R	R_ARP_PREPARE_CMD_DET	See IC_SMBUS_INTR_RAW_STATUS for a detailed description of R_ARP_PREPARE_CMD_DET bit. Reset value: 0x0 Values: 0x1 (ACTIVE): ARP_PREPARE_CMD_DET interrupt is active 0x0 (INACTIVE): ARP_PREPARE_CMD_DET interrupt is inactive Exists: IC_SMBUS_ARP==1 Volatile: true
3	R	R_HOST_NOTIFY_MST_DET	See IC_SMBUS_INTR_RAW_STATUS for a detailed description of R_HOST_NOTIFY_MST_DET bit. Reset value: 0x0 Values: 0x1 (ACTIVE): HOST_NOTIFY_MST_DET interrupt is active 0x0 (INACTIVE): HOST_NOTIFY_MST_DET interrupt is inactive Exists: IC_SMBUS==1 Volatile: true

Bits	Access	Name	Description & Reset Value
2	R	R_QUICK_CMD_DET	See IC_SMBUS_INTR_RAW_STATUS for a detailed description of R_QUICK_CMD_DET bit. Reset value: 0x0 Values: 0x1 (ACTIVE): QUICK_CMD_DET interrupt is active 0x0 (INACTIVE): QUICK_CMD_DET interrupt is inactive Exists: Always Volatile: true
1	R	R_MST_CLOCK_EXTND_TIMEOUT	See IC_SMBUS_INTR_RAW_STATUS for a detailed description of R_MST_CLOCK_EXTND_TIMEOUT bit. Reset value: 0x0 Values: 0x1 (ACTIVE): MST_CLOCK_EXTND_TIMEOUT interrupt is active 0x0 (INACTIVE): MST_CLOCK_EXTND_TIMEOUT interrupt is inactive Exists: Always Volatile: true
0	R	R_SLV_CLOCK_EXTND_TIMEOUT	See IC_SMBUS_INTR_RAW_STATUS for a detailed description of R_SLV_CLOCK_EXTND_TIMEOUT bit. Reset value: 0x0 Values: 0x1 (ACTIVE): SLV_CLOCK_EXTND_TIMEOUT interrupt is active 0x0 (INACTIVE): SLV_CLOCK_EXTND_TIMEOUT interrupt is inactive Exists: Always Volatile: true

12.1.5.2.56 IC_SMBUS_INTR_MASK

- Name: SMBus Interrupt Mask Register.
- Description: SMBus Interrupt Mask Register
- Offset Address: 0xcc

Table 12-1-57 Fields for Register: IC_SMBUS_INTR_MASK

Bits	Access	Name	Description & Reset Value
31:11	R	RSVD_IC_SMBUS_INTR_MASK	IC_SMBUS_INTR_MASK Reserved bits - Read Only Exists: Always

Bits	Access	Name	Description & Reset Value
10	R/W	M_SMBUS_ALERT_DET	This bit masks the R_SMBUS_ALERT_DET interrupt in IC_SMBUS_INTR_STAT register. Reset value: 0x1 Values: 0x1 (DISABLED): SMBUS_ALERT_DET interrupt is unmasked 0x0 (ENABLED): SMBUS_ALERT_DET interrupt is masked Exists: IC_SMBUS_SUSPEND_ALERT==1
9	R/W	M_SMBUS_SUSPEND_DET	This bit masks the R_SMBUS_SUSPEND_DET interrupt in IC_SMBUS_INTR_STAT register. Reset value: 0x1 Values: 0x1 (DISABLED): SMBUS_SUSPEND_DET interrupt is unmasked 0x0 (ENABLED): SMBUS_SUSPEND_DET interrupt is masked Exists: IC_SMBUS_SUSPEND_ALERT==1
8	R/W	M_SLV_RX_PEC_NACK	This bit masks the R_SLV_RX_PEC_NACK interrupt in IC_SMBUS_INTR_STAT register. Reset value: 0x1 Values: 0x1 (DISABLED): SLV_RX_PEC_NACK interrupt is unmasked 0x0 (ENABLED): SLV_RX_PEC_NACK interrupt is masked Exists: IC_SMBUS_ARP==1
7	R/W	M_ARP_ASSGN_ADDR_CMD_DET	This bit masks the R_ARP_ASSGN_ADDR_CMD_DET interrupt in IC_SMBUS_INTR_STAT register. Reset value: 0x1 Values: 0x1 (DISABLED): ARP_ASSGN_ADDR_CMD_DET interrupt is unmasked 0x0 (ENABLED): ARP_ASSGN_ADDR_CMD_DET interrupt is masked Exists: IC_SMBUS_ARP==1
6	R/W	M_ARP_GET_UDID_CMD_DET	This bit masks the R_ARP_GET_UDID_CMD_DET interrupt in IC_SMBUS_INTR_STAT register. Reset value: 0x1 Values: 0x1 (DISABLED): ARP_GET_UDID_CMD_DET interrupt is unmasked 0x0 (ENABLED): ARP_GET_UDID_CMD_DET interrupt is masked Exists: IC_SMBUS_ARP==1

Bits	Access	Name	Description & Reset Value
5	R/W	M_ARP_RST_CMD_DET	This bit masks the R_ARP_RST_CMD_DET interrupt in IC_SMBUS_INTR_STAT register. Reset value: 0x1 Values: 0x1 (DISABLED): ARP_RST_CMD_DET interrupt is unmasked 0x0 (ENABLED): ARP_RST_CMD_DET interrupt is masked Exists: IC_SMBUS_ARP==1
4	R/W	M_ARP_PREPARE_CMD_DET	This bit masks the R_ARP_PREPARE_CMD_DET interrupt in IC_SMBUS_INTR_STAT register. Reset value: 0x1 Values: 0x1 (DISABLED): ARP_PREPARE_CMD_DET interrupt is unmasked 0x0 (ENABLED): ARP_PREPARE_CMD_DET interrupt is masked Exists: IC_SMBUS_ARP==1
3	R/W	M_HOST_NOTIFY_MST_DET	This bit masks the R_HOST_NOTIFY_MST_DET interrupt in IC_SMBUS_INTR_STAT register. Reset value: 0x1 Values: 0x1 (DISABLED): HOST_NOTIFY_MST_DET interrupt is unmasked 0x0 (ENABLED): HOST_NOTIFY_MST_DET interrupt is masked Exists: IC_SMBUS==1
2	R/W	M_QUICK_CMD_DET	This bit masks the R_QUICK_CMD_DET interrupt in IC_SMBUS_INTR_STAT register. Reset value: 0x1 Values: 0x1 (DISABLED): QUICK_CMD_DET interrupt is unmasked 0x0 (ENABLED): QUICK_CMD_DET interrupt is masked Exists: Always
1	R/W	M_MST_CLOCK_EXTND_TIMEOUT	This bit masks the R_MST_CLOCK_EXTND_TIMEOUT interrupt in IC_SMBUS_INTR_STAT register. Reset value: 0x1 Values: 0x1 (DISABLED): MST_CLOCK_EXTND_TIMEOUT interrupt is unmasked 0x0 (ENABLED): MST_CLOCK_EXTND_TIMEOUT interrupt is masked Exists: Always

Bits	Access	Name	Description & Reset Value
0	R/W	M_SLV_CLOCK_EXTND_TIMEOUT	This bit masks the R_SLV_CLOCK_EXTND_TIMEOUT interrupt in IC_SMBUS_INTR_STAT register. Reset value: 0x1 Values: 0x1 (DISABLED): SLV_CLOCK_EXTND_TIMEOUT interrupt is unmasked 0x0 (ENABLED): SLV_CLOCK_EXTND_TIMEOUT interrupt is masked Exists: Always

12.1.5.2.57 IC_SMBUS_RAW_INTR_STAT

- Name: SMBus Raw Interrupt Status Register.
- Description: SMBus Raw Interrupt Status Register Unlike the IC_SMBUS_INTR_STAT register, these bits are not masked so they always show the true status of the I2C controller.
- Offset Address: 0xd0

Table 12-1-58 Fields for Register: IC_SMBUS_RAW_INTR_STAT

Bits	Access	Name	Description & Reset Value
31:11	R	RSVD_IC_SMBUS_RAW_INTR_STAT	IC_SMBUS_RAW_INTR_STAT Reserved bits - Read Only Exists: Always Volatile: true
10	R	SMBUS_ALERT_DET	Indicates whether a SMBALERT (ic_smbalert_in_n) signal is driven low by the slave. Reset value: 0x0 Values: 0x1 (ACTIVE): SMBUS Alert interrupt is active 0x0 (INACTIVE): SMBUS Alert interrupt is inactive Exists: IC_SMBUS_SUSPEND_ALERT==1 Volatile: true
9	R	SMBUS_SUSPEND_DET	Indicates whether a SMBSUS (ic_smbsus_in_n) signal is driven low by the Host. Reset value: 0x0 Values: 0x1 (ACTIVE): SMBUS System Suspended interrupt is active 0x0 (INACTIVE): SMBUS System Suspended interrupt is inactive Exists: IC_SMBUS_SUSPEND_ALERT==1 Volatile: true
8	R	SLV_RX_PEC_NACK	Indicates whether a NACK has been sent due to PEC mismatch while working as ARP slave. Reset value: 0x0 Values: 0x1 (ACTIVE): SLV_RX_PEC_NACK interrupt is active 0x0 (INACTIVE): SLV_RX_PEC_NACK interrupt is inactive Exists: IC_SMBUS_ARP==1 Volatile: true

Bits	Access	Name	Description & Reset Value
7	R	ARP_ASSGN_ADDR_CMD_DET	Indicates whether an Assign Address ARP command has been received. Reset value: 0x0 Values: 0x1 (ACTIVE): ARP_ASSGN_ADDR_CMD_DET interrupt is active 0x0 (INACTIVE): ARP_ASSGN_ADDR_CMD_DET interrupt is inactive Exists: IC_SMBUS_ARP==1 Volatile: true
6	R	ARP_GET_UDID_CMD_DET	Indicates whether a Get UDID ARP command has been received. Reset value: 0x0 Values: 0x1 (ACTIVE): ARP_GET_UDID_CMD_DET interrupt is active 0x0 (INACTIVE): ARP_GET_UDID_CMD_DET interrupt is inactive Exists: IC_SMBUS_ARP==1 Volatile: true
5	R	ARP_RST_CMD_DET	Indicates whether a General or Directed Reset ARP command has been received. Reset value: 0x0 Values: 0x1 (ACTIVE): ARP_RST_CMD_DET interrupt is active 0x0 (INACTIVE): ARP_RST_CMD_DET interrupt is inactive Exists: IC_SMBUS_ARP==1 Volatile: true
4	R	ARP_PREPARE_CMD_DET	Indicates whether a prepare to ARP command has been received. Reset value: 0x0 Values: 0x1 (ACTIVE): ARP_PREPARE_CMD_DET interrupt is active 0x0 (INACTIVE): ARP_PREPARE_CMD_DET interrupt is inactive Exists: IC_SMBUS_ARP==1 Volatile: true

Bits	Access	Name	Description & Reset Value
3	R	HOST_NTFY_MST_DET	Indicates whether a Notify ARP Master ARP command has been received. Reset value: 0x0 Values: 0x1 (ACTIVE): HOST_NTFY_MST_DET interrupt is active 0x0 (INACTIVE): HOST_NTFY_MST_DET interrupt is inactive Exists: IC_SMBUS==1 Volatile: true
2	R	QUICK_CMD_DET	Indicates whether a Quick command has been received on the SMBus interface regardless of whether DW_apb_i2c is operating in slave or master mode. Enabled only when IC_SMBUS=1 is set to 1. Reset value: 0x0 Values: 0x1 (ACTIVE): Quick Command interrupt is active 0x0 (INACTIVE): Quick Command interrupt is inactive Exists: Always Volatile: true
1	R	MST_CLOCK_EXTND_TIMEOUT	Indicates whether the Master device transaction (START-to- ACK, ACK-to-ACK, or ACK-to-STOP) from START to STOP exceeds IC_SMBUS_CLOCK_LOW_MEXT time with in each byte of message. This bit is enabled only when: IC_SMBUS=1 IC_CON[0]=1 IC_EMPTYFIFO_HOLD_MASTER_EN=1 or IC_RX_FULL_HLD_BUS_EN=1 Reset value: 0x0 Values: 0x1 (ACTIVE): Master Clock Extend Timeout interrupt is active 0x0 (INACTIVE): Master Clock Extend Timeout interrupt is inactive Exists: Always Volatile: true

Bits	Access	Name	Description & Reset Value
0	R	SLV_CLOCK_EXTND_TIMEOUT	Indicates whether the transaction from Slave (i.e from START to STOP) exceeds IC_SMBUS_CLK_LOW_SEXT time. This bit is enabled only when: IC_SMBUS=1 IC_CON[0]=1 Reset value: 0x0 Values: 0x1 (ACTIVE): Slave Clock Extend Timeout interrupt is active 0x0 (INACTIVE): Slave Clock Extend Timeout interrupt is inactive Exists: Always Volatile: true

12.1.5.2.58 IC_CLR_SMBUS_INTR

- Name: SMBus Clear Interrupt Register.
- Description: SMBus Clear Interrupt Register
- Offset Address: 0xd4

Table 12-1-59 Fields for Register: IC_CLR_SMBUS_INTR

Bits	Access	Name	Description & Reset Value
31:11	W	RSVD_IC_CLR_SMBUS_INTR	IC_CLR_SMBUS_INTR Reserved bits - Read Only Exists: Always
10	W	CLR_SMBUS_ALERT_DET	Write this register bit to clear the SMBUS_ALERT_DET interrupt (bit 10) of the IC_SMBUS_RAW_INTR_STAT register. Reset value: 0x0 Exists: IC_SMBUS_SUSPEND_ALERT==1
9	W	CLR_SMBUS_SUSPEND_DET	Write this register bit to clear the SMBUS_SUSPEND_DET interrupt (bit 9) of the IC_SMBUS_RAW_INTR_STAT register. Reset value: 0x0 Exists: IC_SMBUS_SUSPEND_ALERT==1
8	W	CLR_SLV_RX_PEC_NACK	Write this register bit to clear the SLV_RX_PEC_NACK interrupt (bit 8) of the IC_SMBUS_RAW_INTR_STAT register. Reset value: 0x0 Exists: IC_SMBUS_ARP==1
7	W	CLR_ARP_ASSGN_ADDR_CMD_DET	Write this register bit to clear the ARP_ASSGN_ADDR_CMD_DET interrupt (bit 7) of the IC_SMBUS_RAW_INTR_STAT register. Reset value: 0x0 Exists: IC_SMBUS_ARP==1

Bits	Access	Name	Description & Reset Value
6	W	CLR_ARP_GET_UDID_CMD_DET	Write this register bit to clear the ARP_GET_UDID_CMD_DET interrupt (bit 6) of the IC_SMBUS_RAW_INTR_STAT register. Reset value: 0x0 Exists: IC_SMBUS_ARP==1
5	W	CLR_ARP_RST_CMD_DET	Write this register bit to clear the ARP_RST_CMD_DET interrupt (bit 5) of the IC_SMBUS_RAW_INTR_STAT register. Reset value: 0x0 Exists: IC_SMBUS_ARP==1
4	W	CLR_ARP_PREPARE_CMD_DET	Write this register bit to clear the ARP_PREPARE_CMD_DET interrupt (bit 4) of the IC_SMBUS_RAW_INTR_STAT register. Reset value: 0x0 Exists: IC_SMBUS_ARP==1
3	W	CLR_HOST_NOTIFY_MST_DET	Write this register bit to clear the HOST_NOTIFY_MST_DET interrupt (bit 3) of the IC_SMBUS_RAW_INTR_STAT register. Reset value: 0x0 Exists: IC_SMBUS==1
2	W	CLR_QUICK_CMD_DET	Write this register bit to clear the QUICK_CMD_DET interrupt (bit 2) of the IC_SMBUS_RAW_INTR_STAT register. Reset value: 0x0 Exists: Always
1	W	CLR_MST_CLOCK_EXTND_TIME EOUT	Write this register bit to clear the MST_CLOCK_EXTND_TIMEOUT interrupt (bit 1) of the IC_SMBUS_RAW_INTR_STAT register. Reset value: 0x0 Exists: Always
0	W	CLR_SLV_CLOCK_EXTND_TIME OUT	Write this register bit to clear the SLV_CLOCK_EXTND_TIMEOUT interrupt (bit 0) of the IC_SMBUS_RAW_INTR_STAT register. Reset value: 0x0 Exists: Always

12.1.5.2.59 IC_OPTIONAL_SAR

- Name: I2C Optional Slave Address Register.
- Description: I2C Optional Slave Address Register Optional Slave address for I2C in SMBus Mode. A same restriction as IC_SAR applies on IC_OPTIONAL_SAR.
- Offset Address: 0xd8

Table 12-1-60 Fields for Register: IC_OPTIONAL_SAR

Bits	Access	Name	Description & Reset Value

Bits	Access	Name	Description & Reset Value
31:7	R	RSVD_IC_OPTIONAL_SAR	IC_OPTIONAL_SAR Reserved bits - Read Only Exists: Always
6:0	R/W	OPTIONAL_SAR	Optional Slave address for DW_apb_i2c when operating as a slave in SMBus Mode. Reset Value: IC_OPTIONAL_SAR_DEFAULT Exists: Always

12.1.5.2.60 IC_SMBUS_UDID_LSB

- Name: SMBUS ARP UDID LSB Register.
- Description: SMBUS ARP UDID LSB Register This Register can be written only when the I2C controller is disabled, which corresponds to IC_ENABLE[0] being set to 0. This register is present only if configuration parameter IC_SMBUS_ARP is set to 1. This register is used to store the LSB 32 bit value of Slave UDID register used in Address Resolution Protocol of SMBus.
- Offset Address: 0xdc

Table 12-1-61 Fields for Register: IC_SMBUS_UDID_LSB

Bits	Access	Name	Description & Reset Value
31:0	R/W	SMBUS_UDID_LSB	This field is used to store the LSB 32 bit value of slave unique device identifier used in Address Resolution Protocol. Reset Value: IC_SMBUS_UDID_LSB_DEFAULT Exists: Always

12.1.5.2.61 IC_COMP_PARAM_1

- Name: Component Parameter Register 1.
- Description: Component Parameter Register 1 Note This is a constant read-only register that contains encoded information about the component's parameter settings. The reset value depends on coreConsultant parameter(s).
- Offset Address: 0xf4

Table 12-1-62 Fields for Register: IC_COMP_PARAM_1

Bits	Access	Name	Description & Reset Value
31:24	R	RSVD_IC_COMP_PARAM_1	IC_COMP_PARAM_1 Reserved bits - Read Only Exists: Always
23:16	R	TX_BUFFER_DEPTH	The value of this register is derived from the IC_TX_BUFFER_DEPTH coreConsultant parameter. 0x00 = Reserved 0x01 = 2 0x02 = 3 ... 0xFF = 256 Exists: Always

Bits	Access	Name	Description & Reset Value
15:8	R	RX_BUFFER_DEPTH	The value of this register is derived from the IC_RX_BUFFER_DEPTH coreConsultant parameter. 0x00: Reserved 0x01: 2 0x02: 3 ... 0xFF: 256 Exists: Always
7	R	ADD_ENCODED_PARAMS	The value of this register is derived from the IC_ADD_ENCODED_PARAMS coreConsultant parameter. Reading 1 in this bit means that the capability of reading these encoded parameters via software has been included. Otherwise, the entire register is 0 regardless of the setting of any other parameters that are encoded in the bits. Values: 0x1 (ENABLED): Enables capability of reading encoded parameters 0x0 (DISABLED): Disables capability of reading encoded parameters Exists: Always
6	R	HAS_DMA	The value of this register is derived from the IC_HAS_DMA coreConsultant parameter. Values: 0x1 (ENABLED): DMA handshaking signals are enabled 0x0 (DISABLED): DMA handshaking signals are disabled Exists: Always
5	R	INTR_IO	The value of this register is derived from the IC_INTR_IO coreConsultant parameter. Values: 0x1 (COMBINED): COMBINED Interrupt outputs 0x0 (INDIVIDUAL): INDIVIDUAL Interrupt outputs Exists: Always
4	R	HC_COUNT_VALUES	The value of this register is derived from the IC_HC_COUNT_VALUES coreConsultant parameter. Values: 0x1 (ENABLED): Hard code the count values for each mode. 0x0 (DISABLED): Programmable count values for each mode. Exists: Always

Bits	Access	Name	Description & Reset Value
3:2	R	MAX_SPEED_MODE	The value of this register is derived from the IC_MAX_SPEED_MODE coreConsultant parameter. 0x0: Reserved 0x1: Standard 0x2: Fast 0x3: High Values: 0x1 (STANDARD): IC MAX SPEED is STANDARD MODE 0x2 (FAST): IC MAX SPEED is FAST MODE 0x3 (HIGH): IC MAX SPEED is HIGH MODE Exists: IC_ULTRA_FAST_MODE==0
1:0	R	APB_DATA_WIDTH	The value of this register is derived from the APB_DATA_WIDTH coreConsultant parameter. Values: 0x0 (APB_08BITS): APB data bus width is 08 bits 0x1 (APB_16BITS): APB data bus width is 16 bits 0x2 (APB_32BITS): APB data bus width is 32 bits 0x3 (RESERVED): Reserved bits Exists: Always

12.1.5.2.62 IC_COMP_VERSION

- Name: I2C Component Version Register.
- Description: I2C Component Version Register
- Offset Address: 0xf8

Table 12-1-63 Fields for Register: IC_COMP_VERSION

Bits	Access	Name	Description & Reset Value
31:0	R	IC_COMP_VERSION	Specific values for this register are described in the Releases Table in the DW_apb_i2c Release Notes Exists: Always

12.1.5.2.63 IC_COMP_TYPE

- Name: I2C Component Type Register.
- Description: I2C Component Type Register
- Offset Address: 0xfc

Table 12-1-64 Fields for Register: IC_COMP_TYPE

Bits	Access	Name	Description & Reset Value
31:0	R	IC_COMP_TYPE	Designware Component Type number = 0x44_57_01_40. This assigned unique hex value is constant and is derived from the two ASCII letters 'DW' followed by a 16-bit unsigned number. Exists: Always

12.1.6 Interface Description

Table 12-1-65 I2C Interface Description

Module pin	Direction	PAD name	IOMUX
i2c0_ic_clk_in_a	I/O	PAD_IO_32;PAD_IO_60;PAD_IO_48	IO_32_sel == 3'd1;IO_60_sel == 3'd2;IO_48_sel == 3'd3
i2c0_ic_data_in_a	I/O	PAD_IO_33;PAD_IO_61;PAD_IO_49	IO_33_sel == 3'd1;IO_61_sel == 3'd2;IO_49_sel == 3'd3
i2c1_ic_clk_in_a	I/O	PAD_IO_34;PAD_IO_40;PAD_IO_9	IO_34_sel == 3'd1;IO_40_sel == 3'd2;IO_9_sel == 3'd3
i2c1_ic_data_in_a	I/O	PAD_IO_35;PAD_IO_41;PAD_IO_10	IO_35_sel == 3'd1;IO_41_sel == 3'd2;IO_10_sel == 3'd3
i2c2_ic_clk_in_a	I/O	PAD_IO_50;PAD_IO_11	IO_50_sel == 3'd2;IO_11_sel == 3'd3
i2c2_ic_data_in_a	I/O	PAD_IO_51;PAD_IO_12	IO_51_sel == 3'd2;IO_12_sel == 3'd3
i2c3_ic_clk_in_a	I/O	PAD_IO_36;PAD_IO_44;PAD_IO_52	IO_36_sel == 3'd1;IO_44_sel == 3'd2;IO_52_sel == 3'd3
i2c3_ic_data_in_a	I/O	PAD_IO_37;PAD_IO_45;PAD_IO_53	IO_37_sel == 3'd1;IO_45_sel == 3'd2;IO_53_sel == 3'd3
i2c4_ic_clk_in_a	I/O	PAD_IO_7;PAD_IO_46	IO_7_sel == 3'd2;IO_46_sel == 3'd3
i2c4_ic_data_in_a	I/O	PAD_IO_8;PAD_IO_47	IO_8_sel == 3'd2;IO_47_sel == 3'd3

12.1.7 Application Notes

12.1.7.1 Programming Example

The flow diagram in Figure 12-1-15 shows a programming example for the I2C controller Master.

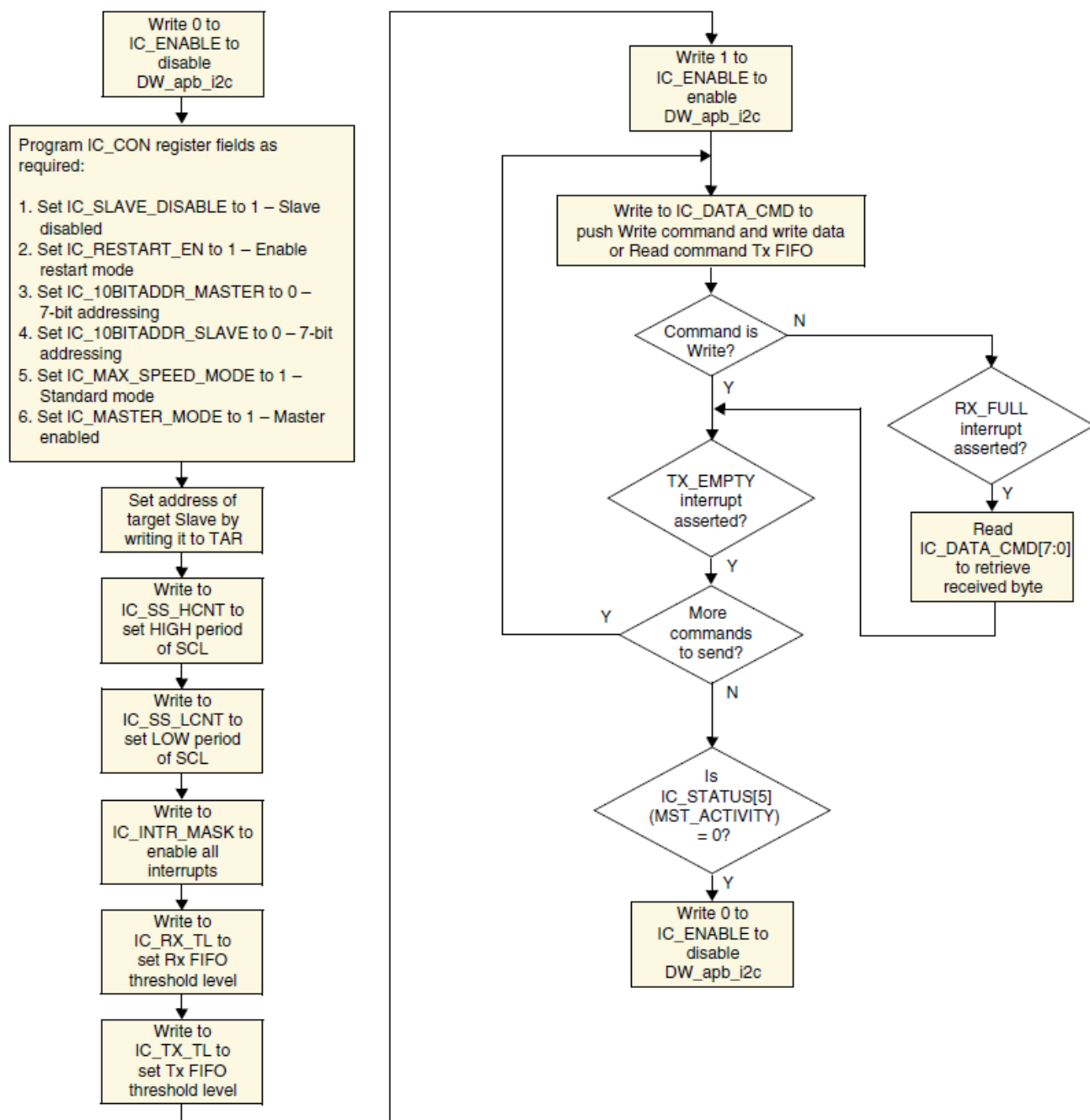


Figure 12-1-15 Flowchart for I2C controller Master

The flow diagram in Figure 12-1-16 shows a programming example for the I2C controller master in standard mode, fast mode, or fast mode plus with 7-bit addressing.

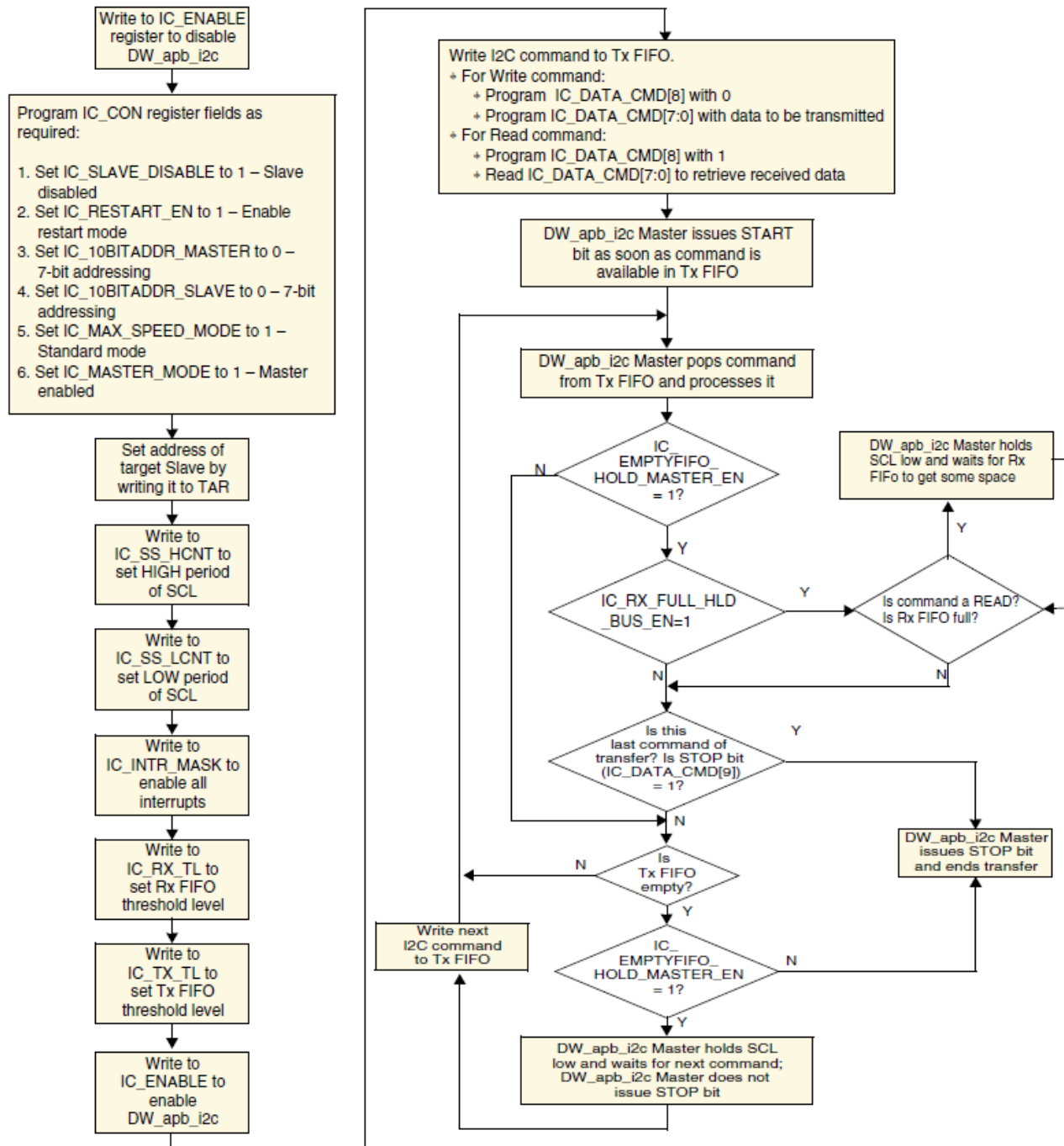


Figure 12-1-16 Flowchart for I2C controller Master in Standard Mode, Fast Mode, or Fast Mode Plus

The flow diagram in Figure 12-1-17 shows a programming example for I2C controller as master with TAR address update. This flow shows how the MST_ON_HOLD interrupt is used when the software needs information from the hardware to safely update the TAR address.

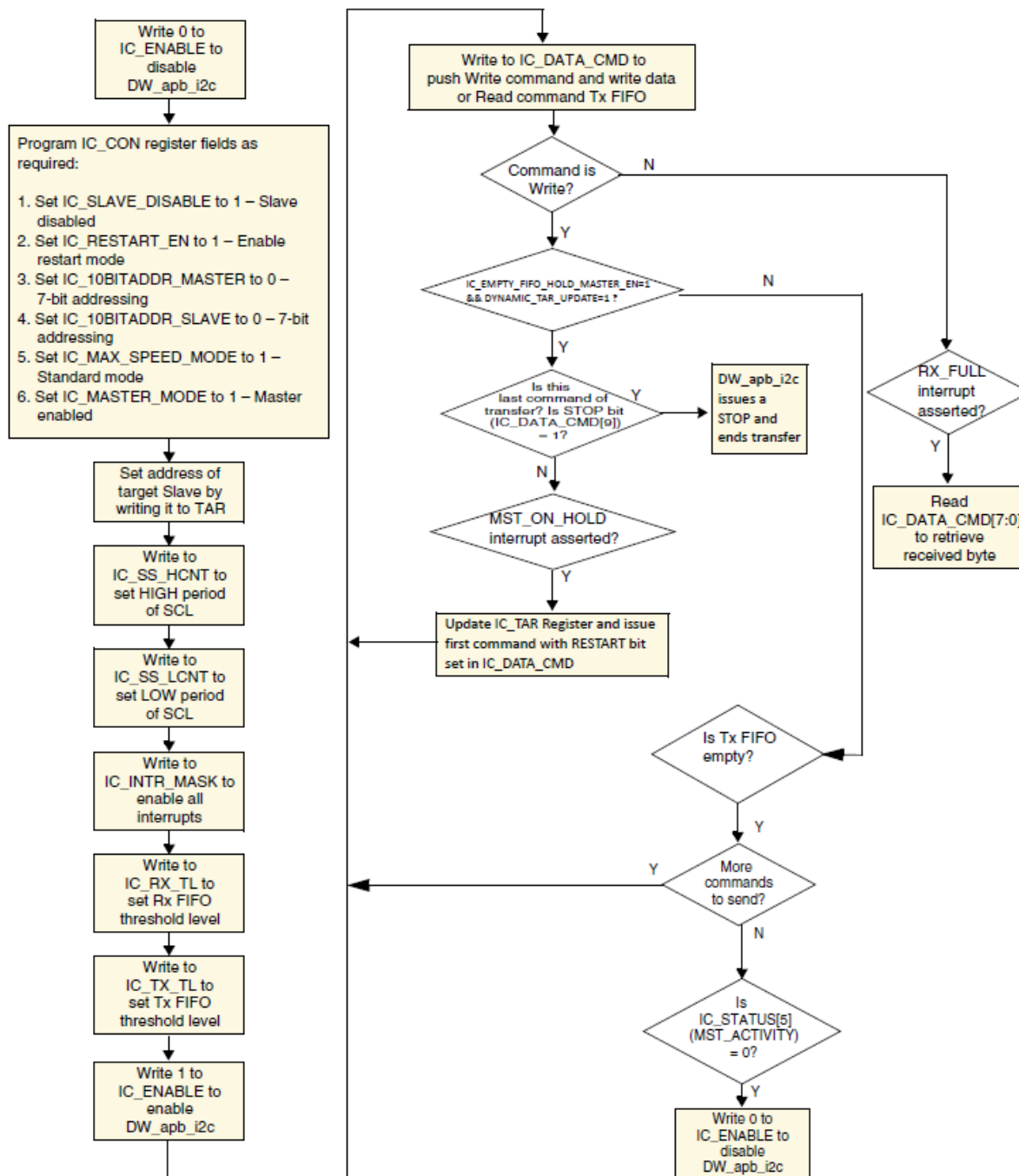


Figure 12-1-17 Flowchart for I2C controller Master with TAR Address Update

12.1.7.2 Programming Flow for SCL and SDA Bus Recovery

The flow diagram in Figure 12-1-18 shows a programming example for SCL and SDA bus recovery.

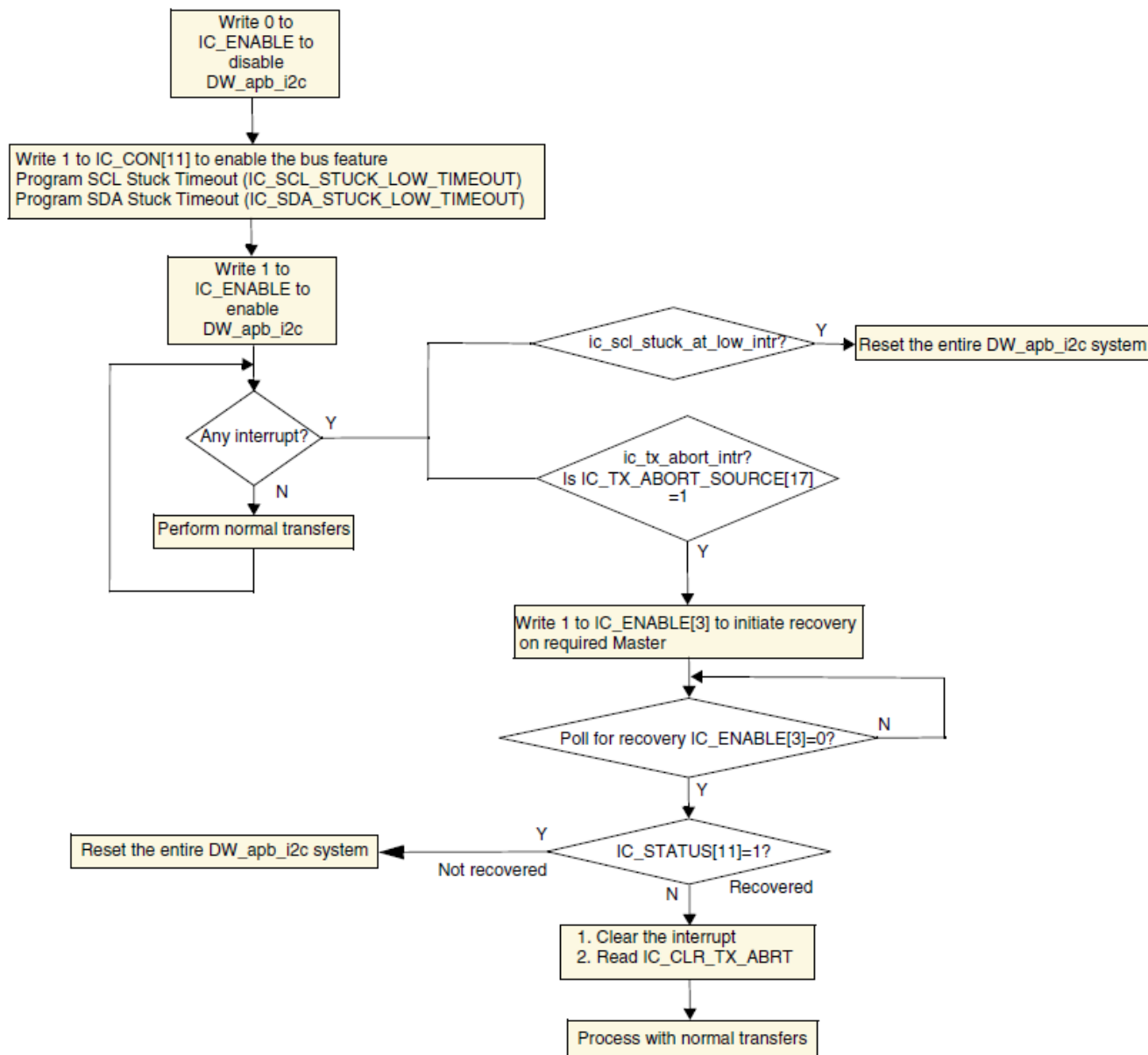


Figure 12-1-18 Flowchart for SCL and SDA Bus Recovery

12.1.7.3 Programming Flow for Reading the Device ID

Figure 12-1-19 shows a programming flow in the master to initiate a Device ID read.

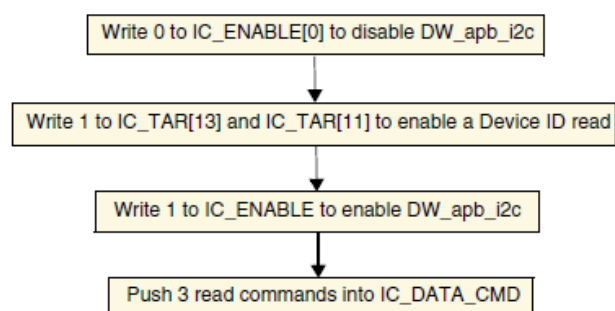


Figure 12-1-19 Flowchart for Reading a Device ID

As the Device ID consists of 3 bytes, the user must issue 3 read commands in IC_DATA_CMD register. One read command populates one byte of Device ID in RX FIFO. If more than 3 commands are issued, the Device ID will roll back.

12.1.7.4 Programming Flow for SMBUS Timeout in Master Mode

Figure 12-1-20 shows a programming flow for SMBus timeout in master mode.

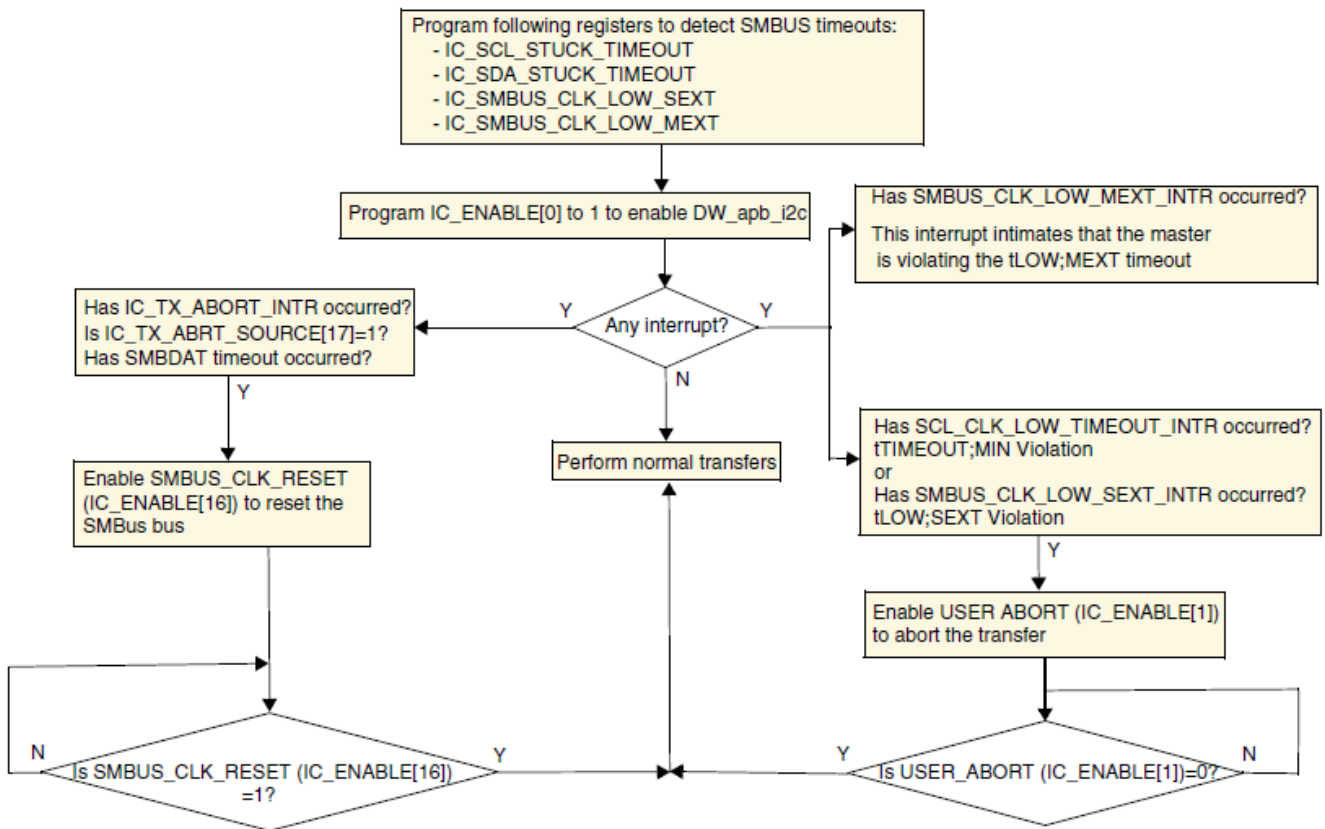


Figure 12-1-20 SMBUS Timeout Programming Flow in Master Mode

12.1.7.5 ARP Master Programming Flow

Figure 12-1-21 shows the programming flow for an ARP master.

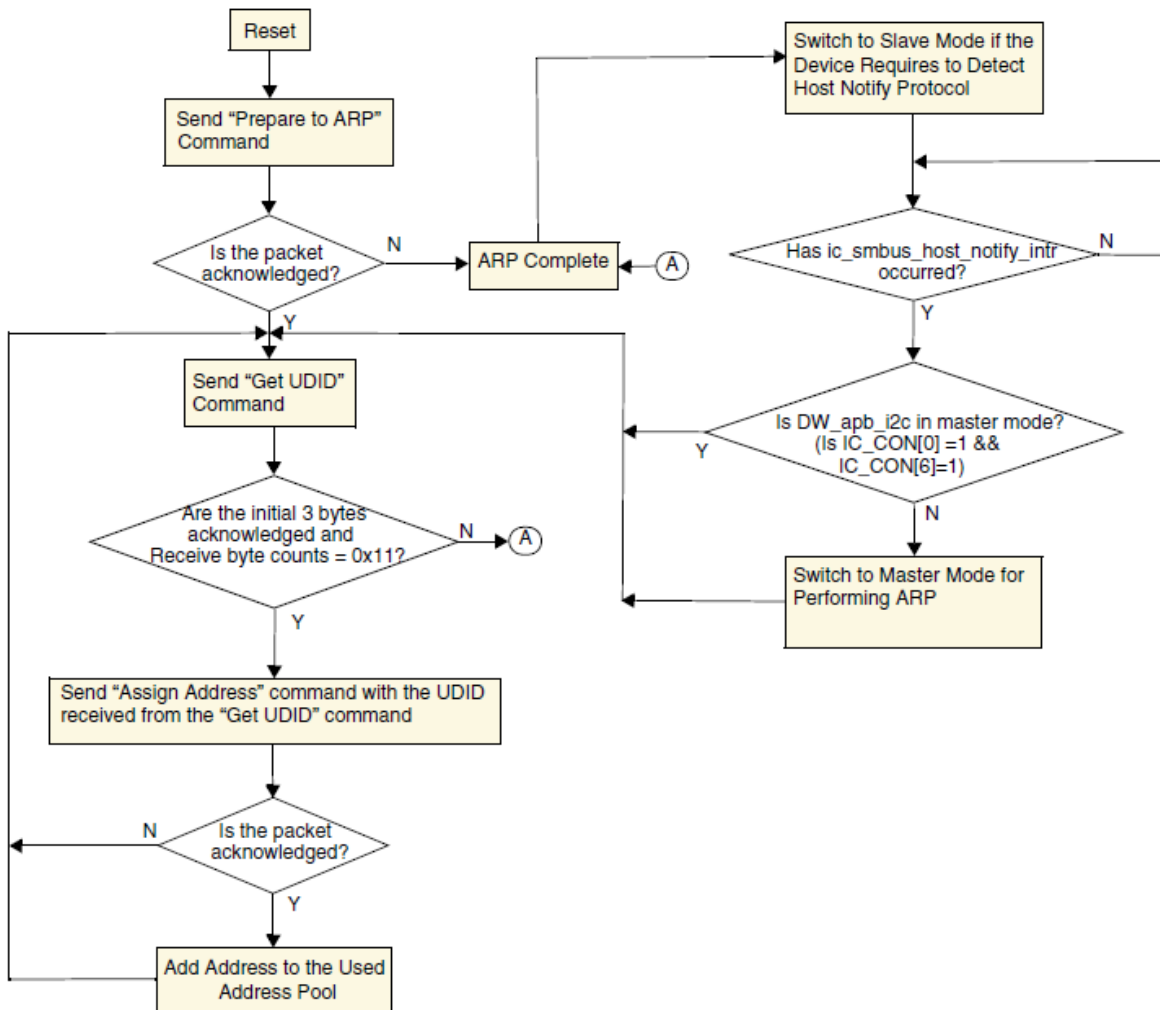


Figure 12-1-21 ARP Master Programming Flow

12.2 UART

12.2.1 Overview

The universal asynchronous receiver transmitter (UART) is an asynchronous serial communication interface. It performs serial-to-parallel conversion on the data received from peripherals and transmits the converted data to the internal bus. It also performs parallel-to-serial conversion on the data that is transmitted to peripherals. The UART is mainly used to interconnect chip with the UART of an external chip so that the two chips can communicate with each other.

The chip provides five UARTs:

- UART0/4: 2-wire UART for debugging
- UART1/2/3: 4-wire UART

12.2.2 Feature

The UART has the following features:

- 9-bit serial data support
- False start bit detection
- Programmable fractional baud rate support
- Multi-drop RS485 interface support
- APB data bus widths of 32

- Additional DMA interface signals for compatibility with DesignWare DMA interface, DMA interface signal active low
- Transmit and receive FIFO depths of 32, Supports 32x32-bit transmit first-in, first-out (FIFO) and 32x32-bit RX FIFO, Internal FIFO (RAM)
- Use of two clocks—pclk and sclk
- IrDA 1.0 SIR mode support with up to 115.2 Kbaud data rate and a pulse duration (width) as specified in the IrDA physical layer specification:
 - $\text{width} = 3/16 \times \text{bit period}$
- IrDA 1.0 SIR low-power reception capabilities
- Baud clock reference output signal
- Clock gate enable output(s) used to indicate that the TX and RX pipeline is clear (no data) and no activity has occurred for more than one character time, so that clocks can be gated
- FIFO access mode—for FIFO testing—enabling the master to write to the receive FIFO and read from the transmit FIFO
- Additional FIFO status registers
- Shadow registers to reduce software overhead and also include a software programmable reset
- Auto Flow Control mode, as specified in the 16750 standard
- Loopback mode that enables greater testing of Modem Control and Auto Flow Control features (Loopback support in IrDA SIR mode is available)
- Transmitter Holding Register Empty (THRE) interrupt mode
- Busy functionality
- Ability to set some configuration parameters during instantiation
- Configuration identification registers present
- Functionality based on the 16550 industry standard
 - Programmable character properties, such as:
 - Number of data bits per character (5-8)
 - Optional parity bit (with odd, even select or Stick Parity)
 - Number of stop bits (1, 1.5 or 2)
 - Line break generation and detection
 - DMA signaling with two programmable modes
 - Prioritized interrupt identification
- Programmable FIFO enable/disable
- Programmable serial data baud rate as calculated by the following:
 - $\text{baud rate} = (\text{serial clock frequency}) / (16 \times \text{divisor})$
- External read enable signal for RAM wake-up when using external RAMs
- Modem and status lines are independently controlled
- Separate system resets for each clock domain to prevent metastability

12.2.3 Block Diagram

The UART is modeled after the industry-standard 16550. However, the register address space is relocated to 32-bit data boundaries for APB bus implementation. The UART is used for serial communication with:

- Peripherals
- Modems (data carrier equipment, DCE)
- Data sets

Data is written from a master (CPU) over the APB bus to the UART, and it is converted to serial form and transmitted to the destination device. Serial data is also received by the UART and stored for the master (CPU) to read back. The UART contains registers that control:

- Character length
- Baud rate
- Parity generation/checking
- Interrupt generation

Although there is only one interrupt output signal (intr) from the UART, there are several prioritized interrupt types that can be responsible for its assertion. Each of the interrupt types can be separately enabled or disabled by the control registers.

The following describe various functionalities of the UART:

- Transmit and receive data FIFOs – To reduce the time demand placed on the master by the UART, the FIFOs are available to buffer transmit and receive data. The master does not have to access the UART each time a single byte of data is received.
- DMA controller interface – The UART can interface with a DMA controller through external signals (dma_tx_req_n and dma_rx_req_n) in order to indicate when data is ready to be read or when the transmit FIFO is empty.
- Asynchronous clock support – To solve problems surrounding CPU data synchronization in relation to the required serial baud clock requirements, a separate serial data clock can be used. Full handshaking and level-synchronization guarantees all data crossing between the two clock domains.
- Auto flow control – The UART uses a 16750-compatible Auto Flow Control Mode to increase system efficiency and decrease software load. When FIFOs and the Auto Flow Control are used and enabled, the request-to-send (rts_n) output and clear-to-send (cts_n) input automatically control serial data flow. Supply by UART1/2/3.
- RS485 interface Support - For integration into systems for which an RS485 interface is required, the UART can be configured for a software-programmable RS485 mode. If this mode is not selected, only the UART (RS232 standard) serial data format is available.
- Programmable Transmit Holding Register Empty (THRE) interrupt – The UART uses a Programmable Transmitter Holding Register Empty (THRE) Interrupt Mode to increase system performance. When FIFOs and the THRE Mode are selected and enabled, THRE Interrupts are active at or below a programmed TX FIFO threshold level. Additionally, the Line Status THRE switches from indicating TX FIFO empty to TX FIFO full, which allows software to set a threshold that keeps the transmitter FIFO from running empty whenever there is data to transmit.
- Serial infrared support – For integration in systems where Infrared SIR serial data format is required, the UART can be configured for a software-programmable IrDA SIR Mode. If this mode is not selected, only the UART (RS232 standard) serial data format is available.
- Increase built-in diagnostic capabilities – To increase the built-in diagnostic capabilities of the UART, the Modem Control Loopback Mode has been extended. Modem Status bits actually reflect Modem Control Register deltas, as well as the bits themselves. Additionally, when FIFOs and Auto Flow Control Mode are selected and enabled, the Modem Control RTS is internally looped back to the CTS in order to control the transmitter, which allows local testing of the Auto CTS mode. Furthermore, the controllability of rts_n through the receiver FIFO threshold can be observed using the RTS Modem Status bit, which allows local verification of the Auto RTS mode.

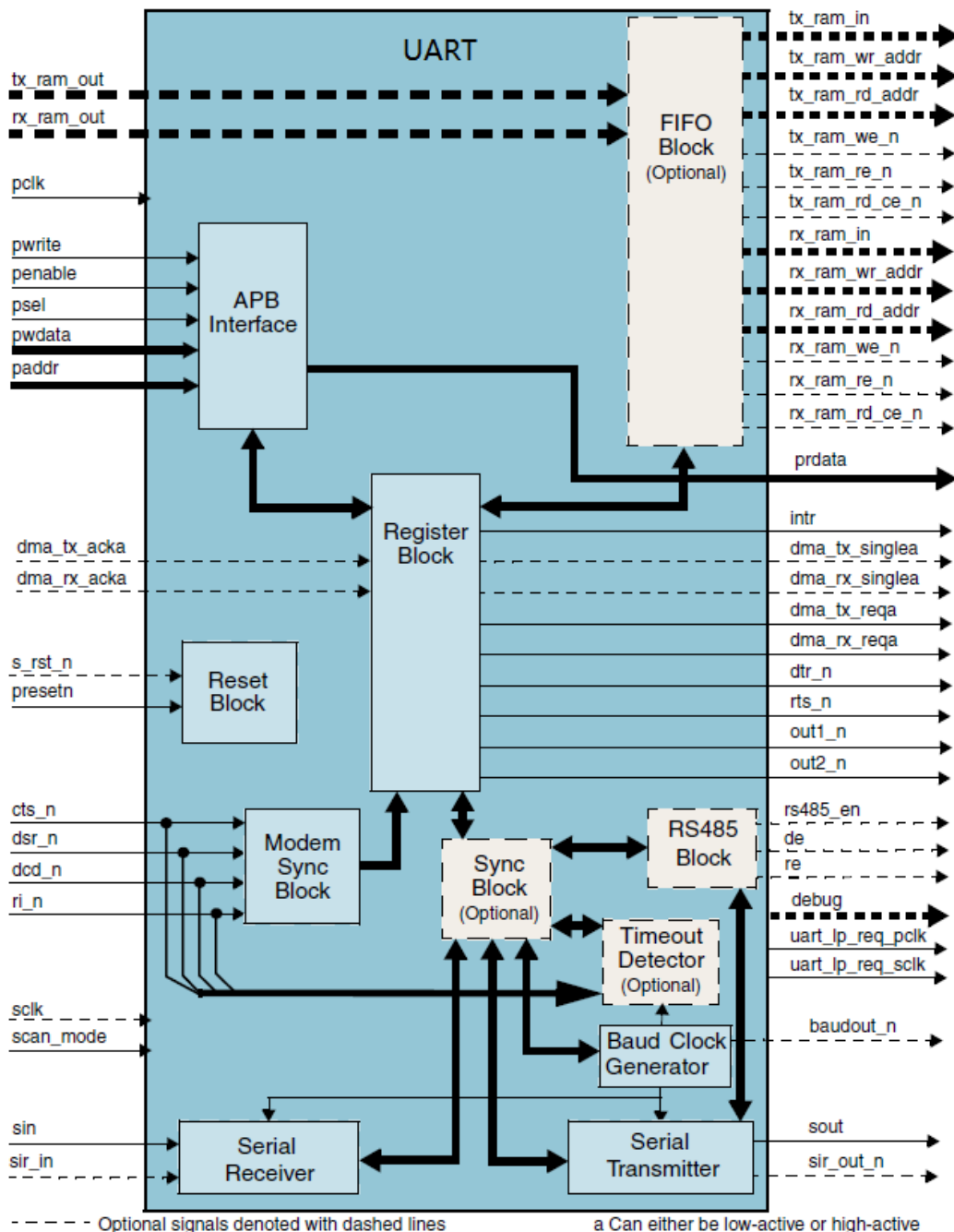


Figure 12-2-1 UART Functional Block Diagram

The following list describes each of the major blocks shown in Figure 12-2-1:

- Reset block – resets clock domains.
- APB slave interface – connects to APB bus.
- Register block – responsible for the main UART functionality including control, status and interrupt generation.
- Modem Synchronization block – synchronizes the modem input signal.
- FIFO block (optional) – responsible for FIFO control and storage—when using internal RAM—or optionally signaling to control external RAM.
- Synchronization block (optional) – implemented when the peripheral is configured to have a separate serial data clock (i.e. two clock implementation).
- Timeout Detector block (optional) – indicates the absence of character data movement in the receiver FIFO within a given time period; this is used to generate character timeout interrupts when enabled.
This block can also have optional clock gate enable outputs—uart_lp_req_pclk for single clock implementations or uart_lp_req_pclk and uart_lp_req_sclk for two clock implementations—in order to indicate:

- TX and RX pipeline is clear; that is, there is no data
 - No activity has occurred
 - Modem control input signals have not changed within a given time period
- Baud Clock Generator – produces the transmitter and receiver baud clock along with the output reference clock signal (baudout_n).
- Serial Transmitter – converts the parallel data—written to the UART—into serial form and adds all additional bits, as specified by the control register, for transmission. These serial data, referred to as a character, can exit the block in two formats:
 - Serial UART
 - IrDA 1.0 SIR
- Serial Receiver – converts the serial data character—specified by the control register—received in either the UART or IrDA 1.0 SIR format to parallel form. This block controls:
 - Parity error detection
 - Framing error detection
 - Line break detection
- RS485 block (optional) – implemented when the peripheral is configured to have an RS485 interface, responsible for the generation of driver enable (de) and receiver enable (re) signals required by the RS485 Transceiver.

12.2.4 Function Description

12.2.4.1 UART (RS232) Serial Protocol

Because the serial communication between the UART and a selected device is asynchronous, additional bits (start and stop) are added to the serial data to indicate the beginning and end. Utilizing these bits allows two devices to be synchronized. This structure of serial data—accompanied by start and stop bits—is referred to as a character, as shown in Figure 12-2-2.

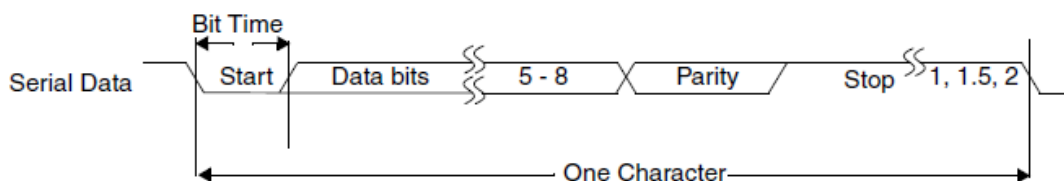


Figure 12-2-2 Serial Data Format

An additional parity bit can be added to the serial character. This bit appears after the last data bit and before the stop bit(s) in the character structure in order to provide the UART with the ability to perform simple error checking on the received data.

The UART Line Control Register (LCR) is used to control the serial character characteristics. The individual bits of the data word are sent after the start bit, starting with the least significant bit (LSB). These are followed by the optional parity bit, followed by the stop bit(s), which can be 1, 1.5, or 2.

All the bits in the transmission are transmitted for exactly the same time duration; the exception to this is the half-stop bit when 1.5 stop bits are used. This duration is referred to as a Bit Period or Bit Time; one Bit Time equals sixteen baud clocks.

To ensure stability on the line, the receiver samples the serial input data at approximately the midpoint of the Bit Time once the start bit has been detected. Because the exact number of baud clocks is known for which each bit was transmitted, calculating the midpoint for sampling is not difficult; that is, every sixteen baud clocks after the midpoint sample of the start bit.

Together with serial input debouncing, this sampling helps to avoid the detection of false start bits. Short glitches are filtered out by debouncing, and no transition is detected on the line. If a glitch is wide enough to avoid filtering by debouncing, a falling edge is detected. However, a start bit is detected only if the line is again sampled low after half a

bit time has elapsed.

Figure 12-2-3 shows the sampling points of the first two bits in a serial character.

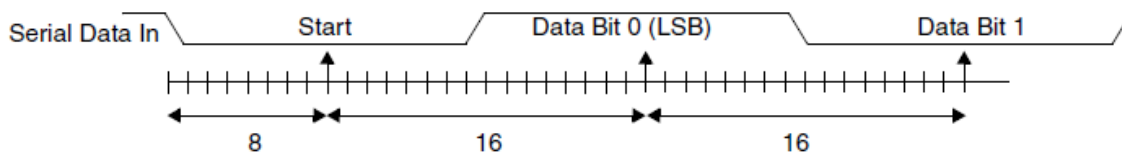


Figure 12-2-3 Receiver Serial Data Sample Points

12.2.4.2 9-bit Data Transfer

The UART can be configured to have 9-bit data transfer in both transmit and receive mode. The 9th bit in the character appears after the 8th bit and before the parity bit in the character. Figure 12-2-4 shows the serial transmission for a character in which D8 represents the 9th bit and also shows general serial transmission in the 9-bit mode.

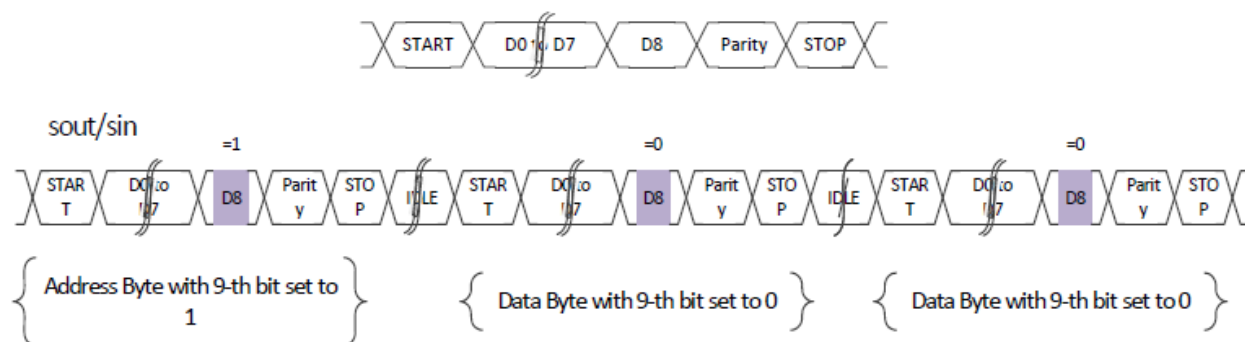


Figure 12-2-4 9-Bit Character

By enabling 9-bit data transfer mode, UART can be used in multi-drop systems where one master is connected to multiple slaves in a system. The master communicates with one of the slaves. When the master wants to transfer a block of data to a slave, it first sends an address byte to identify the target slave.

The differentiation between the address/data byte is done based on the 9th bit in the incoming character. If the 9th bit is set to 0, then the character represents a data byte. If the 9th bit is set to 1, then the character represents address byte. All the slave systems compare the address byte with their own address and only the target slave (in which the address has matched) is enabled to receive data from the master. The master then starts transmitting data bytes to the target slave. The non-addressed slave systems ignore the incoming data until a new address byte is received.

In Figure 12-2-4, note that one address is followed by 2 data bytes. The address byte goes out with the 9th bit (D8) set to 1 and the data bytes go out with 9th bit (D8) set to 0. The parity bit is an optional field.

Configuration of the UART for 9-bit data transfer does the following:

- LCR_EXT[0] bit is used to enable or disable the 9-bit data transfer.
- LCR_EXT[1] bit is used to choose between hardware and software based address match in the case of receive.
- LCR_EXT[2] bit is used to enable to send the address in the case of transmit.
- LCR_EXT[3] bit is used to choose between hardware and software based address transmission.
- TAR and RAR registers are used to transmit address and to match the received address, respectively.
- THR, RBR, STHR and SRBR registers are of 9-bit which is used to do the data transfers in 9-bit mode.
- LSR[8] bit is used to indicate the address received interrupt.

12.2.4.2.1 Transmit Mode

UART supports two types of transmit modes:

- Transmit Mode 0 (when (LCR_EXT[3]) is set to 0)

In transmit mode 0, the address is programmed in the Transmit Address Register (TAR) register and data is written into the Transmit Holding Register (THR) or the Shadow Transmit Holding Register (STHR). The 9th bit of the THR and STHR register is not applicable in this mode.

Figure 12-2-5 illustrates the transmission of address and data based on SEND_ADDR (LCR_EXT[2]), Halt Tx, and TxFIFO/THR empty conditions.

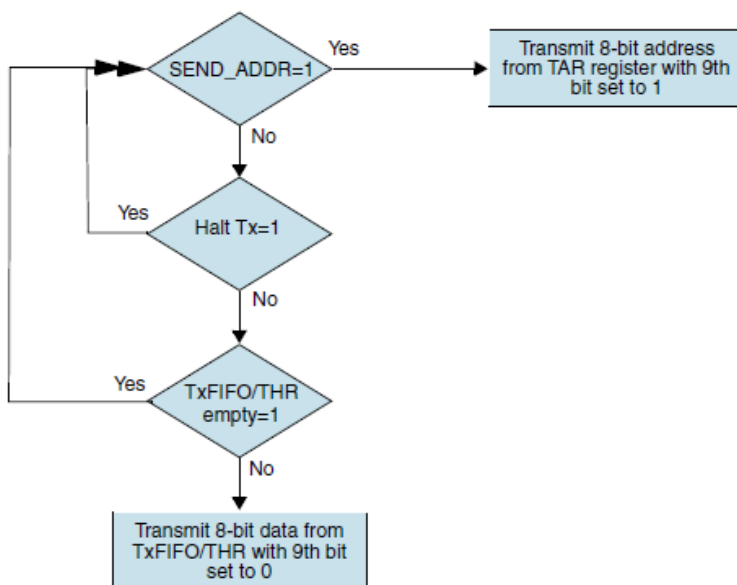


Figure 12-2-5 Auto Address Transmit Flow Chart

The address of the target slave to which the data is to be transmitted is programmed in the TAR register. You must enable the SEND_ADDR (LCR_EXT[2]) bit to transmit the target slave address present in the TAR register on the serial UART line with 9th data bit set to 1 to indicate that the address is being sent to the slave. The UART clears the SEND_ADDR bit after the address character starts transmitting on the UART line.

The data required to transmit to the target slave is programmed through Transmit Holding Register (THR). The data is transmitted on the UART line with 9th data bit set to 0 to indicate data is being sent to the slave. If the application is required to fill the data bytes in the TxTIFO before sending the address on the UART line (before setting LCR_EXT[2]=1), then it is recommended to set the "Halt Tx" to 1 such that UART does not start sending out the data in the TxTIFO as data byte. Once the TxTIFO is filled, then program SEND_ADDR (LCR_EXT[2]) to 1 and then set "Halt Tx" to 0.

- Transmit Mode 1 (when (LCR_EXT[3]) is set to 1)

In transmit mode 1, THR and STHR registers are of 9-bit wide and both address and data are programmed through the THR and STHR registers. The UART does not differentiate between address and data, and both are taken from the TxTIFO. The SEND_ADDR (LCR_EXT[2]) bit and Transmit address register (TAR) are not applicable in this mode. The software must pack the 9th bit with 1/0 depending on whether address/data has to be sent.

12.2.4.2.2 Receive Mode

The UART supports two receive modes:

- Hardware Address Match Receive Mode (when ADDR_MATCH (LCR_EXT[1]) is set to 1)

In the hardware address match receive mode, the UART matches the received character with the address programmed in the Receive Address register (RAR), if the 9th bit of the received character is set to 1. If the received address is matched with the programmed address in RAR register, then subsequent data bytes (with 9th bit set to 0) are pushed into the RxTIFO. If the address matching fails, then UART discards further data characters until a matching address is

received.

Figure 12-2-6 illustrates the flow chart for the reception of data bytes based on the address matching feature.

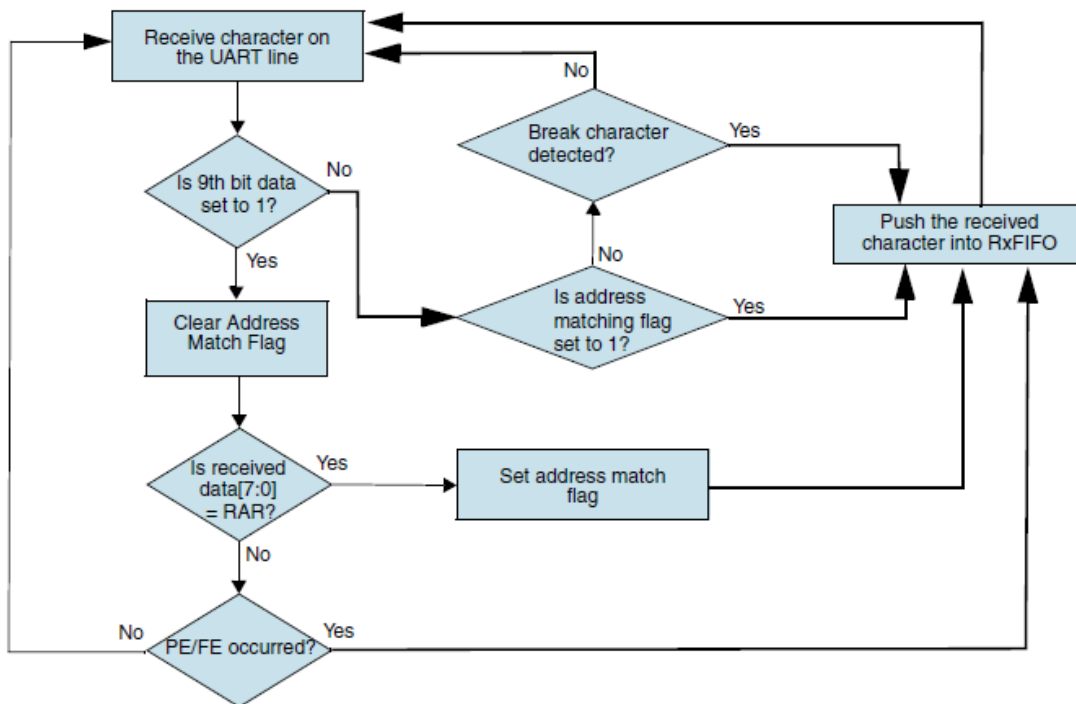


Figure 12-2-6 Hardware Address Match Receive Mode

UART receives the character irrespective of whether the 9th bit data is set to 1. If 9th bit of the received character is set to 1, then it clears internal address match flag and then compares the received 8-bit character information with the address programmed in the RAR register.

If the received address character matches with the address programmed in the RAR register, then the address match flag is set to 1 and the received character is pushed to the Rx FIFO in FIFO-mode or to RBR register in non-FIFO mode and the ADDR_RCVD bit in LSR register is set to indicate that the address has been received.

In case of parity or if a framing error is found in the received address character and if the address is not matched with the RAR register, then the received address character is still pushed to Rx FIFO or RBR register with ADDR_RCVD and PE/FE error bit set to 1.

The subsequent data bytes (9th bit of received character is set to 0) are pushed to the Rx_FIFO in FIFO mode or to the RBR register in non-FIFO mode until the new address character is received.

If any break character is received, UART treats it as a special character and pushes to the Rx FIFO or RBR register based on the FIFO_MODE irrespective of address match flag.

- Software Address Match Receive Mode (when ADDR_MATCH (LCR_EXT[1]) is set to 0)

In this mode of operation, the UART does not perform the address matching for the received address character (9th bit data set to 1) with the RAR register. The UART always receives the 9-bit data and pushes in to Rx FIFO in FIFO mode or to the RBR register in non-FIFO mode. The user must compare the address whenever address byte is received and indicated through ADDR_RCVD bit in the Line Status register. The user can flush/reset the Rx FIFO in case of address not matched through 'RCVR FIFO Reset' bit in FIFO control register (FCR).

12.2.4.3 RS485 Serial Protocol

The RS485 standard supports serial communication over a twisted pair configuration, such as RS232. The difference between the RS232 and RS485 standards is its use of a balanced line for transmission. This usage is also known as the differential format that sends the same signal on two separate lines with phase delay and then compares the signals at the end, subtracts any noise, and adds them to regain signal strength. This process allows the RS485

standard to be viable over significantly longer distances than its short range RS232 counterpart.

UART supports the RS485 serial protocol that enables transfer of serial data using the RS485 interface. The driver enable (DE) and receiver enable (RE) signals are generated for enabling the RS485 interface support. The de and re signals are hardware generated and the assertion/de-assertion times for these signals are programmable. The active level of these signals are configurable.

Configuration of the UART for RS485 interface does the following:

1. Bit 0 of the Transceiver Control Register (TCR) enables or disables the RS485 mode.
2. Bit 1 and bit 2 of TCR are used to select the polarity of RE and DE signals.
3. Bit [4:3] of the TCR selects the type of transfer in RS485 mode.
4. Driver output enable (DE_EN) and Receiver output enable (RE_EN) registers are used for software control of DE and RE signals.
5. Driver output Enable Timing (DET) register is used to program the assertion and deassertion timings of DE signal.
6. TurnAround Timing (TAT) register is used to program the turnaround time from DE to RE and RE to DE.

12.2.4.3.1 DE Assertion and De-assertion Timing

The assertion and deassertion timings of the DE signal are controlled through the DET register:

- DE assertion time (DET[7:0]): The assertion time is the time between the activation of the DE signal and the beginning of the START bit. The value represented is in terms of serial clock cycles.
- DE de-assertion time (DET[15:8]): The de-assertion time is the time between the end of the last stop bit, in a transmitted character, and the de-activation of the DE signal. The value represented is in terms of serial clock cycles.

Hardware ensures that these values are met for DE assertion and DE deassertion before/after active data transmission.

In Figure 12-2-7, t1 represents DE assertion time and t2 represents DE de-assertion time. Note that for simplicity only one data is illustrated in Figure 12-2-7; however, DE will not get de-asserted if there are more data characters in transmit FIFO. DE gets de-asserted only after all the data characters are transmitted.

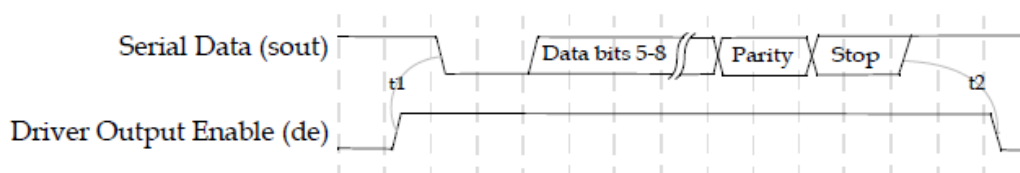


Figure 12-2-7 DE Assertion and De-Assertion

12.2.4.3.2 RS485 Modes

UART consists of the following RS485 modes based on the XFER_MODE field in the Transceiver Control Register (TCR) register:

1. Full Duplex Mode – In this mode, XFER_MODE of TCR is set to 0.

The full duplex mode supports both transmit and receive transfers simultaneously.

In Full Duplex mode, the de signal:

- Goes active if both these conditions are satisfied:
 - When the DE Enable (DE_EN[0]) field of Driver Output Enable Register is set to 1.
 - Transmitter Holding Register is not empty in non-FIFO mode or transmitter FIFO is not empty in FIFO mode.
- Goes inactive if both these conditions are satisfied
 - When the current ongoing transmitting serial transfer is completed.

- Either DE Enable (DE_EN[0]) of Driver Output Enable Register is set to 0, transmitter FIFO is empty in FIFO mode or Transmitter Holding Register is empty in non-FIFO mode.

In Full Duplex mode, the re signal:

- Goes active when RE Enable (RE_EN[0]) of Receiver Output Enable Register is set to 1.
- Goes inactive when RE Enable (RE_EN[0]) of Receiver Output Enable Register is set to 0.

The user can choose when to transmit or when to receive. Both 're' and 'de' can be simultaneously asserted or de-asserted at any time. UART does not impose any turnaround time between transmit and receive ('de to re') or receive to transmit ('re to de') in this mode. This mode can directly be used in full duplex operation where separate differential pair of wires is present for transmit and receive.

2. Software-Controlled Half Duplex Mode – In this mode, XFER_MODE of TCR is set to 1.

The software-controlled half duplex mode supports either transmit or receive transfers at a time but not both simultaneously. The switching between transmit to receive or receive to transmit is through programming the Driver output enable (DE_EN) and Receiver output enable (RE_EN) registers.

In software-controlled Half Duplex mode, the de signal:

- Goes active if the following conditions are satisfied:
 - The DE Enable (DE_EN[0]) field of the Driver Output Enable Register is set to 1.
 - Transmitter Holding Register is not empty in non-FIFO mode or transmitter FIFO is not empty in FIFO mode.
 - If any receive transfer is ongoing, then the signal waits until receive has finished, and after the turnaround time counter ('re to de') has elapsed.
- Goes inactive if the following conditions are satisfied:
 - The current ongoing transmitting serial transfer is completed.
 - The DE Enable (DE_EN[0]) field of Driver Output Enable Register is set to 0.
 - Either transmitter FIFO is empty in FIFO mode or Transmitter Holding Register is empty in non- FIFO mode.

In software-controlled half duplex mode, the re signal:

- Goes active if the following conditions are satisfied:
 - When RE Enable (RE_EN[0]) field of Receiver Output Enable Register is set to 1.
 - If any transmit transfer is ongoing, then the signal waits until transmit has finished and after the turnaround time counter ('de to re') has elapsed.
- Goes in-active under the following conditions:
 - The current ongoing receive serial transfer is completed.
 - When RE Enable (RE_EN[0]) of Receiver Output Enable Register is set to 0.

The user must enable either DE or RE but not both at any point of time. As 're' and 'de' signals are mutually exclusive, the user must ensure that both of them are not programmed to be active at any point of time.

In this mode, the hardware ensures that a proper turnaround time is maintained while switching from 're' to 'de' or from 'de' to 're' (value of turnaround is obtained from the TAT register, in terms of serial clock cycles) as shown in Figure 12-2-8 and Figure 12-2-9.

UART inserts the wait state (as programmed in TAT[31:16] times serial clock) before switching to transmit mode from receive mode as shown in the Figure 12-2-8 (applicable only when TCR[4:3] =1 or 2 (XFER_MODE).)

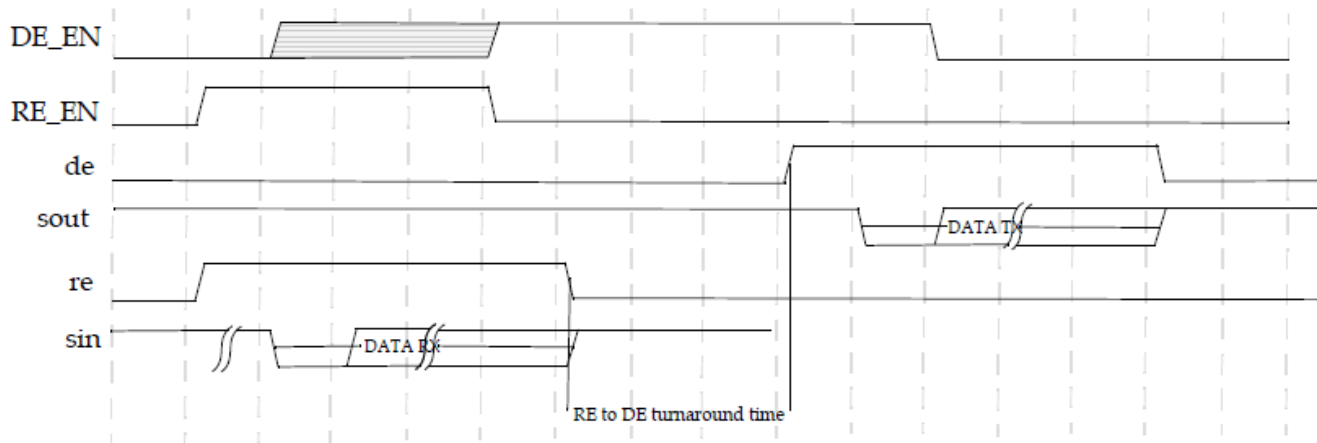


Figure 12-2-8 DE RE to DE Turnaround Time

UART inserts the wait state (as programmed in TAT[15:0] times serial clock) before switching to receive mode from transmit mode as shown in the Figure 12-2-9 (applicable only when TCR[4:3] =1 or 2 (XFER_MODE).)

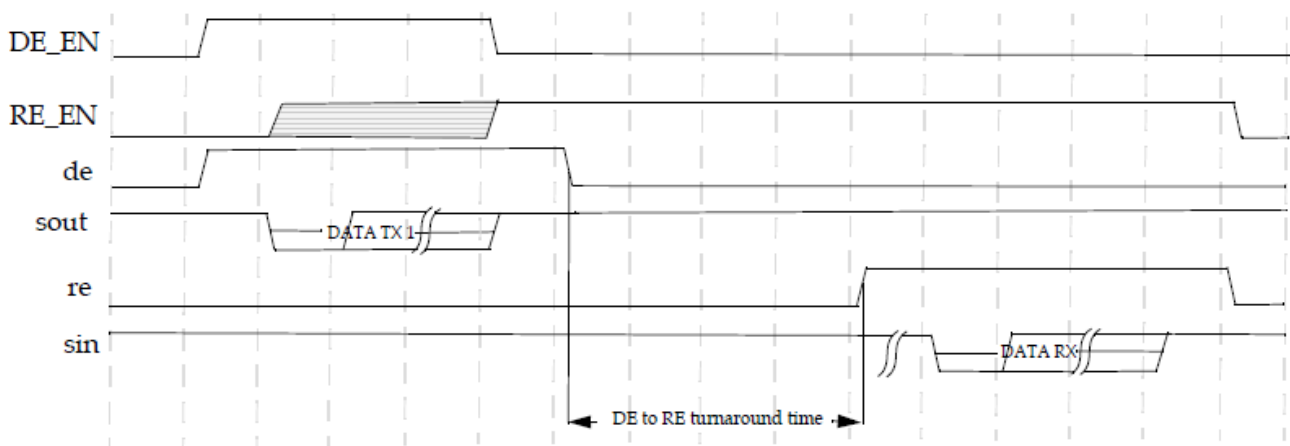


Figure 12-2-9 DE to RE Turnaround Time

3. Hardware-Controlled Half Duplex Mode – In this mode, XFER_MODE of TCR is set to 2

The hardware-controlled half duplex mode supports either transmit or receive transfers at a time but not both simultaneously. If both 'DE Enable' and 'RE Enable' bits of Driver output enable (DE_EN) and Receiver output enable (RE_EN) registers are enabled, the switching between transmit to receive or receive to transmit is automatically done by the hardware based on the empty condition of Tx-FIFO.

In hardware-controlled half duplex mode, the de signal:

- Goes active if the following conditions are satisfied:
 - The DE Enable (DE_EN[0]) field of Driver Output Enable Register is set to 1.
 - Transmitter Holding Register is not empty in non-FIFO mode or transmitter FIFO is not empty in FIFO mode.
 - If any receive transfer is ongoing, then the signal waits until receive is finished and after the turnaround time counter ('re to de') has elapsed.
- Goes inactive if the following conditions are satisfied
 - The current ongoing transmitting serial transfer is completed.
 - Either transmitter FIFO is empty in FIFO mode or Transmitter Holding Register is empty in non- FIFO mode or the DE Enable (DE_EN[0]) of Driver output Enable Register is set to 0.

In hardware-controlled half duplex mode, the re signal:

- Goes active if the following conditions are satisfied:
 - When RE Enable (RE_EN[0]) field of Receiver Output Enable Register is set to 1.
 - Either transmitter FIFO is empty in FIFO mode or Transmitter Holding Register is empty in non- FIFO mode.
 - If any transmit transfer is ongoing, then the signal waits until transmit is finished and after the turnaround time counter ('de to re') has elapsed.
- Goes inactive under the following conditions:
 - The current ongoing receive serial transfer has completed.
 - Either transmitter FIFO is non-empty in FIFO mode or Transmitter Holding Register is non empty in non- FIFO mode or the RE Enable (RE_EN[0]) of Receiver output Enable Register is set to 0.

In this mode, the hardware ensures that a proper turnaround time is maintained while switching from 're' to 'de' or from 'de' to 're' (value of turnaround is obtained from the TAT register, in terms of serial clock cycles) as shown in Figure 12-2-8 and Figure 12-2-9.

12.2.4.4 Fractional Baud Rate Support

UART supports fractional baud rate that enables a user to program the fractional part of the divisor value to generate fractional baud rate that results in reduced frequency error. The UART interface usage has been evolving to include ever increasing baud rate speeds. The UART needs to be software configurable to handle the baud rates within 2% frequency error.

The Baud rate of UART is controlled by sclk in asynchronous serial clock (CLOCK_MODE=2) implementation or pclk in single clock implementation (CLOCK_MODE=1) and the Divisor Latch Register (DLH and DLL).

The baud rate is determined by the following factors:

- Serial clock operating frequency (sclk in Asynchronous serial clock implementation or pclk in single clock implementation)
- The desired baud rate.
- The baud rate generator divisor value, DIVISOR (composed of DLH & DLL registers).
- The acceptable Baud-rate error, %ERROR

The equation to calculate the baud rate is as follows:

$$\text{Baud Rate} = \text{Serial Clock Operating Frequency} / (16 \times \text{DIVISOR}) \text{ ---(1)}$$

Where,

DIVISOR – Number (in hexadecimal) to program the DLL and DLH.

Serial clock frequency – Frequency at sclk or pclk pin of UART.

From Equation (1), DIVISOR can be calculated as:

$$\text{DIVISOR} = \text{Serial Clock Operating Frequency} / (16 \times \text{Baud Rate}) \text{ ---(2)}$$

Also from Equation (1), it can also be shown that:

$$\text{Serial clock frequency} = \text{Baud Rate} \times 16 \times \text{DIVISOR} \text{ ---(3)}$$

The Error between the Baud rate and Baud rate (selected) is given as:

$$\text{Percentage ERROR} = |\text{Baud Rate} - \text{Baud Rate (selected)}| \times 100 / \text{Baud Rate} \text{ ---(4)}$$

Configuration of the UART for Fractional Baud Rate does the following:

- The configurable parameter DLF_SIZE is used to choose the width of the register that stores fractional part of the divisor.
- The fractional value of the divisor is programmed in the Divisor Latch Fraction Register (DLF) register. The fractional value is computed by using the (Divisor Fraction value)/(2^{DLF_SIZE}) formula. Table 12-2-1 shows

fractional values when the DLF_SIZE=4.

Table 12-2-1 Divisor Latch Fractional Values

DLF Value	Fraction	Fractional Value
0000	0/16	0.0000
0001	1/16	0.0625
0010	2/16	0.125
0011	3/16	0.1875
0100	4/16	0.25
0101	5/16	0.3125
0110	6/16	0.375
0111	7/16	0.4375
1000	8/16	0.5
1001	9/16	0.5625
1010	10/16	0.625
1011	11/16	0.6875
1100	12/16	0.75
1101	13/16	0.8125
1110	14/16	0.875
1111	15/16	0.9375

The programmable fractional baud rate divisor enables a finer resolution of baud clock than the conventional integer divider. The programmable fractional baud clock divider allows for the programmability of both an integer divisor as well as fractional component. The average frequency of the baud clock from the fractional baud rate divisor is dependent upon both the integer divisor and the fractional component, thereby providing a finer resolution to the average frequency of the baud clock.

$$\text{Baud Rate Divisor} = \text{Serial Clock Frequency} / (16 \times \text{Required Baud Rate}) = \text{BRDI} + \text{BRDF} \text{ ---(5)}$$

Where,

BRDI - Integer part of the divisor.

BRDF - Fractional part of the divisor.

1. Fractional Division Used to Generate Baud Clock

Fractional division of clock is used by the $N/N+1$ divider, where N is the integer part of the divisor. $N/N+1$ division works on the basis of achieving the required average timing over a long period by alternating the division between two numbers. If $N=1$ and ratio of $N/N+1$ is same, which means equal number of divide by 1 and divide by 2 over a period of time, average time period would come out to be divided by 1.5. Varying the ratio of $N/N+1$ any value can be achieved above 1 and below 2.

2. Calculating the Fractional Value Error

Following is a sample for calculating the fractional value error.

Consider the following values:

- Required Baud Rate (RBR) = 4454400
- Serial Clock (SCLK) = 133MHz
- DLF_SIZE = 4

Then, as per equation (5), Baud Rate Divisor (BRD) is as follows:

$$BRD = 133 / (16 \times 4454400) = 1.866132364 \text{ ---(6)}$$

In (6), the integer and fractional parts are as follows:

- Integer part (BRDI) = 1
- Fractional part (BRDF) = 0.866132364

Therefore, Baud Rate Divisor Latch Fractional Value (DLF) is as follows:

$$DLF = BRDF \times 2DLF\_SIZE = 0.866132364 \times 16 = 13.858117824 = 14 \text{ (roundoff value) ---(7)}$$

The Generated Baud Rate Divider (GD) is as follows:

$$GD = BRDI + DLF / 2DLF\_SIZE = 1 + 14/16 = 1.875 \text{ ---(8)}$$

Therefore, the Generated Baud Rate (GBR) is as follows:

$$GBR = \text{Serial Clock} / (16 \times GD) = 133 / (16 \times 1.875) = 4433333.333 \text{ ---(9)}$$

Now the error is calculated as follows:

$$\text{Error} = (GBR - RBR) / RBR = 0.004729 \text{ ---(11)}$$

The error percentage is as follows:

$$\text{Error \%} = 0.004729 \times 100 = 0.473 \text{ ---(12)}$$

If serial clock is 25 MHz and Baud rate required = 892857, divisor comes out to be 1.75. Without a fractional division a value of 1 or 2 will result in baud rate of 1562500 or 781250 which is more than 2% frequency error. However, if we divide 12 clocks by 2 and then 4 clocks by 1, over an average period of 16 clocks we'll achieve division by 1.75. Using this divisor baud rate of 892857 can be achieved.

As shown in the Figure 12-2-10, the fractional baud clock is generated between N(1) and N+1(2) values to generate the fractional baud rate of 1.75 to achieve the divisor baud rate of 892857 with 0% frequency error compared to 12.49% frequency error in integer baud clock generator.

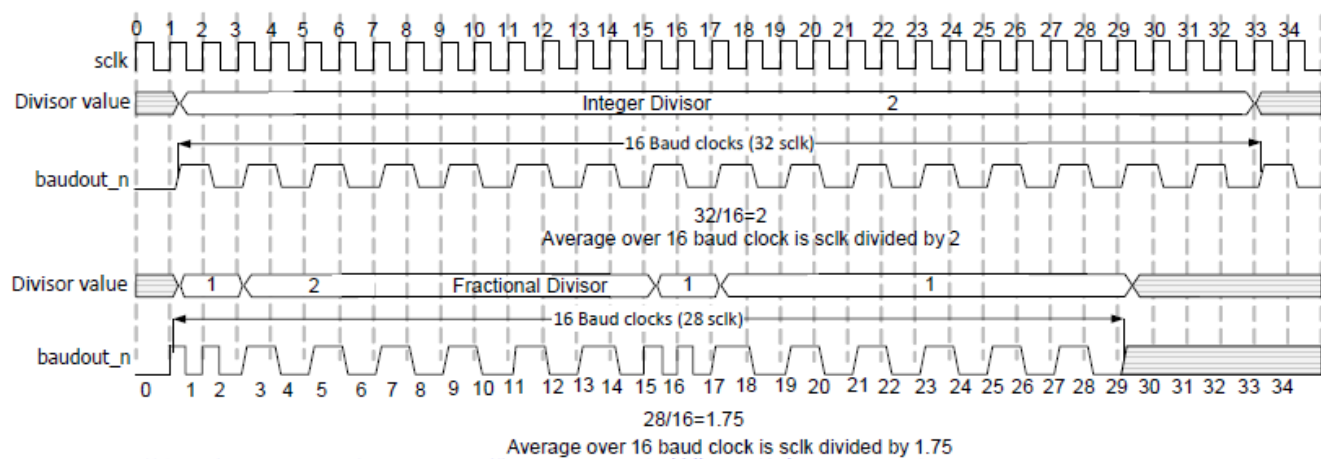


Figure 12-2-10 Example of Integer and Fractional Division Over 16 Clock Periods

12.2.4.5 IrDA 1.0 SIR Protocol

The Infrared Data Association (IrDA) 1.0 Serial Infrared (SIR) mode supports bi-directional data communications with remote devices using infrared radiation as the transmission medium. IrDA 1.0 SIR mode specifies a maximum baud rate of 115.2 Kbaud.

IrDA specification can be obtained from the following website:

<<http://www.irda.org>

> The data format is similar to the standard serial—sout and sin—data format. Each data character is sent serially in this order:

1. Begins with a start bit
2. Followed by 8 data bits
3. Ends with at least one stop bit

Thus, the number of data bits that can be sent is fixed. No parity information can be supplied, and only one stop bit is used in this mode. Trying to adjust the number of data bits sent or enable parity with the Line Control Register (LCR) has no effect.

Configuration of the UART for IrDA 1.0 SIR does the following:

- Bit 6 of the Mode Control Register (MCR) enables or disables the IrDA 1.0 SIR mode.
- Disabling IrDA SIR mode causes the logic to not be implemented; the mode cannot be activated, which reduces total gate counts.
- When IrDA SIR mode is enabled and active, serial data is transmitted and received on the sir_out_n and sir_in ports, respectively.

Transmission or non-transmission of a single infrared pulse indicates the following:

- Transmitting a single infrared pulse indicates logic 0
- Non-transmission of a pulse indicates logic 1

The width of each pulse is 3/16ths of a normal serial bit time. Thus, each new character begins with an infrared pulse for the start bit. However, received data is inverted from transmitted data due to infrared pulses energizing the photo transistor base of the IrDA receiver, which pulls its output low. This inverted transistor output is then fed to the UART sir_in port, which gives it the correct UART polarity.

Figure 12-2-11 shows the timing diagram for the IrDA SIR data format in comparison to standard serial format.

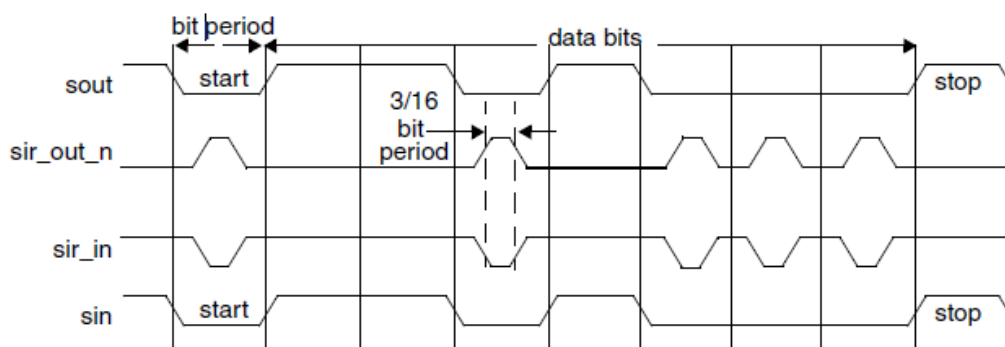


Figure 12-2-11 IrDA SIR Data Format

As previously mentioned, the UART can be configured to support a low-power reception mode. When the UART is configured in this mode, it is possible to receive SIR pulses of 1.41 microseconds (minimum pulse duration), as well as nominal 3/16 of a normal serial bit time. In order to use this lowpower reception mode, you must program the Low Power Divisor Latch (LPDLL/LPDLH) registers.

For all sclk frequencies greater than or equal to 7.37MHz, pulses of 1.41uS are detectable; these pulses comply with the requirements of the Low Power Divisor Latch registers. However, there are several values of sclk that do not allow detection of such a narrow pulse, as indicated in Table 12-2-2.

Table 12-2-2 Narrow Pulse Exceptions

SCLK	Low Power Divisor Latch Register Value	Min Pulse Width for Detection
1.84MHz	1	3.77uS
3.69MHz	2	2.086uS
5.53MHz	3	1.584uS

When IrDA SIR mode is enabled, the UART operates in a manner similar to when the mode is disabled, with one exception: data transfers can only occur in half-duplex fashion when IrDA SIR mode is enabled. This is because the IrDA SIR physical layer specifies a minimum of 10ms delay between transmission and reception; this 10ms delay must be generated by software.

12.2.4.6 FIFO Support

You can configure the UART to implement FIFOs that buffer transmit and receive data. Figure 12-2-12 shows the timing diagram for both asynchronous and synchronous RAMs.

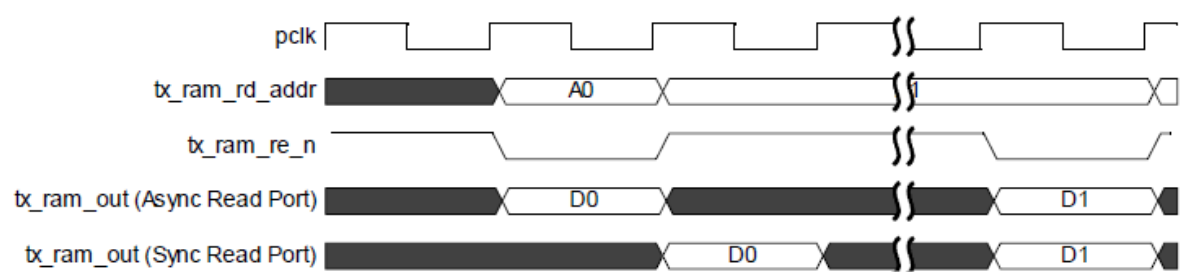


Figure 12-2-12 Timing for RAM Reads

Similarly, you can use synchronous RAM for writes, which registers the data at the current address out. Figure 12-2-13 shows the timing diagram for RAM writes.

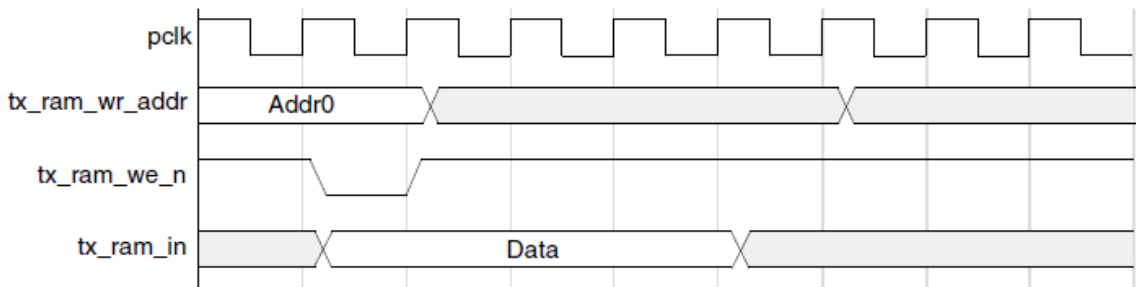


Figure 12-2-13 Timing for RAM Writes

When FIFO support is selected, an optional programmable FIFO Access mode is available for test purposes, which allows:

- Receive FIFO to be written by master
- Transmit FIFO to be read by master

When FIFO Access mode is not selected, none of the corresponding logic is implemented and the mode cannot be enabled, reducing overall gate counts. When FIFO Access mode has been selected it can be enabled with the FIFO Access Register (FAR[0]). Once enabled, the control portions of the transmit and receive FIFOs are reset and the FIFOs are treated as empty. Data can be written to the transmit FIFO as normal; however no serial transmission occurs in this mode— normal

operation halted—and thus no data leave the FIFO. The data that has been written to the transmit FIFO can be read back with the Transmit FIFO Read (TFR) register, which when read gives the current data at the top of the transmit FIFO.

Similarly, data can be read from the receive FIFO as normal. Since the normal operation of the UART is halted in this mode, data must be written to the receive FIFO so the data can be read back.

Data is written to the receive FIFO using the Receive FIFO Write (RFW) register. The upper two bits of the 10-bit register are used to write framing error and parity error detection information to the receive FIFO, as follows:

- RFW[9] indicates framing error
- RFW[8] indicates parity error

Although these bits cannot be read back through the Receive Buffer Register, they can be checked by reading the Line Status Register and checking the corresponding bits when the data in question is at the top of the receive FIFO.

12.2.4.7 Clock Support

The UART have two system clocks (pclk and sclk). The serial clock (sclk) accommodates accurate serial baud rate settings, as well as APB bus interface requirements. A synchronization module is implemented for synchronization of all control and data across the two-system clock boundaries.

The arrival of new source domain data is indicated by the assertion of start. Since data is now available for synchronization, the process is started and busy status is set. If start is asserted while busy and pending data capability has been selected, the new data is stored.

When no longer busy, the synchronization process starts on the stored pending data. Otherwise the busy status is removed when the current data has been synchronized to the destination domain and the process continues. If only one clock is implemented, all synchronization logic is absent and signals are simply passed through this module.

There are two types of signal synchronization:

- Data-synchronized signals – full synchronization handshake takes place on signals
- Level-synchronized signals – signals are passed through two destination clock registers

Both synchronization types incur additional data path latencies. However, this additional latency has no negative affect on received or transmitted data, other than to limit how much faster sclk can be in relation to pclk for back-to-back serial communications with no idle assertion.

A serial clock that exceeds this limit does not leave enough time for a complete incoming character to be received and pushed into the receiver FIFO. To ensure that you do not exceed the limit, the following equation must hold true:

$((2 \text{ pclk\_cycles}) + 4) < (39 \text{ (Baud Divisor)})$ Where:

pclk_cycles is expressed in sclk cycles

For example, if the Baud Divisor is programmed to 1 and a serial clock is 18 times faster than the pclk signal, the equation becomes:

$$((2 \cdot 18) + 4) < (39 \cdot 1) \geq 40 < 39$$

Thus the equation does not hold true, and the ratio 18:1 (sclk:pclk) exceeds the limit at this Baud rate. Here are a few things to keep in mind:

- A divisor greater than 1 at a clock ratio of 18:1 (sclk:pclk) does not cause data corruption issues due to synchronization, as the synchronization process has more time to transfer the received data to the peripheral clock domain before the next character bit is received.
- In most cases, however, the pclk signal is faster than sclk, so this should never be an issue.
- There is slightly more time required after initial serial control register programming before serial data can be transmitted or received.

- The serial clock modules must have time to see new register values and reset their respective state machines. This total time is guaranteed to be no more than eight clock cycles of the slower of the two system clocks. Therefore, no data should be transmitted or received before this maximum time expires, after initial configuration.

Each NOP usually takes one bus cycle to retire. However, the actual number of NOPs that need to be inserted in the assembly code is dependent on the maximum number of instructions that can be retired in a single cycle. So for example, if the processor uses a 4-dispatch pipe, then four NOPs could potentially retire in one bus cycle. Assuming that the next opcode (NOP) is fetched as per the slower clock—with eight clock cycles of the slower clock as the reference—a minimum of thirty-two NOPs need to be included in the assembly code after a software reset. In systems where only one clock is implemented, there are no additional latencies.

12.2.4.8 Back-to-Back Character Stream Transmission

When the Transmit FIFO contains multiple data entries, the UART transmits the characters in the FIFO back-to-back on the serial bus. Synchronization delays in the UART can cause an IDLE period between the end of the current STOP bit and the beginning of the next START bit; this appears as an extended STOP bit duration on the serial bus. the UART has a synchronization delay between the transmitter in the sclk domain and the TX FIFO in the pclk domain when querying if another character is ready for transmission. The transmitter begins the handshake one baud clock cycle before the end of the current STOP bit. The duration of the synchronization delay is given by the following equations:

$$\text{sync\_delay} = (1\text{sclk} + 3\text{pclk}) + 1\text{pclk} + (1\text{pclk} + 3\text{sclk})$$

$$\text{sync\_delay} = 4\text{sclk} + 5\text{pclk}$$

If the sync_delay duration is longer than one baud clock period, an IDLE period will be inserted between the end of a STOP bit and the beginning of the next START bit.

To prevent insertion of the IDLE period, the following condition must be true:

$$\text{sync\_delay} \leq \text{bclk\_period}$$

The baud clock period is given by the following equation:

$$\text{bclk\_period} = \{\text{DLH}, \text{DLL}\} * \text{sclk}$$

The worst case timing of the inserted IDLE period is given by:

$$\text{worst\_case\_idle\_duration} = \text{sync\_delay} + (15 * \text{bclk\_period})$$

The worst_case_idle_duration can be added to the programmed STOP bit duration to give the overall STOP bit period. Figure 12-2-14 illustrates an example of character finish to character start delay.

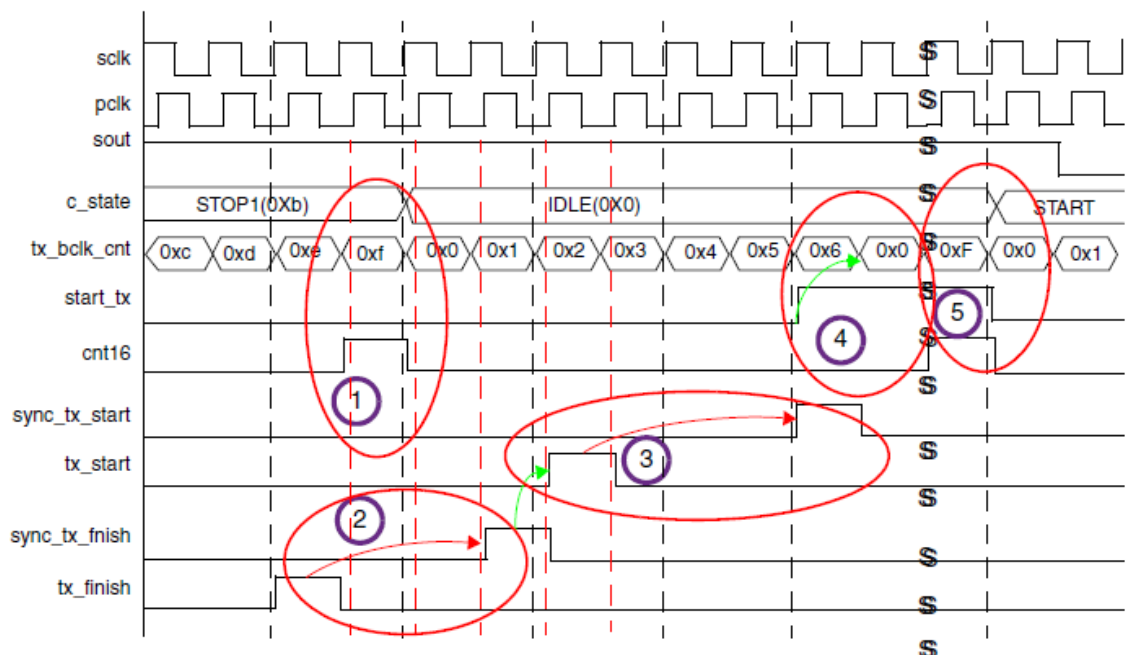


Figure 12-2-14 Character Finish to Character Start Delay

1. The baud divisor is set to 1 ($\{DLH, DLL\} = 1$), so every sclk is a baud clock cycle. The transmit state machine changes state every sixteen baud clocks—eight in the case of a half STOP bit. At this point in Figure 12-2-14, after 16 baud clock cycles of the STOP1 state, the state machine enters the IDLE state on the next cycle because start_tx is not yet asserted.
2. One baud clock before the end of the STOP state, the transmit state machine decodes that the current character is complete and asserts tx_finish, which is synchronized to the pclk domain to become sync_tx_finish; this synchronization accounts for the “1sclk + 3pclk” term in sync_delay.
3. In the pclk domain, there is a one-pclk cycle delay—“1pclk” term in sync_delay—before the signal tx_start is asserted from the assertion of sync_tx_finish. Tx_start must then be synchronized to the sclk domain—“1pclk + 3sclk” term in sync_delay—to instruct the state machine to commence the START bit of the next character.
4. Start_tx asserts in the sclk domain, and causes the baud clock counter (tx_bclk_cnt) to go to 0.
5. Once sixteen baud clocks have been counted, the state machine can transition into the START state, and one cycle later sout is de-asserted.

12.2.4.9 Interrupts

Assertion of the UART interrupt output signal (intr)—a positive-level interrupt—occurs whenever one of the several prioritized interrupt types are enabled and active.

When an interrupt occurs, the master accesses the IIR register.

The following interrupt types can be enabled with the IER register:

- Receiver Error
- Receiver Data Available
- Character Timeout (in FIFO mode only)
- Transmitter Holding Register Empty at/below threshold (in Programmable THRE interrupt mode)
- Modem Status
- Busy Detect Indication

These interrupt types are explained in detail in Table 12-2-3.

Table 12-2-3 Interrupt Control Functions (Continued)

Interrupt ID	Interrupt Set and Reset Functions

Bit 3	Bit 2	Bit 1	Bit 0	Priority Level	Interrupt Type	Interrupt Source	Interrupt Reset Control
0	0	0	1	–	None	None	–
0	1	1	0	Highest	Receiver line status	Overrun/parity/ framing errors, break interrupt, or address received interrupt	For Overrun/parity/framing/break interrupt reset control, the behavior is as follows: - If LSR_STATUS_CLEAR=0 (RBR Read or LSR Read), then the status is cleared on: -- Reading the line status register - Or -- In addition to an LSR read, the Receiver line status is also cleared when RX_FIFO is read. - If LSR_STATUS_CLEAR=1 (LSR Read), the status is cleared only on: -- Reading the line status register. - For address received interrupt, the status is cleared on: -- Reading the line status register
0	1	0	0	Second	Received data available	Receiver data available (non-FIFO mode or FIFOs disabled) or RCVR FIFO trigger level reached (FIFO mode and FIFOs enabled)	Reading the receiver buffer register (non-FIFO mode or FIFOs disabled) or the FIFO drops below the trigger level (FIFO mode and FIFOs enabled)
1	1	0	0	Second	Character timeout indication	No characters in or out of the RCVR FIFO during the last 4 character times and there is at least 1 character in it during this time	Reading the receiver buffer register
0	0	1	0	Third	Transmit holding register empty	Transmitter holding register empty (Prog. THRE Mode disabled) or XMIT FIFO at or below threshold (Prog. THRE Mode enabled)	Reading the IIR register (if source of interrupt); or, writing into THR (FIFOs or THRE Mode not selected or disabled) or XMIT FIFO above threshold (FIFOs and THRE Mode selected and enabled).
0	0	0	0	Fourth	Modem status	Clear to send or data set ready or ring indicator or data carrier detect. Note that if auto flow control mode is enabled, a change in CTS (that is, DCTS set) does not cause an interrupt.	Reading the Modem status register
0	1	1	1	Fifth	Busy detect	UART_16550_COMPATIBLE = NO and master has tried to	Reading the UART status register

					indication	write to the Line Control Register while the UART is busy (USR[0] is set to 1).

12.2.4.10 Interrupts

The UART has a 16750-compatible Auto RTS and Auto CTS serial data flow control mode. It can be enabled with the Modem Control Register (MCR[5]).

Figure 12-2-15 shows a block diagram of the Auto Flow Control functionality.

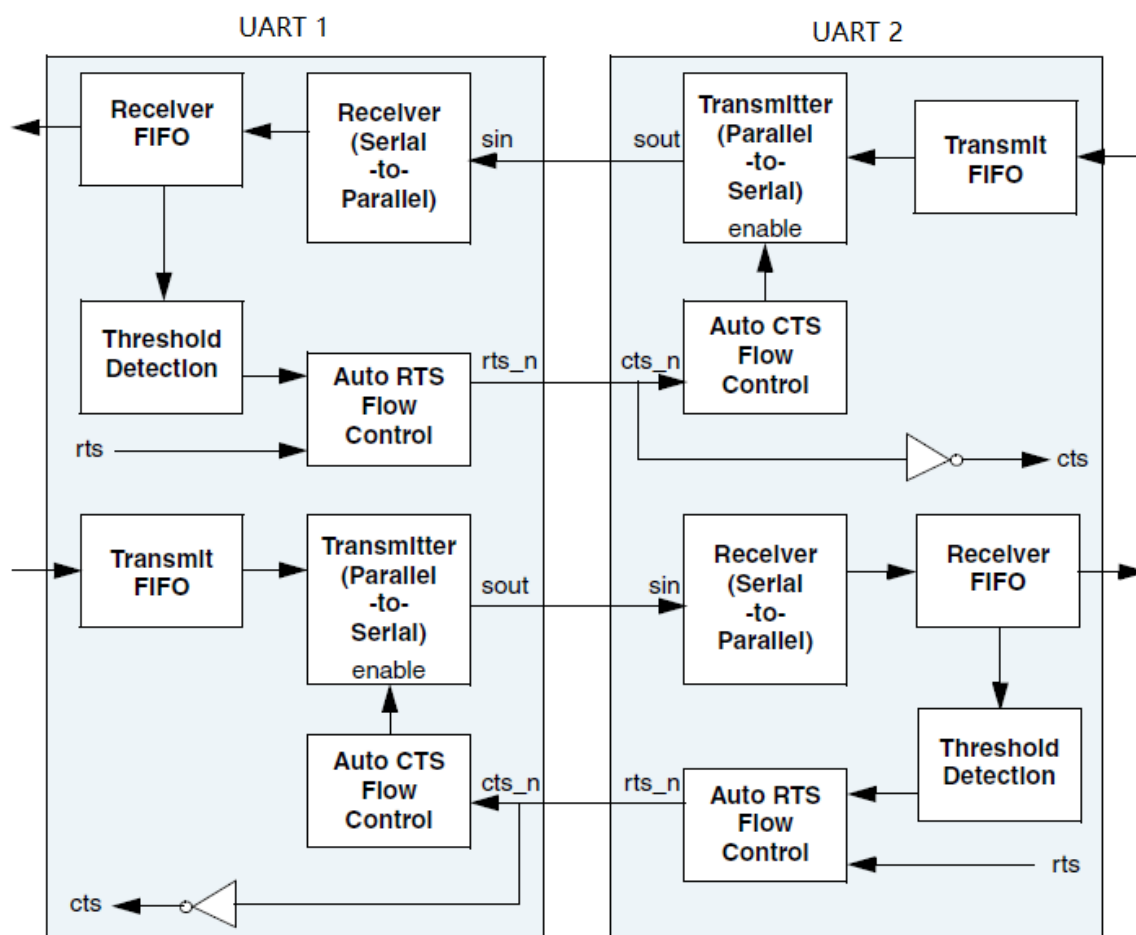


Figure 12-2-15 Auto Flow Control Block Diagram

Auto RTS and Auto CTS are described as follows:

1. Auto RTS – Becomes active when the following occurs:

- Auto Flow Control is selected during configuration
- FIFOs are implemented
- RTS (MCR[1] bit and MCR[5] bit are both set)
- FIFOs are enabled (FCR[0] bit is set)
- SIR mode is disabled (MCR[6] bit is not set)

When Auto RTS is enabled, the rts_n output is forced inactive (high) when the receiver FIFO level reaches the threshold set by FCR[7:6], but only if the RTC flow-control trigger is disabled. Otherwise, the rts_n output is forced inactive (high) when the FIFO is almost full, where “almost full” refers to two available slots in the FIFO. When rts_n is connected to the cts_n input of another UART device, the other UART stops sending serial data until the receiver FIFO has available space; that is, until it is completely empty.

The selectable receiver FIFO threshold values are:

- 1
- $\frac{1}{4}$
- $\frac{1}{2}$
- 2 less than full

Since one additional character can be transmitted to the UART after `rts_n` has become inactive—due to data already having entered the transmitter block in the other UART—setting the threshold to “2 less than full” allows maximum use of the FIFO with a safety zone of one character.

Once the receiver FIFO becomes completely empty by reading the Receiver Buffer Register (RBR), `rts_n` again becomes active (low), signalling the other UART to continue sending data.

Figure 12-2-16 shows a timing diagram of the Auto RTS operation.

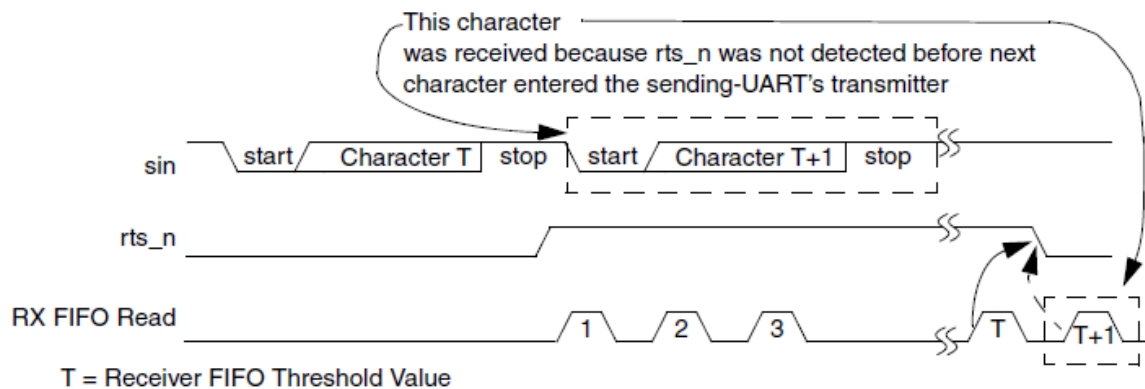


Figure 12-2-16 Auto RTS Timing

2. Auto CTS – becomes active when the following occurs:

- Auto Flow Control is selected during configuration
- FIFOs are implemented
- AFCE (MCR[5] bit = 1)
- FIFOs are enabled through FIFO Control Register FCR[0] bit
- SIR mode is disabled (MCR[6] bit = 0)

When Auto CTS is enabled (active), the UART transmitter is disabled whenever the `cts_n` input becomes inactive (high); this prevents overflowing the FIFO of the receiving UART.

If the `cts_n` input is not inactivated before the middle of the last stop bit, another character is transmitted before the transmitter is disabled. While the transmitter is disabled, the transmitter FIFO can still be written to, and even overflowed.

Therefore, when using this mode, the following happens:

- UART status register can be read to check if transmit FIFO is full (USR[1] set to 0)
- Current FIFO level can be read using TFL register
- Programmable THRE Interrupt mode must be enabled to access “FIFO full” status using Line Status Register (LSR)

When using the “FIFO full” status, software can poll this before each write to the Transmitter FIFO. When the `cts_n` input becomes active (low) again, transmission resumes.

When everything else is selected, if the FIFOs are disabled using FCR[0], Auto Flow Control is also disabled. When Auto CTS is not implemented or disabled, the transmitter is unaffected by `cts_n`.

Figure 12-2-17 illustrates a timing diagram that shows the Auto CTS operation.

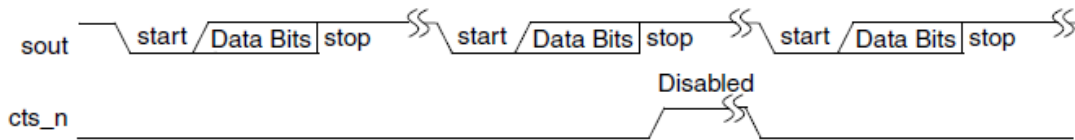


Figure 12-2-17 Auto CTS Timing

12.2.4.11 Programmable THRE Interrupt

The UART has a Programmable THRE Interrupt mode which increases system performance; if FIFOs are not implemented, then this mode cannot be selected.

- When Programmable THRE Interrupt mode is not selected, none of the logic is implemented and the mode cannot be enabled, reducing the overall gate counts.
- When Programmable THRE Interrupt mode is selected, it can be enabled using the Interrupt Enable Register (IER[7]).

When FIFOs and THRE mode are implemented and enabled, the THRE Interrupts and dma_tx_req_n are active at, and below, a programmed transmitter FIFO empty threshold level, as opposed to empty, as shown in the flowchart in the figure below.

For the THRE interrupt to be controlled as shown here, the following must be true:

- FIFO_MODE != NONE
- THRE_MODE = Enabled
- FIFOs enabled (FCR[0] = 1)
- THRE mode enabled (IER[7] = 1)

Under the condition that there are no other pending interrupts, the interrupt signal (intr) is asserted

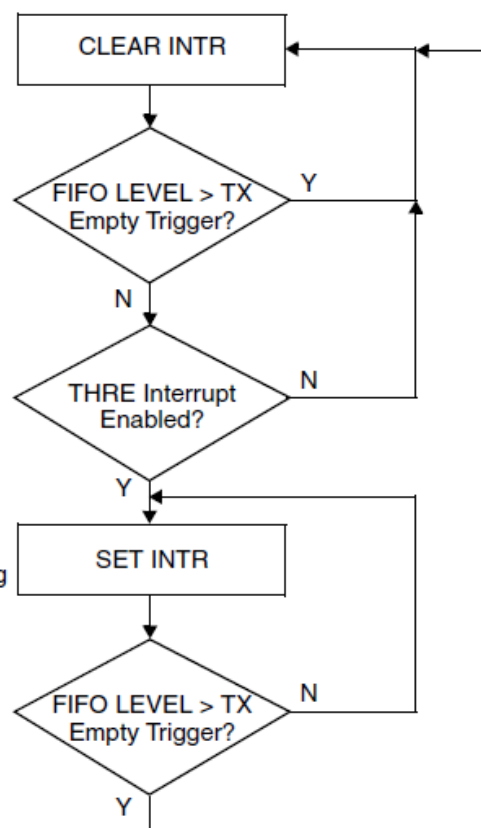


Figure 12-2-18 Flowchart of Interrupt Generation for Programmable THRE Interrupt Mode

The threshold level is programmed into FCR[5:4]. Available empty thresholds are:

- empty
- 2
- $\frac{1}{4}$
- $\frac{1}{2}$

Selection of the best threshold value depends on the system's ability to begin a new transmission sequence in a timely manner. However, one of these thresholds should be optimal for increasing system performance by preventing the transmitter FIFO from running empty.

In addition to the interrupt change, the Line Status Register (LSR[5]) also switches from indicating that the transmitter FIFO is empty to the FIFO being full. This allows software to fill the FIFO for each transmit sequence by polling LSR[5] before writing another character. The flow then allows the transmitter FIFO to be filled whenever an interrupt occurs and there is data to transmit, rather than waiting until the FIFO is completely empty. Waiting until the FIFO is empty causes a reduction in performance whenever the system is too busy to respond immediately. Further system efficiency is achieved when this mode is enabled in combination with Auto Flow Control.

Even if everything else is selected and enabled, if the FIFOs are disabled using the FCR[0] bit, the Programmable THRE Interrupt mode is also disabled. When not selected or disabled, THRE interrupts and the LSR[5] bit function normally, signifying an empty THR or FIFO.

12.2.4.12 Clock Gate Enable

The UART can be configured to have a clock gate enable output.

- When the clock gate enable option is not selected, no logic is implemented, which reduces the overall gate count.
- When the clock gate enable option is selected, the clock gate enable signal(s)—uart_lp_req_pclk for single clock implementations or uart_lp_req_pclk and uart_lp_req_sclk for two clock implementations—is used to indicate the following:
 - Transmit and receive pipeline is clear (no data).
 - No activity has occurred.
 - Modem control input signals have not changed in more than one character time—the time taken to TX/RX a character—so that clocks can be gated.

A character is made up of:

start_bit + data_bits + parity (optional) + stop_bit(s))

The assertion of clock gate enable signals is an indication that the UART is inactive, so clocks may be gated in order to put the device in a low-power (lp) mode. Therefore, the following must be true for at least one character time for the assertion of the clock gate enable signal(s) to occur:

- No data in the RBR (in non-FIFO mode) or the RX FIFO is empty (in FIFO mode)
- No data in the THR (in non-FIFO mode) or the TX FIFO is empty (in FIFO mode)
- sin/sir_in and sout/sir_out_n are inactive (sin/sir_in are kept high and sout is high or sir_out_n is low) indicating no activity
- No change on the modem control input signals

Note, the clock gate enable assertion does not occur in the following modes of operation:

- Loopback mode
- FIFO access mode
- When transmitting a break

For example, assume a UART that is configured to have a single clock (pclk) and is programmed to transmit and receive characters of 7 bits (1 start bit, 5 data bits and 1 stop bit) and the baud clock divisor is set to 1. Therefore, the uart_lp_req_pclk signal is asserted if the transmit and receive pipeline is clear, no activity has occurred and the modem control input signals have not changed for 112 (7×16) pclk cycles. Figure 12-2-19 illustrates this example.

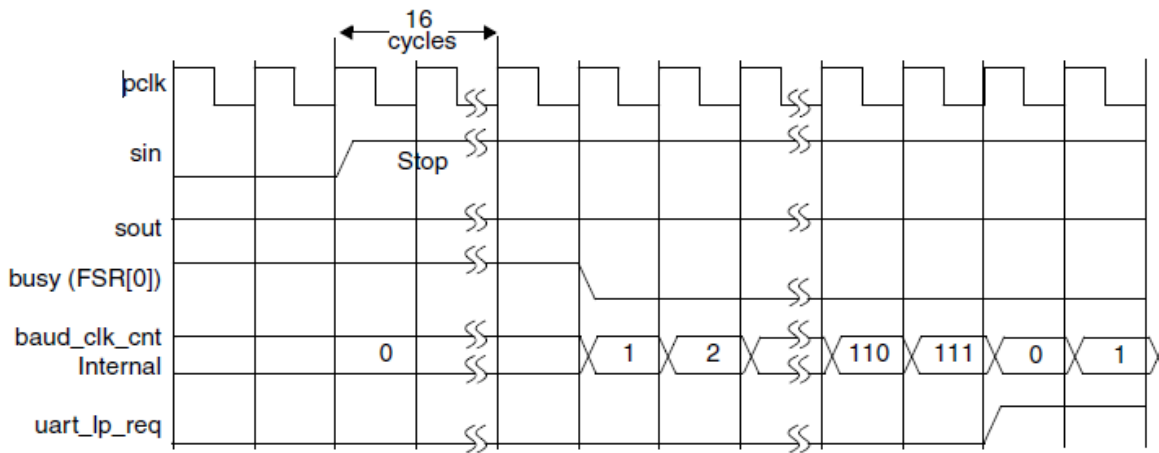


Figure 12-2-19 Clock Gate Enable Timing

When the assertion criteria are no longer met, the clock gate enable signal(s) are de-asserted and the clock(s) is resumed under any of these conditions:

- Either sin signal or sir_in signal goes low
- Write to any of registers is performed
- Modem control input signals have changed when UART is in low-power (sleep) mode

The clock gate enable signals are de-asserted asynchronously on arrival of above mentioned events because a clock is not available to synchronize the events. Therefore, user can decide to include 2-flop syncs externally before the clock gate cell to avoid metastability issues for clock-gate latch.

The time taken for the clock(s) to resume is important in preventing receive data synchronization problems, due to the UART RX block sampling:

1. At mid-point of each bit period—after approximately 8 baud clocks—in UART (RS323) mode.
2. After that, every 16 baud clocks for a baud divisor of 1 that is 16 sclks; for a single clock implementation, this is 16 pclks.

Thus, if eight or more sclk periods pass before the serial clock starts up again, the UART can get out of synchronization with the serial data it is receiving; that is, the receiver can sample into the second bit period, and if it is still 0, the receiver uses this as the start bit, and so on.

In order to avoid this problem, the clock should be resumed within five clock periods of the baud clock, which is the same as sclk if the baud divisor is set to 1; this is worst-case. If the divisor is greater, it gives a greater number of sclk cycles available before the clock must resume. This means a sample point at the 13 baud clock (at the latest) out of the 16 that are transmitted for each bit period of the character in non-SIR mode. Figure 3-26 shows the timing diagram that illustrates the previous scenario.

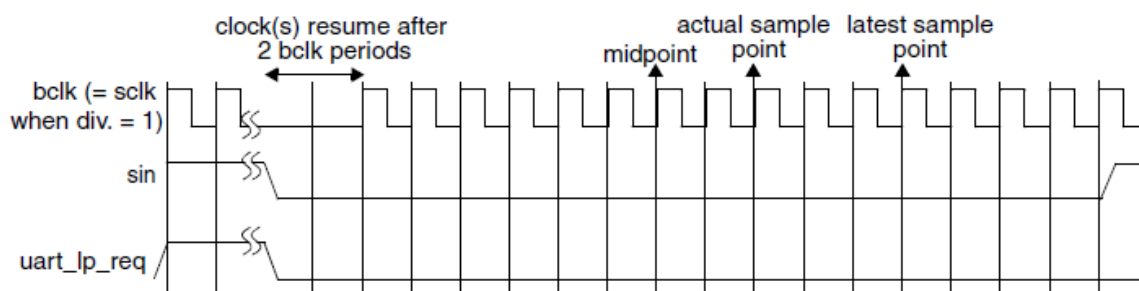


Figure 12-2-20 Resuming Clock(s) After Low Power Mode Timing

This synchronization problem is magnified in SIR mode because the pulse width is only 3/16 of a bit period—three baud clocks, which for a divisor of 1 is three sclks; thus, the pulse can be missed completely. The clocks must resume

before three baud clock periods elapse. However, if the first character received while in sleep mode is used only for wake-up reasons and the actual character value is unimportant, this may not become a problem.

When the UART is configured to have two clocks, if the timing of the received signal is not affected by the synchronization problem, then the minimum time to receive a character—if the baud divisor is 1—is 112 sclk:

$$1 \text{ start\_bit} + 5 \text{ data\_bits} + 1 \text{ stop\_bit} = 7 \times 16 = 112$$

Therefore, the pclk must be available before 112 sclk cycles pass in order for the received character to be synchronized to the pclk domain and stored in the RBR (in non-FIFO mode) or the RX FIFO (in FIFO mode).

12.2.4.13 DMA Support

The UART uses two DMA channels—one for transmit data and one for receive data. DMA supports single DMA data transfers at a time. the dma_tx_req_n signal:

- Goes active-low under the following conditions:
- When Transmitter Holding Register is empty
- When transmitter FIFO is at or below programmed threshold
- Goes inactive when:
- Transmitter FIFO is above threshold with

dma_rx_req_n signal:

- Goes active-low when single character is available in Receiver FIFO or Receive Buffer Register
- Goes inactive when Receive Buffer Register or Receiver FIFO are empty

12.2.4.13.1 DMA Modes

The UART uses two DMA channels—one for transmit data and one for receive data. There are two DMA modes:

- mode 0 – bit 3 of FIFO Control Register set to 0
- mode 1 – bit 3 of FIFO Control Register set to 1

DMA mode 0 supports single DMA data transfers at a time. In mode 0, the dma_tx_req_n signal:

- Goes active-low under the following conditions:
- When Transmitter Holding Register is empty in non-FIFO mode
- When transmitter FIFO is empty in FIFO mode with Programmable THRE interrupt mode disabled
- When transmitter FIFO is at or below programmed threshold with Programmable THRE interrupt mode enabled
- Goes inactive when:
- Single character has been written into Transmitter Holding Register or transmitter FIFO with Programmable THRE interrupt mode disabled
- Transmitter FIFO is above threshold with Programmable THRE interrupt mode enabled In mode 0, the dma_rx_req_n signal:
- Goes active-low when single character is available in Receiver FIFO or Receive Buffer Register
- Goes inactive when Receive Buffer Register or Receiver FIFO are empty, depending on FIFO mode

DMA mode 1 supports multi-DMA data transfers, where multiple transfers are made continuously until the receiver FIFO has been emptied or the transmit FIFO has been filled.

In mode 1, the dma_tx_req_n signal is asserted:

- When transmitter FIFO is empty with Programmable THRE interrupt mode disabled
- When transmitter FIFO is at or below programmed threshold with Programmable THRE interrupt mode enabled

In mode 1, the dma_tx_req_n signal is de-asserted when the transmitter FIFO is completely full. In mode 1, the dma_rx_req_n signal is asserted:

- When Receiver FIFO is at or above programmed trigger level
- When character timeout has occurred; ERBFI does not need to be set

In mode 1, the dma_rx_req_n signal is de-asserted when the receiver FIFO becomes empty. the dma_tx_req_n signal is asserted under the following conditions:

- When the Transmitter Holding Register is empty in non-FIFO mode
- When the transmitter FIFO is empty in FIFO mode with Programmable THRE interrupt mode disabled
- When the transmitter FIFO is at, or below the programmed threshold with Programmable THRE interrupt mode enabled.
- When configured for additional DMA signals, the dma_rx_req_n signal is asserted under the following conditions:
 - When a single character is available in Receive Buffer Register in non-FIFO mode
 - When Receiver FIFO is at or above programmed trigger level in FIFO mode

With the presence of the additional handshaking signals, the UART does not have to rely on internal status and level values to recognize the completion of a request and hence remove the request. Instead, the de-assertion of the DMA transmit and receive request is controlled by the assertion of the DMA transmit and receive acknowledge respectively. When the UART is configured for additional DMA signals, responsibility of the data flow (transfer lengths) falls on the DMA and is controlled by the programmed burst transaction lengths. Thus, there is no need for DMA modes, and programming the FCR[3] has no effect.

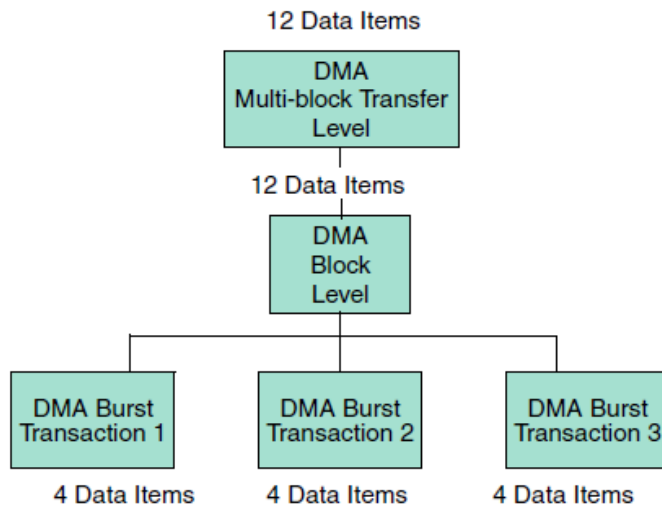
12.2.4.13.2 Example DMA Flow

The extra handshaking signals are explained in the following DMA flow for a UART that is configured with FIFOs and Programmable THRE interrupt mode.

As a block flow control device, the DMA Controller is programmed by the processor with the number of data items (block size) that are to be transmitted or received by the UART; this is programmed into the BLOCK_TS field of the CTLx register.

The block is broken into a number of transactions, each initiated by a request from the UART. The DMA Controller must also be programmed with the number of data items (in this case, UART FIFO entries) to be transferred for each DMA request. This is also known as the burst transaction length, and is programmed into the SRC_MSIZE/DEST_MSIZE fields of the DMA controller's register for source and destination, respectively.

Figure 12-2-21 shows a single block transfer, where the block size programmed into the DMA Controller is 12 and the burst transaction length is set to 4.



Block Size : `DMA.CTLx.BLOCK_TS=12`

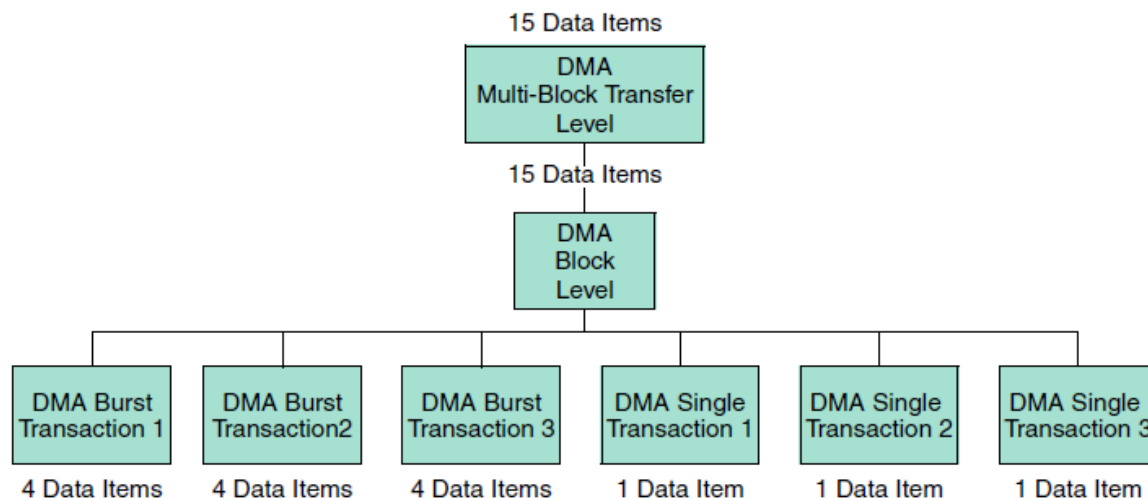
Number of data items per source burst transaction : `DMA.CTLx.SRC_MSIZ = 4`

For a FIFO depth of 16: `UART.FCR[7:6] = 01 = FIFO 1/4 full = DMA.CTLx.SRC_MSIZ`

Figure 12-2-21 Breakdown of DMA Transfer into Burst Transactions

In this case, the block size is a multiple of the burst transaction length. Therefore, the DMA block transfer consists of a series of burst transactions. If the UART makes a transmit request to this channel, four data items are written to the UART transmit FIFO. Similarly, if the UART makes a receive request to this channel, four data items are read from the UART receive FIFO. Three separate requests must be made to this DMA channel before all twelve data items are written or read.

When the block size programmed into the DMA Controller is not a multiple of the burst transaction length, as shown in Figure 12-2-22, a series of burst transactions followed by single transactions are needed to complete the block transfer.



Block Size : `DMA.CTLx.BLOCK_TS=15`

Number of data items per burst transaction : `DMA.CTLx.DEST_MSIZ = 4`

For a FIFO depth of 16: `UART.FCR[5:4] = 10 = FIFO 1/4 full = 4 = DMA.CTLx.DEST_MSIZ`

Figure 12-2-22 Breakdown of DMA Transfer into Single and Burst Transactions

12.2.4.14 Reset Signals

When configured for asynchronous serial clock operation, the UART includes two separate reset signals, each dedicated to its own clock domain:

- `presetn` resets logic in `pclk` clock domain
- `s_rst_n` resets logic in `sclk` clock domain

In order to avoid serious operational failures, both clock domains of the UART must be reset before any attempt is made to send or receive data on the serial line; that is, it is an illegal operation to reset just one clock domain of the UART without resetting the other clock domain.

Each reset signal must be de-asserted synchronously with the corresponding clock signal.

When asserting the reset signals, the `s_rst_n` signal should be asserted before or at the same time as `presetn`; this prevents any unexpected activity on the serial line that might result from resetting the programming registers without resetting the serial logic.

Similarly, when de-asserting the reset signals, `s_rst_n` should be de-asserted before `presetn` is de-asserted. The safest procedure for resetting UART is as follows:

1. Assert `s_rst_n` and `presetn`; the sequence of asserting these two signals and their timing relationship with `sclk` and `pclk` are not important
2. De-assert `s_rst_n` synchronously with `sclk`
3. De-assert `presetn` synchronously with `pclk`

Both reset signals should be active for at least three cycles of the respective clock signal.

12.2.5 Register Description

12.2.5.1 Register Summary

The UART base addresses are as follows:

- The base address for UART0 registers is 0x9140_0000.
- The base address for UART1 registers is 0x9140_1000.
- The base address for UART2 registers is 0x9140_2000.
- The base address for UART3 registers is 0x9140_3000.
- The base address for UART4 registers is 0x9140_4000.

Table 12-2-4 Summary of UART registers

Offset Address	Register	Description	Page
0x00	RBR	Receive Buffer Register. Reset Value: 0x0 This register can be accessed only when the DLAB bit...	12-2-5
0x00	DLL	Divisor Latch (Low). Reset Value: 0x0. If UART_16550_COMPATIBLE = No, then this register can be...	12-2-6
0x00	THR	Transmit Holding Register. Reset Value: 0x0. This register can be accessed only when the DLAB bit...	12-2-7
0x04	DLH	Divisor Latch High (DLH) Register. Reset Value: 0x0 If UART_16550_COMPATIBLE = No, then this register...	12-2-8
0x04	IER	Interrupt Enable Register. Reset Value: 0x0. This register can be accessed only when the DLAB...	12-2-9
0x08	FCR	This register is only valid when the UART is configured to have FIFO's implemented (FIFO_MODE...	12-2-10
0x08	IIR	Interrupt Identification Register Reset Value: 0x1	12-2-11
0x0c	LCR	Line Control Register Reset Value: 0x0	12-2-12

Offset Address	Register	Description	Page
0x10	MCR	Modem Control Register Reset Value: 0x0	12-2-13
0x14	LSR	Line Status Register Reset Value: 0x60	12-2-14
0x18	MSR	Whenever bits 0, 1, 2 or 3 is set to logic one, to indicate a change on the modem control inputs,...	12-2-15
0x1c	SCR	Scratchpad Register Reset Value: 0x0	12-2-16
0x20	LPDLL	Low Power Divisor Latch Low Register. This register is only valid when the UART is configured...	12-2-17
0x24	LPDLH	Low Power Divisor Latch High Register . This register is valid only when the UART is configured...	12-2-18
0x30 + n*0x4	SRBRn (for n = 0; n <= 15)	This register is valid only when the UART is configured to have additional shadow registers implemented...	12-2-19
0x30 + n*0x4	STHRn (for n = 0; n <= 15)	Shadow Transmit Holding Register. This register is valid only when the UART is configured...	12-2-20
0x70	FAR	FIFO Access Register	12-2-21
0x74	TFR	This register is valid only when the UART is configured to have the FIFO access test mode...	12-2-22
0x78	RFW	This register is valid only when the UART is configured to have the FIFO access test mode...	12-2-23
0x7c	USR	UART Status register. Reset Value: 0x6	12-2-24
0x80	TFL	TFL register is valid only when the UART is configured to have additional FIFO status registers implemented...	12-2-25
0x84	RFL	RFL register is valid only when the UART is configured to have additional FIFO status registers implemented...	12-2-26
0x88	SRR	This register is valid only when the UART is configured to have additional shadow registers implemented...	12-2-27
0x8c	SRTS	SRTS register is valid only when the UART is configured to have additional shadow registers implemented...	12-2-28
0x90	SBCR	SBCR register is valid only when the UART is configured to have additional shadow registers implemented...	12-2-29
0x94	SDMAM	This register is valid only when the UART is configured to have additional FIFO registers...	12-2-30

Offset Address	Register	Description	Page
0x98	SFE	SFE register is valid only when the UART is configured to have additional FIFO registers implemented...	12-2-31
0x9c	SRT	SRT register is valid only when the UART is configured to have additional FIFO registers implemented...	12-2-32
0xa0	STET	This register is valid only when the UART is configured to have FIFOs implemented (FIFO_MODE...	12-2-33
0xa4	HTX	Halt TX Reset Value: 0x0	12-2-34
0xa8	DMA_SA	DMA Software Acknowledge Register Reset Value: 0x0	12-2-35
0xac	TCR	This register is used to enable or disable RS485 mode and also control the polarity values for Driven...	12-2-36
0xb0	DE_EN	The Driver Output Enable Register (DE_EN) is used to control the assertion and de-assertion of the...	12-2-37
0xb4	RE_EN	The Receiver Output Enable Register (RE_EN) is used to control the assertion and de-assertion of...	12-2-38
0xb8	DET	The Driver Output Enable Timing Register (DET) is used to control the DE assertion and de-assertion...	12-2-39
0xbc	TAT	The TurnAround Timing Register (TAT) is used to hold the turnaround time between switching of 're'...	12-2-40
0xc0	DLF	This register is only valid when the UART is configured to have Fractional Baud rate Divisor...	12-2-41
0xc4	RAR	Receive Address Register Reset Value: 0x0	12-2-42
0xc8	TAR	Transmit Address Register Reset Value: 0x0	12-2-43
0xcc	LCR_EXT	Line Extended Control Register Reset Value: 0x0	12-2-44
0xf4	CPR	Component Parameter Register. This register is valid only when UART_ADD_ENCODED_PARAMS = 1. If the...	12-2-45
0xf8	UCV	This register is valid only when the UART is configured to have additional features implemented...	12-2-46
0xfc	CTR	This register is valid only when the UART is configured to have additional features implemented...	12-2-47

12.2.5.2 Register Description

12.2.5.2.1 RBR

- Name: Receive Buffer Register
- Description: Receive Buffer Register. This register can be accessed only when the DLAB bit (LCR[7]) is cleared.

- Reset Value: 0x0
- Offset Address: 0x00

Table 12-2-5 Fields for Register: RBR

Bits	Access	Name	Description & Reset Value
x:y	R	RSVD_RBR	RBR 31to9or8 Reserved bits and read as zero (0). Exists: Always Range Variable[x]: (UART_9BIT_DATA_EN==1) ? 23 : 24 + (UART_9BIT_DATA_EN==1) ? 0x9 : 0x8 - 1 Range Variable[y]: (UART_9BIT_DATA_EN==1) ? 0x9 : 0x8
x:0	R	RBR	Receive Buffer Register. This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If in non-FIFO mode (FIFO_MODE == NONE) or FIFOs are disabled (FCR[0] set to 0), the data in the RBR must be read before the next data arrives, otherwise it will be overwritten, resulting in an over-run error. If in FIFO mode (FIFO_MODE != NONE) and FIFOs are enabled (FCR[0] set to 1), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO will be preserved but any incoming data will be lost and an over-run error occurs. Note: When UART_9BIT_DATA_EN=0, this field width is 8. When UART_9BIT_DATA_EN=1, this field width is 9. Reset Value: 0x0 Exists: Always Range Variable[x]: (UART_9BIT_DATA_EN==1) ? 9 : 8 - 1

12.2.5.2.2 DLL

- Name: DivisorLatch(Low).
- Description: Divisor Latch (Low). If UART_16550_COMPATIBLE = No, then this register can be accessed only when the DLAB bit (LCR[7]) is set and the UART is not busy - that is, USR[0] is 0; otherwise this register can be accessed only when the DLAB bit (LCR[7]) is set.
- Reset Value: 0x0.
- Offset Address: 0x00

Table 12-2-6 Fields for Register: DLL

Bits	Access	Name	Description & Reset Value
31:8	R	RSVD_DLL_31to8	DLL 31to8 Reserved bits and read as zero (0). Exists: Always

Bits	Access	Name	Description & Reset Value
7:0	R/W	DLL	Divisor Latch (Low). This register makes up the lower 8-bits of a 16-bit, read/ write, Divisor Latch register that contains the baud rate divisor for the UART. The output baud rate is equal to the serial clock (pclk if one clock design, sclk if two clock design (CLOCK_MODE == Enabled)) frequency divided by sixteen times the value of the baud rate divisor, as follows: baud rate = (serial clock freq) / (16 * divisor). Note that with the Divisor Latch Registers (DLL and DLH) set to zero, the baud clock is disabled and no serial communications will occur. Also, once the DLL is set, at least 8 clock cycles of the slowest DW_apb_uart clock should be allowed to pass before transmitting or receiving data. Reset Value: 0x0 Exists: Always

12.2.5.2.3 THR

- Name: TransmitHoldingRegister.
- Description: Transmit Holding Register. This register can be accessed only when the DLAB bit (LCR[7]) is cleared.
- Reset Value: 0x0.
- Offset Address: 0x00

Table 12-2-7 Fields for Register: THR

Bits	Access	Name	Description & Reset Value
x:y	R	RSVD_THR	THR 31to9or8 Reserved bits and read as zero (0). Exists: Always Range Variable[x]: (UART_9BIT_DATA_EN==1) ? 23 : 24 + (UART_9BIT_DATA_EN==1) ? 0x9 : 0x8 - 1 Range Variable[y]: (UART_9BIT_DATA_EN==1) ? 0x9 : 0x8
x:0	W	THR	Transmit Holding Register. This register contains data to be transmitted on the serial output port (sout) in UART mode or the serial infrared output (sir_out_n) in infrared mode. Data should only be written to the THR when the THR Empty (THRE) bit (LSR[5]) is set. If in non-FIFO mode or FIFO's are disabled (FCR[0] set to zero) and THRE is set, writing a single character to the THR clears the THRE. Any additional writes to the THR before the THRE is set again causes the THR data to be overwritten. If in FIFO mode and FIFO's are enabled (FCR[0] set to one) and THRE is set, x number of characters of data may be written to the THR before the FIFO is full. The number x (default=16) is determined by the value of FIFO Depth that is set during configuration. Any attempt to write data when the FIFO is full results in the write data being lost. Note: When UART_9BIT_DATA_EN=0, this field width is 8. When UART_9BIT_DATA_EN=1, this field width is 9. The 9th bit is applicable only when LCR_EXT[3]=1. Reset Value: 0x0 Exists: Always Range Variable[x]: (UART_9BIT_DATA_EN==1) ? 9 : 8 - 1

12.2.5.2.4 DLH

- Name: DivisorLatchHigh(DLH)Register.
- Description: Divisor Latch High (DLH) Register. If UART_16550_COMPATIBLE = No, then this register can be accessed only when the DLAB bit (LCR[7]) is set and the UART is not busy, that is, USR[0] is 0; otherwise this register can be accessed only when the DLAB bit (LCR[7]) is set.
- Reset Value: 0x0.
- Offset Address: 0x04

Table 12-2-8 Fields for Register: DLH

Bits	Access	Name	Description & Reset Value
31:8	R	RSVD_DLH	DLH 31to8 Reserved bits and read as zero (0). Exists: Always
7:0	R/W	dlh	Upper 8-bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. The output baud rate is equal to the serial clock (pclk if one clock design, sclk if two clock design (CLOCK_MODE == Enabled)) frequency divided by sixteen times the value of the baud rate divisor, as follows: baud rate = (serial clock freq) / (16 * divisor). Note that with the Divisor Latch Registers (DLL and DLH) set to zero, the baud clock is disabled and no serial communications will occur. Also, once the DLH is set, at least 8 clock cycles of the slowest DW_apb_uart clock should be allowed to pass before transmitting or receiving data. Exists: Always

12.2.5.2.5 IER

- Name: InterruptEnableRegister.
- Description: Interrupt Enable Register. This register can be accessed only when the DLAB bit (LCR[7]) is cleared.
- Reset Value: 0x0.
- Offset Address: 0x04

Table 12-2-9 Fields for Register: IER

Bits	Access	Name	Description & Reset Value
31:8	R	RSVD_IER_31to8	IER 31to8 Reserved bits and read as zero (0). Exists: Always
7	*Varies	PTIME	Programmable THRE Interrupt Mode Enable. Writeable only when THRE_MODE_USER == Enabled, always readable. This is used to enable/disable the generation of THRE Interrupt. Reset Value: 0x0 Values: 0x0 (DISABLED): Disable Programmable THRE Interrupt Mode 0x1 (ENABLED): Enable Programmable THRE Interrupt Mode Exists: Always Memory Access: (THRE_MODE_USER==1 && FIFO_MODE!=0) ? read-write : read-only
6:5	R	RSVD_IER_6to5	IER 6to5 Reserved bits read as zero (0). Exists: Always

Bits	Access	Name	Description & Reset Value
4	*Varies	ELCOLR	Interrupt Enable Register: ELCOLR, this bit controls the method for clearing the status in the LSR register. This is applicable only for Overrun Error, Parity Error, Framing Error, and Break Interrupt status bits. 0 = LSR status bits are cleared either on reading Rx FIFO (RBR Read) or On reading LSR register. 1 = LSR status bits are cleared only on reading LSR register. Writeable only when LSR_STATUS_CLEAR == Enabled, always readable. Values: 0x0 (DISABLED): Disable ALC 0x1 (ENABLED): Enable ALC Exists: Always Memory Access: (LSR_STATUS_CLEAR==1) ? read- write : read-only
3	R/W	EDSSI	Enable Modem Status Interrupt. This is used to enable/ disable the generation of Modem Status Interrupt. This is the fourth highest priority interrupt. Reset Value: 0x0 Values: 0x0 (DISABLED): Disable Modem Status Interrupt 0x1 (ENABLED): Enable Modem Status Interrupt Exists: Always
2	R/W	ELSI	Enable Receiver Line Status Interrupt. This is used to enable/disable the generation of Receiver Line Status Interrupt. This is the highest priority interrupt. Reset Value: 0x0 Values: 0x0 (DISABLED): Disable Receiver Line Status Interrupt 0x1 (ENABLED): Enable Receiver Line Status Interrupt Exists: Always
1	R/W	ETBEI	Enable Transmit Holding Register Empty Interrupt. This is used to enable/disable the generation of Transmitter Holding Register Empty Interrupt. This is the third highest priority interrupt. Reset Value: 0x0 Values: 0x0 (DISABLED): Disable Transmit empty interrupt 0x1 (ENABLED): Enable Transmit empty interrupt Exists: Always
0	R/W	ERBFI	Enable Received Data Available Interrupt. This is used to enable/disable the generation of Received Data Available Interrupt and the Character Timeout Interrupt (if in FIFO mode and FIFO's enabled). These are the second highest priority interrupts. Reset Value: 0x0 Values: 0x0 (DISABLED): Disable Receive data Interrupt 0x1 (ENABLED): Enable Receive data Interrupt Exists: Always

12.2.5.2.6 FCR

- Name: FIFOControlRegister
- Description: FIFO Control Register This register is only valid when the UART is configured to have FIFO's implemented (FIFO_MODE != NONE). If FIFO's are not implemented, this register does not exist and writing to this register address will have no effect.
- Reset Value: 0x0.
- Offset Address: 0x08

Table 12-2-10 Fields for Register: FCR

Bits	Access	Name	Description & Reset Value
31:8	R	RSVD_FCR_31to8	FCR 31to8 Reserved bits and read as 0. Exists: Always
7:6	W	RT	RCVR Trigger (or RT). This is used to select the trigger level in the receiver FIFO at which the Received Data Available Interrupt will be generated. In auto flow control mode, it is used to determine when the rts_n signal will be de-asserted only when RTC_FCT is disabled. It also determines when the dma_rx_req_n signal will be asserted when in certain modes of operation. For details on DMA support, refer to 'DMA Support' section of data book. Reset Value: 0x0 Values: 0x0 (FIFO_CHAR_1): 1 character in FIFO 0x1 (FIFO_QUARTER_FULL): FIFO 1/4 full 0x2 (FIFO_HALF_FULL): FIFO 1/2 full 0x3 (FIFO_FULL_2): FIFO 2 less than full Exists: Always
5:4	*Varies	TET	TX Empty Trigger (or TET). Writes will have no effect when THRE_MODE_USER == Disabled. This is used to select the empty threshold level at which the THRE Interrupts will be generated when the mode is active. It also determines when the dma_tx_req_n signal will be asserted when in certain modes of operation. For details on DMA support, refer to 'DMA Support' section of data book. Reset Value: 0x0 Values: 0x0 (FIFO_EMPTY): FIFO Empty 0x1 (FIFO_CHAR_2): 2 characters in FIFO 0x2 (FIFO_QUARTER_FULL): FIFO 1/4 full 0x3 (FIFO_HALF_FULL): FIFO 1/2 full Exists: Always Memory Access: (THRE_MODE_USER==1) ? write- only : read-only
3	W	DMAM	DMA Mode (or DMAM). This determines the DMA signalling mode used for the dma_tx_req_n and dma_rx_req_n output signals when additional DMA handshaking signals are not selected (DMA_EXTRA == NO). For details on DMA support, refer to 'DMA Support' section of data book. Reset Value: 0x0 Values: 0x0 (MODE0): Mode 0 0x1 (MODE1): Mode 1 Exists: Always

Bits	Access	Name	Description & Reset Value
2	W	XFIFOR	XMIT FIFO Reset (or XFIFOR). This resets the control portion of the transmit FIFO and treats the FIFO as empty. This will also de-assert the DMA TX request and single signals when additional DMA handshaking signals are selected (DMA_EXTRA == YES). Note that this bit is 'self-clearing' and it is not necessary to clear this bit. Reset Value: 0x0 Values: 0x1 (RESET): Transmit FIFO reset Exists: Always
1	W	RFIFOR	RCVR FIFO Reset (or RFIFOR). This resets the control portion of the receive FIFO and treats the FIFO as empty. This will also de-assert the DMA RX request and single signals when additional DMA handshaking signals are selected (DMA_EXTRA == YES). Note that this bit is 'self-clearing' and it is not necessary to clear this bit. Reset Value: 0x0 Values: 0x1 (RESET): Receive FIFO reset Exists: Always
0	W	FIFOE	FIFO Enable (or FIFOE). This enables/disables the transmit (XMIT) and receive (RCVR) FIFOs. Whenever the value of this bit is changed both the XMIT and RCVR controller portion of FIFOs is reset. Reset Value: 0x0 Values: 0x0 (DISABLED): FIFO disabled 0x1 (ENABLED): FIFO enabled Exists: Always

12.2.5.2.7 IIR

- Name: InterruptIdentificationRegister
- Description: Interrupt Identification Register
- Reset Value: 0x1.
- Offset Address: 0x08

Table 12-2-11 Fields for Register: IIR

Bits	Access	Name	Description & Reset Value
31:8	R	RSVD_IIR_31to8	IIR 31to8 Reserved bits and read as 0. Exists: Always
7:6	R	FIFOSE	FIFOs Enabled (or FIFOSE). This is used to indicate whether the FIFOs are enabled or disabled. Reset Value: 0x0 Values: 0x0 (DISABLED): FIFOs are disabled 0x3 (ENABLED): FIFOs are enabled Exists: Always

Bits	Access	Name	Description & Reset Value
5:4	R	RSVD_IIR_5to4	IIR 5to4 Reserved bits read as 0. Exists: Always
3:0	R	IID	Interrupt ID (or IID). This indicates the highest priority pending interrupt which can be one of the following types specified in Values. For information on several levels into which the interrupt priorities are split into, see the 'Interrupts' section in the DW_apb_uart Databook. Note: an interrupt of type 0111 (busy detect) will never get indicated if UART_16550_COMPATIBLE == YES in coreConsultant. Reset Value: 0x1 Bit 3 indicates an interrupt can only occur when the FIFOs are enabled and used to distinguish a Character Timeout condition interrupt. Values: 0x0 (MODEM_STATUS): modem status 0x1 (NO_INTERRUPT_PENDING): no interrupt pending 0x2 (THR_EMPTY): THR empty 0x4 (RECEIVED_DATA_AVAILABLE): received data available 0x6 (RECEIVER_LINE_STATUS): receiver line status 0x7 (BUSY_DETECT): busy detect 0xc (CHARACTER_TIMEOUT): character timeout Exists: Always

12.2.5.2.8 LCR

- Name: LineControlRegister
- Description: Line Control Register
- Reset Value: 0x0.
- Offset Address: 0x0c

Table 12-2-12 Fields for Register: LCR

Bits	Access	Name	Description & Reset Value
31:8	R	RSVD_LCR_31to8	LCR 31to8 Reserved bits and read as 0. Exists: Always
7	R/W	DLAB	Divisor Latch Access Bit. If UART_16550_COMPATIBLE == NO then, writeable only when UART is not busy (USR[0] is zero), otherwise always writable and always readable. This bit is used to enable reading and writing of the Divisor Latch register (DLL and DLH/LPDLL and LPDLH) to set the baud rate of the UART. This bit must be cleared after initial baud rate setup in order to access other registers. Reset Value: 0x0 Values: 0x0 (DISABLED): Divisor Latch register is writable only when UART Not BUSY 0x1 (ENABLED): Divisor Latch register is always readable and writable Exists: Always

Bits	Access	Name	Description & Reset Value
6	R/W	BC	Break Control Bit. This is used to cause a break condition to be transmitted to the receiving device. If set to one the serial output is forced to the spacing (logic 0) state. When not in Loopback Mode, as determined by MCR[4], the sout line is forced low until the Break bit is cleared. If SIR_MODE == Enabled and active (MCR[6] set to one) the sir_out_n line is continuously pulsed. When in Loopback Mode, the break condition is internally looped back to the receiver and the sir_out_n line is forced low. Reset Value: 0x0 Values: 0x0 (DISABLED): Serial output is released for data transmission 0x1 (ENABLED): Serial output is forced to spacing state Exists: Always
5	R/W	SP	Stick Parity. If UART_16550_COMPATIBLE = NO, then writeable only when UART is not busy (USR[0] is 0); otherwise always writable and always readable. This bit is used to force parity value. When PEN, EPS and Stick Parity are set to 1, the parity bit is transmitted and checked as logic 0. If PEN and Stick Parity are set to 1 and EPS is a logic 0, then parity bit is transmitted and checked as a logic 1. If this bit is set to 0, Stick Parity is disabled. Reset Value: 0x0 Values: 0x0 (DISABLED): Stick parity disabled 0x1 (ENABLED): Stick parity enabled Exists: Always
4	R/W	EPS	Even Parity Select. If UART_16550_COMPATIBLE == NO then, writeable only when UART is not busy (USR[0] is zero), otherwise always writable and always readable. This is used to select between even and odd parity, when parity is enabled (PEN set to one). If set to one, an even number of logic '1's is transmitted or checked. If set to zero, an odd number of logic '1's is transmitted or checked. Reset Value: 0x0 Values: 0x0 (ODD_PARITY): an odd parity is transmitted or checked 0x1 (EVEN_PARITY): an even parity is transmitted or checked Exists: Always
3	R/W	PEN	Parity Enable. If UART_16550_COMPATIBLE == NO then, writeable only when UART is not busy (USR[0] is zero), otherwise always writable and always readable. This bit is used to enable and disable parity generation and detection in transmitted and received serial character respectively. Reset Value: 0x0 Values: 0x0 (DISABLED): disable parity 0x1 (ENABLED): enable parity Exists: Always

Bits	Access	Name	Description & Reset Value
2	R/W	STOP	Number of stop bits. If UART_16550_COMPATIBLE == NO then, writeable only when UART is not busy (USR[0] is zero), otherwise always writable and always readable. This is used to select the number of stop bits per character that the peripheral will transmit and receive. If set to zero, one stop bit is transmitted in the serial data. If set to one and the data bits are set to 5 (LCR[1:0] set to zero) one and a half stop bits is transmitted. Otherwise, two stop bits are transmitted. Note that regardless of the number of stop bits selected the receiver will only check the first stop bit. Note: NOTE: The STOP bit duration implemented by DW_apb_uart may appear longer due to idle time inserted between characters for some configurations and baud clock divisor values in the transmit direction; for details on idle time between transmitted transfers, refer to 'Back-to-Back Character Stream Transmission' section in data book. Reset Value: 0x0 Values: 0x0 (STOP_1BIT): 1 stop bit 0x1 (STOP_1_5BIT_OR_2BIT): 1.5 stop bits when DLS (LCR[1:0]) is zero, else 2 stop bit Exists: Always
1:0	R/W	DLS	Data Length Select (or CLS as used in legacy). If UART_16550_COMPATIBLE == NO then, writeable only when UART is not busy (USR[0] is zero), otherwise always writable and always readable. When DLS_E in LCR_EXT is set to 0, this register is used to select the number of data bits per character that the peripheral will transmit and receive. Reset Value: 0x0 Values: 0x0 (CHAR_5BITS): 5 data bits per character 0x1 (CHAR_6BITS): 6 data bits per character 0x2 (CHAR_7BITS): 7 data bits per character 0x3 (CHAR_8BITS): 8 data bits per character Exists: Always

12.2.5.2.9 MCR

- Name: ModemControlRegister
- Description: Modem Control Register
- Reset Value: 0x0.
- Offset Address: 0x10

Table 12-2-13 Fields for Register: MCR

Bits	Access	Name	Description & Reset Value
31:7	R	RSVD_MCR_31to7	MCR 31to7 Reserved bits read as 0. Exists: Always

Bits	Access	Name	Description & Reset Value
6	*Varies	SIRE	SIR Mode Enable. Writeable only when SIR_MODE == Enabled, always readable. This is used to enable/ disable the IrDA SIR Mode features as described in section 'IrDA 1.0 SIR Protocol' in the databook. Note: To enable SIR mode, write the appropriate value to the MCR register before writing to the LCR register. For details of the recommended programming sequence, refer to 'Programing Examples' section of data book. Reset Value: 0x0 Values: 0x0 (DISABLED): IrDA SIR Mode disabled 0x1 (ENABLED): IrDA SIR Mode enabled Exists: Always Memory Access: (SIR_MODE==1) ? read-write : read- only
5	*Varies	AFCE	Auto Flow Control Enable. Writeable only when AFCE_MODE == Enabled, always readable. When FIFOs are enabled and the Auto Flow Control Enable (AFCE) bit is set, Auto Flow Control features are enabled as described in section 'Auto Flow Control' in data book. Reset Value: 0x0 Values: 0x0 (DISABLED): Auto Flow Control Mode disabled 0x1 (ENABLED): Auto Flow Control Mode enabled Exists: Always Memory Access: (AFCE_MODE==1) ? read-write : read-only
4	R/W	LoopBack	LoopBack Bit. This is used to put the UART into a diagnostic mode for test purposes. If operating in UART mode (SIR_MODE != Enabled OR NOT active, MCR[6] set to zero), data on the sout line is held high, while serial data output is looped back to the sin line, internally. In this mode all the interrupts are fully functional. Also, in loopback mode, the modem control inputs (dsr_n, cts_n, ri_n, dcd_n) are disconnected and the modem control outputs (dtr_n, rts_n, out1_n, out2_n) are looped back to the inputs, internally. If operating in infrared mode (SIR_MODE == Enabled AND active, MCR[6] set to one), data on the sir_out_n line is held low, while serial data output is inverted and looped back to the sir_in line. Reset Value: 0x0 Values: 0x0 (DISABLED): Loopback mode disabled 0x1 (ENABLED): Loopback mode enabled Exists: Always

Bits	Access	Name	Description & Reset Value
3	R/W	OUT2	OUT2. This is used to directly control the user-designated Output2 (out2_n) output. The value written to this location is inverted and driven out on out2_n. Note that in Loopback mode (MCR[4] set to one), the out2_n output is held inactive high while the value of this location is internally looped back to an input. Reset Value: 0x0 Values: 0x0 (OUT2_0): out2_n de-asserted (logic 1) 0x1 (OUT2_1): out2_n asserted (logic 0) Exists: Always
2	R/W	OUT1	OUT1. This is used to directly control the user-designated Output1 (out1_n) output. The value written to this location is inverted and driven out on out1_n. Note that in Loopback mode (MCR[4] set to one), the out1_n output is held inactive high while the value of this location is internally looped back to an input. Reset Value: 0x0 Values: 0x0 (OUT1_0): out1_n de-asserted (logic 1) 0x1 (OUT1_1): out1_n asserted (logic 0) Exists: Always
1	R/W	RTS	Request to Send. This is used to directly control the Request to Send (rts_n) output. The Request To Send (rts_n) output is used to inform the modem or data set that the UART is ready to exchange data. When Auto RTS Flow Control is not enabled (MCR[5] set to zero), the rts_n signal is set low by programming MCR[1] (RTS) to a high. In Auto Flow Control, AFCE_MODE == Enabled and active (MCR[5] set to one) and FIFO's enable (FCR[0] set to one), the rts_n output is controlled in the same way, but is also gated with the receiver FIFO threshold trigger (rts_n is inactive high when above the threshold). The rts_n signal will be de-asserted when MCR[1] is set low. Note that in Loopback mode (MCR[4] set to one), the rts_n output is held inactive high while the value of this location is internally looped back to an input. Reset Value: 0x0 Values: 0x0 (INACTIVE): Request to Send rts_n de-asserted (logic 1) 0x1 (ACTIVE): Request to Send rts_n asserted (logic 0) Exists: Always

Bits	Access	Name	Description & Reset Value
0	R/W	DTR	Data Terminal Ready. This is used to directly control the Data Terminal Ready (dtr_n) output. The value written to this location is inverted and driven out on dtr_n. The Data Terminal Ready output is used to inform the modem or data set that the UART is ready to establish communications. Note that in Loopback mode (MCR[4] set to one), the dtr_n output is held inactive high while the value of this location is internally looped back to an input. Reset Value: 0x0 Values: 0x0 (INACTIVE): dtr_n de-asserted (logic1) 0x1 (ACTIVE): dtr_n asserted (logic 0) Exists: Always

12.2.5.2.10 LSR

- Name: LineStatusRegister
- Description: Line Status Register
- Reset Value: 0x60.
- Offset Address: 0x14

Table 12-2-14 Fields for Register: LSR

Bits	Access	Name	Description & Reset Value
31:9	R	RSVD_LSR_31to9	LSR 31to9 Reserved bits read as zero. Exists: Always
8	R	ADDR_RCVD	Address Received Bit. If 9Bit data mode (LCR_EXT[0]=1) is enabled, this bit is used to indicate the 9th bit of the receive data is set to 1. This bit can also be used to indicate whether the incoming character is address or data. 1 = Indicates the character is address. 0 = Indicates the character is data. In the FIFO mode, since the 9th bit is associated with a character received, it is revealed when the character with the 9th bit set to 1 is at the top of the FIFO. Reading the LSR clears the 9BIT. Note: User needs to ensure that interrupt gets cleared (reading LSR register) before the next address byte arrives. If there is a delay in clearing the interrupt, then Software will not be able to distinguish between multiple address related interrupt. Reset Value: 0x0 Exists: UART_9BIT_DATA_EN == 1

Bits	Access	Name	Description & Reset Value
7	R	RFE	Receiver FIFO Error bit. This bit is only relevant when FIFO_MODE != NONE AND FIFO's are enabled (FCR[0] set to one). This is used to indicate if there is at least one parity error, framing error, or break indication in the FIFO. This bit is cleared when the LSR is read and the character with the error is at the top of the receiver FIFO and there are no subsequent errors in the FIFO. Reset Value: 0x0 Values: 0x0 (NO_RX_FIFO_ERROR): No error in RX FIFO 0x1 (RX_FIFO_ERROR): Error in RX FIFO Exists: FIFO_MODE != 0
6	R	TEMT	Transmitter Empty bit. If in FIFO mode (FIFO_MODE != NONE) and FIFO's are enabled (FCR[0] set to one), this bit is set whenever the Transmitter Shift Register and the FIFO are both empty. If in the non-FIFO mode or FIFO's are disabled, this bit is set whenever the Transmitter Holding Register and the Transmitter Shift Register are both empty. Reset Value: 0x1 Values: 0x0 (DISABLED): Transmitter not empty 0x1 (ENABLED): Transmitter empty Exists: Always
5	R	THRE	Transmit Holding Register Empty bit. If THRE_MODE_USER = Disabled or THRE mode is disabled (IER[7] set to zero) and regardless of FIFO's being implemented/enabled or not, this bit indicates that the THR or TX FIFO is empty. This bit is set whenever data is transferred from the THR or TX FIFO to the transmitter shift register and no new data has been written to the THR or TX FIFO. This also causes a THRE Interrupt to occur, if the THRE Interrupt is enabled. If THRE_MODE_USER == Enabled AND FIFO_MODE != NONE and both modes are active (IER[7] set to one and FCR[0] set to one respectively), the functionality is switched to indicate the transmitter FIFO is full, and no longer controls THRE interrupts, which are then controlled by the FCR[5:4] threshold setting. Programmable THRE interrupt mode operation is described in detail in section 'Programmable THRE Interrupt' in data book. Reset Value: 0x1 Values: 0x0 (DISABLED): THRE interrupt control is disabled 0x1 (ENABLED): THRE interrupt control is enabled Exists: Always

Bits	Access	Name	Description & Reset Value
4	R	BI	Break Interrupt bit. This is used to indicate the detection of a break sequence on the serial input data. If in UART mode it is set whenever the serial input, sin, is held in a logic '0' state for longer than the sum of start time + data bits + parity + stop bits. If in infrared mode it is set whenever the serial input, sir_in, is continuously pulsed to logic '0' for longer than the sum of start time + data bits + parity + stop bits. A break condition on serial input causes one and only one character, consisting of all zeros, to be received by the UART. In the FIFO mode, the character associated with the break condition is carried through the FIFO and is revealed when the character is at the top of the FIFO. Reading the LSR clears the BI bit (if LSR_STATUS_CLEAR==1) Or Reading the LSR or RBR clears the BI bit (if LSR_STATUS_CLEAR==0). In the non-FIFO mode, the BI indication occurs immediately and persists until the LSR is read. Note: If a FIFO is full when a break condition is received, a FIFO overrun occurs. The break condition and all the information associated with it-parity and framing errors-is discarded; any information that a break character was received is lost. Reset Value: 0x0 Values: 0x0 (NO_BREAK): No break sequence detected 0x1 (BREAK): Break sequence detected Exists: Always
3	R	FE	Framing Error bit. This is used to indicate the occurrence of a framing error in the receiver. A framing error occurs when the receiver does not detect a valid STOP bit in the received data. In the FIFO mode, since the framing error is associated with a character received, it is revealed when the character with the framing error is at the top of the FIFO. When a framing error occurs the UART will try resynchronize. It does this by assuming that the error was due to the start bit of the next character and then continues receiving the other bit i.e. data, and/or parity and stop. It should be noted that the Framing Error (FE) bit (LSR[3]) will be set if a break interrupt has occurred, as indicated by Break Interrupt (BI) bit (LSR[4]). This happens because the break character implicitly generates a framing error by holding the sin input to logic 0 for longer than the duration of a character. Reading the LSR clears the FE bit (if LSR_STATUS_CLEAR==1) Or Reading the LSR or RBR clears the FE bit (if LSR_STATUS_CLEAR==0). Reset Value: 0x0 Values: 0x0 (NO_FRAMING_ERROR): no framing error 0x1 (FRAMING_ERROR): framing error Exists: Always

Bits	Access	Name	Description & Reset Value
2	R	PE	Parity Error bit. This is used to indicate the occurrence of a parity error in the receiver if the Parity Enable (PEN) bit (LCR[3]) is set. In the FIFO mode, since the parity error is associated with a character received, it is revealed when the character with the parity error arrives at the top of the FIFO. It should be noted that the Parity Error (PE) bit (LSR[2]) will be set if a break interrupt has occurred, as indicated by Break Interrupt (BI) bit (LSR[4]). In this situation, the Parity Error bit is set if parity generation and detection is enabled (LCR[3]=1) and the parity is set to odd (LCR[4]=0). Reading the LSR clears the PE bit (if LSR_STATUS_CLEAR==1) Or Reading the LSR or RBR clears the PE bit (if LSR_STATUS_CLEAR==0). Reset Value: 0x0 Values: 0x0 (NO_PARITY_ERROR): no parity error 0x1 (PARITY_ERROR): parity error Exists: Always
1	R	OE	Overrun error bit. This is used to indicate the occurrence of an overrun error. This occurs if a new data character was received before the previous data was read. In the non-FIFO mode, the OE bit is set when a new character arrives in the receiver before the previous character was read from the RBR. When this happens, the data in the RBR is overwritten. In the FIFO mode, an overrun error occurs when the FIFO is full and a new character arrives at the receiver. The data in the FIFO is retained and the data in the receive shift register is lost. Reading the LSR clears the OE bit (if LSR_STATUS_CLEAR==1) Or Reading the LSR or RBR clears the OE bit (if LSR_STATUS_CLEAR==0). Reset Value: 0x0 Values: 0x0 (NO_OVER_RUN_ERROR): no overrun error 0x1 (OVER_RUN_ERROR): overrun error Exists: Always
0	R	DR	Data Ready bit. This is used to indicate that the receiver contains at least one character in the RBR or the receiver FIFO. This bit is cleared when the RBR is read in the non-FIFO mode, or when the receiver FIFO is empty, in the FIFO mode. Reset Value: 0x0 Values: 0x0 (NOT_READY): data not ready 0x1 (READY): data ready Exists: Always

12.2.5.2.11 MSR

- Name: ModemStatusRegister
- Description: Modem Status Register Whenever bits 0, 1, 2 or 3 is set to logic one, to indicate a change on the modem control inputs, a modem status interrupt will be generated if enabled via the IER regardless of when the change occurred. The bits (bits 0, 1, 3) can be set after a reset-even though their respective modem signals are inactive-because the synchronized version of the modem signals have a reset value of 0 and change to value 1

after reset. To prevent unwanted interrupts due to this change, a read of the MSR register can be performed after reset.

- Reset Value: 0x0.
- Offset Address: 0x18

Table 12-2-15 Fields for Register: MSR

Bits	Access	Name	Description & Reset Value
31:8	R	RSVD_MSR_31to8	MSR 31to8 Reserved bits read as 0. Exists: Always
7	R	DCD	Data Carrier Detect. This is used to indicate the current state of the modem control line dcd_n. That is this bit is the complement dcd_n. When the Data Carrier Detect input (dcd_n) is asserted it is an indication that the carrier has been detected by the modem or data set. In Loopback Mode (MCR[4] set to one), DCD is the same as MCR[3] (Out2). Reset Value: 0x0 Values: 0x0 (DEASSERTED): dcd_n input is de-asserted (logic 1) 0x1 (ASSERTED): dcd_n input is asserted (logic 0) Exists: Always
6	R	RI	Ring Indicator. This is used to indicate the current state of the modem control line ri_n. That is this bit is the complement ri_n. When the Ring Indicator input (ri_n) is asserted it is an indication that a telephone ringing signal has been received by the modem or data set. In Loopback Mode (MCR[4] set to one), RI is the same as MCR[2] (Out1). Reset Value: 0x0 Values: 0x0 (DEASSERTED): ri_n input is de-asserted (logic 1) 0x1 (ASSERTED): ri_n input is asserted (logic 0) Exists: Always
5	R	DSR	Data Set Ready. This is used to indicate the current state of the modem control line dsr_n. That is this bit is the complement dsr_n. When the Data Set Ready input (dsr_n) is asserted it is an indication that the modem or data set is ready to establish communications with the DW_apb_uart. In Loopback Mode (MCR[4] set to one), DSR is the same as MCR[0] (DTR). Reset Value: 0x0 Values: 0x0 (DEASSERTED): dsr_n input is de-asserted (logic 1) 0x1 (ASSERTED): dsr_n input is asserted (logic 0) Exists: Always

Bits	Access	Name	Description & Reset Value
4	R	CTS	Clear to Send. This is used to indicate the current state of the modem control line cts_n. That is, this bit is the complement cts_n. When the Clear to Send input (cts_n) is asserted it is an indication that the modem or data set is ready to exchange data with the DW_apb_uart. In Loopback Mode (MCR[4] set to one), CTS is the same as MCR[1] (RTS). Reset Value: 0x0 Values: 0x0 (DEASSERTED): cts_n input is de-asserted (logic 1) 0x1 (ASSERTED): cts_n input is asserted (logic 0) Exists: Always
3	R	DDCD	Delta Data Carrier Detect. This is used to indicate that the modem control line dcd_n has changed since the last time the MSR was read. Reading the MSR clears the DDCD bit. In Loopback Mode (MCR[4] set to one), DDCD reflects changes on MCR[3] (Out2). Note, if the DDCD bit is not set and the dcd_n signal is asserted (low) and a reset occurs (software or otherwise), then the DDCD bit will get set when the reset is removed if the dcd_n signal remains asserted. Reset Value: 0x0 Values: 0x0 (NO_CHANGE): No change on dcd_n since last read of MSR 0x1 (CHANGE): change on dcd_n since last read of MSR Exists: Always
2	R	TERI	Trailing Edge of Ring Indicator. This is used to indicate that a change on the input ri_n (from an active low, to an inactive high state) has occurred since the last time the MSR was read. Reading the MSR clears the TERI bit. In Loopback Mode (MCR[4] set to one), TERI reflects when MCR[2] (Out1) has changed state from a high to a low. Reset Value: 0x0 Values: 0x0 (NO_CHANGE): no change on ri_n since last read of MSR 0x1 (CHANGE): change on ri_n since last read of MSR Exists: Always
1	R	DDSR	Delta Data Set Ready. This is used to indicate that the modem control line dsr_n has changed since the last time the MSR was read. Reading the MSR clears the DDSR bit. In Loopback Mode (MCR[4] set to one), DDSR reflects changes on MCR[0] (DTR). Note, if the DDSR bit is not set and the dsr_n signal is asserted (low) and a reset occurs (software or otherwise), then the DDSR bit will get set when the reset is removed if the dsr_n signal remains asserted. Reset Value: 0x0 Values: 0x0 (NO_CHANGE): no change on dsr_n since last read of MSR 0x1 (CHANGE): change on dsr_n since last read of MSR Exists: Always

Bits	Access	Name	Description & Reset Value
0	R	DCTS	Delta Clear to Send. This is used to indicate that the modem control line cts_n has changed since the last time the MSR was read. Reading the MSR clears the DCTS bit. In Loopback Mode (MCR[4] set to one), DCTS reflects changes on MCR[1] (RTS). Note, if the DCTS bit is not set and the cts_n signal is asserted (low) and a reset occurs (software or otherwise), then the DCTS bit will get set when the reset is removed if the cts_n signal remains asserted. Reset Value: 0x0 Values: 0x0 (NO_CHANGE): no change on cts_n since last read of MSR 0x1 (CHANGE): change on cts_n since last read of MSR Exists: Always

12.2.5.2.12 SCR

- Name: ScratchpadRegister
- Description: Scratchpad Register
- Reset Value: 0x0
- Offset Address: 0x1c

Table 12-2-16 Fields for Register: SCR

Bits	Access	Name	Description & Reset Value
31:8	R	RSVD_SCR_31to8	SCR 31to8 Reserved bits read as 0. Exists: Always
7:0	R/W	SCR	This register is for programmers to use as a temporary storage space. It has no defined purpose in the DW_apb_uart. Reset Value: 0x0 Exists: Always

12.2.5.2.13 LPDLL

- Name: LowPowerDivisorLatchLowRegister.
- Description: Low Power Divisor Latch Low Register. This register is only valid when the UART is configured to have SIR low-power reception capabilities implemented (SIR_LP_RX = Yes). If SIR low-power reception capabilities are not implemented, this register does not exist and reading from this register address returns 0. If UART_16550_COMPATIBLE = No, then this register can be accessed only when the DLAB bit (LCR[7]) is set and the UART is not busy, that is, USR[0] is 0; otherwise this register can be accessed only when the DLAB bit (LCR[7]) is set.
- Reset Value: 0x0.
- Offset Address: 0x20

Table 12-2-17 Fields for Register: LPDLL

Bits	Access	Name	Description & Reset Value
31:8	R	RSVD_LPDLL_31to8	LPDLL 31to8 Reserved bits read as 0. Exists: Always

Bits	Access	Name	Description & Reset Value
7:0	R/W	LPDLL	This register makes up the lower 8-bits of a 16-bit, read/ write, Low Power Divisor Latch register that contains the baud rate divisor for the UART which must give a baud rate of 115.2K. This is required for SIR Low Power (minimum pulse width) detection at the receiver. The output low power baud rate is equal to the serial clock (sclk) frequency divided by sixteen times the value of the baud rate divisor, as follows: Low power baud rate = (serial clock freq) / (16 * divisor) Therefore a divisor must be selected to give a baud rate of 115.2K. Note: When the Low Power Divisor Latch Registers (LPDLL and LPDLH) are set to zero, the low power baud clock is disabled and no low power pulse detection (or any pulse detection for that matter) will occur at the receiver. Also, once the LPDLL is set at least 8 clock cycles of the slowest DW_apb_uart clock should be allowed to pass before transmitting or receiving data. Reset Value: 0x0 Exists: Always

12.2.5.2.14 LPDLH

- Name: LowPowerDivisorLatchHighRegister.
- Description: Low Power Divisor Latch High Register. This register is valid only when the UART is configured to have SIR low-power reception capabilities implemented (SIR_LP_RX = Yes). If SIR low-power reception capabilities are not implemented, this register does not exist and reading from this register address returns 0. If UART_16550_COMPATIBLE = No, then this register can be accessed only when the DLAB bit (LCR[7]) is set and the UART is not busy that is, USR[0] is 0; otherwise this register can be accessed only when the DLAB bit (LCR[7]) is set.
- Reset Value: 0x0.
- Offset Address: 0x24

Table 12-2-18 Fields for Register: LPDLH

Bits	Access	Name	Description & Reset Value
31:8	R	RSVD_LPDLH_31to8	LPDLH 31to8 Reserved and read as 0. Exists: Always

Bits	Access	Name	Description & Reset Value
7:0	R/W	LPDLH	This register makes up the upper 8-bits of a 16-bit, read/ write, Low Power Divisor Latch register that contains the baud rate divisor for the UART which must give a baud rate of 115.2K. This is required for SIR Low Power (minimum pulse width) detection at the receiver. The output low power baud rate is equal to the serial clock (sclk) frequency divided by sixteen times the value of the baud rate divisor, as follows: Low power baud rate = (serial clock freq) / (16 * divisor) Therefore a divisor must be selected to give a baud rate of 115.2K. Note: When the Low Power Divisor Latch Registers (LPDLL and LPDLH) are set to zero, the low power baud clock is disabled and no low power pulse detection (or any pulse detection for that matter) will occur at the receiver. Also, once the LPDLH is set, at least 8 clock cycles of the slowest DW_apb_uart clock should be allowed to pass before transmitting or receiving data. Reset Value: 0x0 Exists: Always

12.2.5.2.15 SRBRn

- Name: ShadowReceiveBufferRegister
- Description: Shadow Receive Buffer Register This register is valid only when the UART is configured to have additional shadow registers implemented (SHADOW = YES). If shadow registers are not implemented, this register does not exist and reading from this register address returns 0.
This register can be accessed only when the DLAB bit (LCR[7]) is cleared.
- Reset Value: 0x0.
- Offset Address: 0x30

Table 12-2-19 Fields for Register: SRBRn

Bits	Access	Name	Description & Reset Value
31:y	R	RSVD_SRBRn	SRBR0 31 to SRBRN_REG_SIZE Reserved bits read read as 0. Exists: Always Range Variable[y]: SRBRN_REG_SIZE

Bits	Access	Name	Description & Reset Value
x:0	R	SRBRn	Shadow Receive Buffer Register n. This is a shadow register for the RBR and has been allocated sixteen 32-bit locations so as to accommodate burst accesses from the master. This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If in non-FIFO mode (FIFO_MODE == NONE) or FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it will be overwritten, resulting in an overrun error. If in FIFO mode (FIFO_MODE != NONE) and FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO will be preserved but any incoming data will be lost. An overrun error will also occur. Note: When UART_9BIT_DATA_EN=0, this field width is 8. When UART_9BIT_DATA_EN=1, this field width is 9. Exists: Always Range Variable[x]: SRBRN_REG_SIZE - 1

12.2.5.2.16 STHRn

- Name: ShadowTransmitHoldingRegister.
- Description: Shadow Transmit Holding Register. This register is valid only when the UART is configured to have additional shadow registers implemented (SHADOW = YES). If shadow registers are not implemented, this register does not exist, and reading from this register address returns 0.
This register can be accessed only when the DLAB bit (LCR[7]) is cleared.
- Reset Value: 0x0.
- Offset Address: 0x30

Table 12-2-20 Fields for Register: STHRn

Bits	Access	Name	Description & Reset Value
31:y	R	RSVD_STHRn	STHR0 31 to STHRN_REG_SIZE Reserved bits read as 0. Exists: Always Range Variable[y]: STHRN_REG_SIZE

Bits	Access	Name	Description & Reset Value
x:0	W	STHRn	Shadow Transmit Holding Register n. This is a shadow register for the THR and has been allocated sixteen 32-bit locations so as to accommodate burst accesses from the master. This register contains data to be transmitted on the serial output port (sout) in UART mode or the serial infrared output (sir_out_n) in infrared mode. Data should only be written to the THR when the THR Empty (THRE) bit (LSR[5]) is set. If in non-FIFO mode or FIFO's are disabled (FCR[0] set to zero) and THRE is set, writing a single character to the THR clears the THRE. Any additional writes to the THR before the THRE is set again causes the THR data to be overwritten. If in FIFO mode and FIFO's are enabled (FCR[0] set to one) and THRE is set, x number of characters of data may be written to the THR before the FIFO is full. The number x (default=16) is determined by the value of FIFO Depth that you set during configuration. Any attempt to write data when the FIFO is full results in the write data being lost. Note: When UART_9BIT_DATA_EN=0, this field width is 8. When UART_9BIT_DATA_EN=1, this field width is 9. The 9th bit is applicable only when LCR_EXT[3]=1. Reset Value: 0x0 Exists: Always Range Variable[x]: STHRN_REG_SIZE - 1

12.2.5.2.17 FAR

- Name: FIFOAccessRegister
- Description: FIFO Access Register
- Reset Value: 0x0.
- Offset Address: 0x70

Table 12-2-21 Fields for Register: FAR

Bits	Access	Name	Description & Reset Value
31:1	R	RSVD_FAR_31to1	FAR 31to1 Reserved bits read as 0. Exists: Always
0	*Varies	FAR	Writes will have no effect when FIFO_ACCESS == No, always readable. This register is use to enable a FIFO access mode for testing, so that the receive FIFO can be written by the master and the transmit FIFO can be read by the master when FIFO's are implemented and enabled. When FIFOs are not implemented or not enabled it allows the RBR to be written by the master and the THR to be read by the master. Note, that when the FIFO access mode is enabled/disabled, the control portion of the receive FIFO and transmit FIFO is reset and the FIFO's are treated as empty. Reset Value: 0x0 Values: 0x0 (DISABLED): FIFO access mode disabled 0x1 (ENABLED): FIFO access mode enabled Exists: Always Memory Access: (FIFO_ACCESS==1) ? read-write : read-only

12.2.5.2.18 TFR

- Name: TransmitFIFORead.
- Description: Transmit FIFO Read. This register is valid only when the UART is configured to have the FIFO access test mode available (FIFO_ACCESS = YES). If not configured, this register does not exist and reading from this register address returns 0.
- Reset Value: 0x0.
- Offset Address: 0x74

Table 12-2-22 Fields for Register: TFR

Bits	Access	Name	Description & Reset Value
31:8	R	RSVD_TFR_31to8	TFR 31to8 Reserved bits read as 0. Exists: Always
7:0	R	TFR	Transmit FIFO Read. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one). When FIFO's are implemented and enabled, reading this register gives the data at the top of the transmit FIFO. Each consecutive read pops the transmit FIFO and gives the next data value that is currently at the top of the FIFO. When FIFO's are not implemented or not enabled, reading this register gives the data in the THR. Reset Value: 0x0 Exists: Always

12.2.5.2.19 RFW

- Name: ReceiveFIFOWrite
- Description: Receive FIFO Write This register is valid only when the UART is configured to have the FIFO access test mode available (FIFO_ACCESS = YES). If not configured, this register does not exist and reading from this register address returns 0.
- Reset Value: 0x0.
- Offset Address: 0x78

Table 12-2-23 Fields for Register: RFW

Bits	Access	Name	Description & Reset Value
31:10	R	RSVD_RFW_31to10	RFW 31to10 Reserved bits read as 0. Exists: Always
9	W	RFFE	Receive FIFO Framing Error. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one). When FIFO's are implemented and enabled, this bit is used to write framing error detection information to the receive FIFO. When FIFO's are not implemented or not enabled, this bit is used to write framing error detection information to the RBR. Reset Value: 0x0 Values: 0x0 (DISABLED): Frame error disabled 0x1 (ENABLED): Frame error enabled Exists: Always

Bits	Access	Name	Description & Reset Value
8	W	RFPE	Receive FIFO Parity Error. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one). When FIFO's are implemented and enabled, this bit is used to write parity error detection information to the receive FIFO. When FIFO's are not implemented or not enabled, this bit is used to write parity error detection information to the RBR. Reset Value: 0x0 Values: 0x0 (DISABLED): Parity error disabled 0x1 (ENABLED): Parity error enabled Exists: Always
7:0	W	RFWD	Receive FIFO Write Data. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one). When FIFO's are implemented and enabled, the data that is written to the RFWD is pushed into the receive FIFO. Each consecutive write pushes the new data to the next write location in the receive FIFO. When FIFO's are not implemented or not enabled, the data that is written to the RFWD is pushed into the RBR. Reset Value: 0x0 Exists: Always

12.2.5.2.20 USR

- Name: UARTStatusregister.
- Description: UART Status register.
- Reset Value: 0x6.
- Offset Address: 0x7c

Table 12-2-24 Fields for Register: USR

Bits	Access	Name	Description & Reset Value
31:5	R	RSVD_USR_31to5	USR 31to5 Reserved bits read as 0. Exists: Always Volatile: true
4	R	RFF	Receive FIFO Full. This bit is only valid when FIFO_STAT == YES. This is used to indicate that the receive FIFO is completely full. That is: This bit is cleared when the RX FIFO is no longer full. Reset Value: 0x0 Values: 0x0 (NOT_FULL): Receive FIFO not full 0x1 (FULL): Receive FIFO full Exists: (FIFO_STAT == 1) && (FIFO_MODE != 0) && (ADDITIONAL_FEATURES == 1) Volatile: true

Bits	Access	Name	Description & Reset Value
3	R	RFNE	Receive FIFO Not Empty. This bit is only valid when FIFO_STAT == YES. This is used to indicate that the receive FIFO contains one or more entries. This bit is cleared when the RX FIFO is empty. Reset Value: 0x0 Values: 0x0 (EMPTY): Receive FIFO is empty 0x1 (NOT_EMPTY): Receive FIFO is not empty Exists: (FIFO_STAT == 1) && (FIFO_MODE != 0) && (ADDITIONAL_FEATURES == 1) Volatile: true
2	R	TFE	Transmit FIFO Empty. This bit is only valid when FIFO_STAT == YES. This is used to indicate that the transmit FIFO is completely empty. This bit is cleared when the TX FIFO is no longer empty. Reset Value: 0x1 Values: 0x0 (NOT_EMPTY): Transmit FIFO is not empty 0x1 (EMPTY): Transmit FIFO is empty Exists: (FIFO_STAT == 1) && (FIFO_MODE != 0) && (ADDITIONAL_FEATURES == 1) Volatile: true
1	R	TFNF	Transmit FIFO Not Full. This bit is only valid when FIFO_STAT == YES. This is used to indicate that the transmit FIFO is not full. This bit is cleared when the TX FIFO is full. Reset Value: 0x1 Values: 0x0 (FULL): Transmit FIFO is full 0x1 (NOT_FULL): Transmit FIFO is not full Exists: (FIFO_STAT == 1) && (FIFO_MODE != 0) && (ADDITIONAL_FEATURES == 1) Volatile: true

Bits	Access	Name	Description & Reset Value
0	R	BUSY	UART Busy. This bit is only valid when UART_16550_COMPATIBLE == NO. This indicates that a serial transfer is in progress, when cleared indicates that the DW_apb_uart is idle or inactive. This bit will be set to 1 (busy) under any of the following conditions: - Transmission in progress on serial interface - Transmit data present in THR, when FIFO access mode is not being used (FAR = 0) and the baud divisor is non-zero ({DLH,DLL} does not equal 0) when the divisor latch access bit is 0 (LCR.DLAB = 0) - Reception in progress on the interface - Receive data present in RBR, when FIFO access mode is not being used (FAR = 0) Note: It is possible for the UART Busy bit to be cleared even though a new character may have been sent from another device. That is, if the DW_apb_uart has no data in the THR and RBR and there is no transmission in progress and a start bit of a new character has just reached the DW_apb_uart. This is due to the fact that a valid start is not seen until the middle of the bit period and this duration is dependent on the baud divisor that has been programmed. If a second system clock has been implemented (CLOCK_MODE == Enabled), the assertion of this bit will also be delayed by several cycles of the slower clock. Reset Value: 0x0 Values: 0x0 (IDLE): DW_apb_uart is idle or inactive 0x1 (BUSY): DW_apb_uart is busy (actively transferring data) Exists: UART_16550_COMPATIBLE == 0 Volatile: true

12.2.5.2.21 TFL

- Name: TransmitFIFOLevel.
- Description: Transmit FIFO Level. TFL register is valid only when the UART is configured to have additional FIFO status registers implemented (FIFO_STAT = YES). If status registers are not implemented, this register does not exist and reading from this register address returns 0.
- Reset Value: 0x0.
- Offset Address: 0x80

Table 12-2-25 Fields for Register: TFL

Bits	Access	Name	Description & Reset Value
31:y	R	RSVD_TFL_31toADDR_WIDTH	TFL 31 to ADDR_WIDTH Reserved bits read as 0. Exists: Always Range Variable[y]: FIFO_ADDR_WIDTH + 1
x:0	R	tfl	Transmit FIFO Level. This indicates the number of data entries in the transmit FIFO. Reset Value: 0x0 Exists: Always Range Variable[x]: FIFO_ADDR_WIDTH

12.2.5.2.22 RFL

- Name: ReceiveFIFOLevel.
- Description: Receive FIFO Level. RFL register is valid only when the UART is configured to have additional FIFO status registers implemented (FIFO_STAT = YES). If status registers are not implemented, this register does not exist and reading from this register address returns 0.
- Reset Value: 0x0.
- Offset Address: 0x84

Table 12-2-26 Fields for Register: RFL

Bits	Access	Name	Description & Reset Value
31:y	R	RSVD_RFL_31toADDR_WIDTH	RFL 31 to ADDR_WIDTH Reserved bits read as 0. Exists: Always Range Variable[y]: FIFO_ADDR_WIDTH + 1
x:0	R	rfl	Receive FIFO Level. This indicates the number of data entries in the receive FIFO. Reset Value: 0x0 Exists: Always Range Variable[x]: FIFO_ADDR_WIDTH

12.2.5.2.23 SRR

- Name: SoftwareResetRegister.
- Description: Software Reset Register. This register is valid only when the UART is configured to have additional shadow registers implemented (SHADOW = YES). If shadow registers are not implemented, this register does not exist and reading from this register address returns 0.
For more information on the amount of time that serial clock modules need in order to see new register values and reset their respective state machines, refer to the 'Clock Support' subsection in the data book.
- Reset Value: 0x0.
- Offset Address: 0x88

Table 12-2-27 Fields for Register: SRR

Bits	Access	Name	Description & Reset Value
31:3	R	RSVD_SRR_31to3	SRR 31to3 Reserved bits read as 0. Exists: Always
2	*Varies	XFR	XMIT FIFO Reset. Writes will have no effect when FIFO_MODE == NONE. This is a shadow register for the XMIT FIFO Reset bit (FCR[2]). This can be used to remove the burden on software having to store previously written FCR values (which are pretty static) just to reset the transmit FIFO. This resets the control portion of the transmit FIFO and treats the FIFO as empty. This will also de-assert the DMA TX request and single signals when additional DMA handshaking signals are selected (DMA_EXTRA = YES). Note that this bit is 'self-clearing'. It is not necessary to clear this bit. Reset Value: 0x0 Exists: Always Memory Access: (FIFO_MODE != 0) ? write-only : read-only

Bits	Access	Name	Description & Reset Value
1	*Varies	RFR	RCVR FIFO Reset. Writes will have no effect when FIFO_MODE == NONE. This is a shadow register for the RCVR FIFO Reset bit (FCR[1]). This can be used to remove the burden on software having to store previously written FCR values (which are pretty static) just to reset the receive FIFO. This resets the control portion of the receive FIFO and treats the FIFO as empty. This will also de-assert the DMA RX request and single signals when additional DMA handshaking signals are selected (DMA_EXTRA == YES). Note that this bit is 'self-clearing' and it is not necessary to clear this bit. Reset Value: 0x0 Exists: Always Memory Access: (FIFO_MODE != 0) ? write-only : read-only
0	W	UR	UART Reset. This asynchronously resets the DW_apb_uart and synchronously removes the reset assertion. For a two clock implementation both pclk and sclk domains will be reset. Reset Value: 0x0 Values: 0x0 (NO_RESET): No Uart Reset 0x1 (RESET): Uart reset Exists: Always

12.2.5.2.24 SRTS

- Name: ShadowRequestToSend.
- Description: Shadow Request to Send. SRTS register is valid only when the UART is configured to have additional shadow registers implemented (SHADOW = YES). If shadow registers are not implemented, this register does not exist and reading from this register address returns 0.
- Reset Value: 0x0.
- Offset Address: 0x8c

Table 12-2-28 Fields for Register: SRTS

Bits	Access	Name	Description & Reset Value
31:1	R	RSVD_SRTS_31to1	SRTS 31to1 Reserved bits read as 0. Exists: Always

Bits	Access	Name	Description & Reset Value
0	R/W	SRTS	Shadow Request to Send. This is a shadow register for the RTS bit (MCR[1]), this can be used to remove the burden of having to performing a read modify write on the MCR. This is used to directly control the Request to Send (rts_n) output. The Request To Send (rts_n) output is used to inform the modem or data set that the UART is ready to exchange data. When Auto RTS Flow Control is not enabled (MCR[5] set to zero), the rts_n signal is set low by programming MCR[1] (RTS) to a high. In Auto Flow Control, AFCE_MODE == Enabled and active (MCR[5] set to one) and FIFO's enable (FCR[0] set to one), the rts_n output is controlled in the same way, but is also gated with the receiver FIFO threshold trigger (rts_n is inactive high when above the threshold) only when RTC Flow Trigger is disabled; otherwise it is gated by the receiver FIFO almost-full trigger, where 'almost full' refers to two available slots in the FIFO (rts_n is inactive high when above the threshold). Note that in Loopback mode (MCR[4] set to one), the rts_n output is held inactive high while the value of this location is internally looped back to an input. Reset Value: 0x0 Values: 0x0 (DEASSERTED): Shadow Request to Send uart_rts_n logic1 0x1 (ASSERTED): Shadow Request to Send uart_rts_n logic0 Exists: Always

12.2.5.2.25 SBCR

- Name: ShadowBreakControlRegister.
- Description: Shadow Break Control Register. SBCR register is valid only when the UART is configured to have additional shadow registers implemented (SHADOW = YES). If shadow registers are not implemented, this register does not exist and reading from this register address returns 0.
- Reset Value: 0x0.
- Offset Address: 0x90

Table 12-2-29 Fields for Register: SBCR

Bits	Access	Name	Description & Reset Value
31:1	R	RSVD_SBCR_31to1	SBCR 31to1 Reserved bits read as 0. Exists: Always

Bits	Access	Name	Description & Reset Value
0	R/W	SBCB	Shadow Break Control Bit. This is a shadow register for the Break bit (LCR[6]), this can be used to remove the burden of having to performing a read modify write on the LCR. This is used to cause a break condition to be transmitted to the receiving device. If set to one the serial output is forced to the spacing (logic 0) state. When not in Loopback Mode, as determined by MCR[4], the sout line is forced low until the Break bit is cleared. If SIR_MODE == Enabled and active (MCR[6] set to one) the sir_out_n line is continuously pulsed. When in Loopback Mode, the break condition is internally looped back to the receiver. Reset Value: 0x0 Values: 0x0 (NO_BREAK): No spacing on serial output 0x1 (BREAK): Serial output forced to the spacing Exists: Always

12.2.5.2.26 SDMAM

- Name: ShadowDMAModeRegister.
- Description: Shadow DMA Mode Register. This register is valid only when the UART is configured to have additional FIFO registers implemented (FIFO_MODE != None) and additional shadow registers implemented (SHADOW = YES). If these registers are not implemented, this register does not exist and reading from this register address returns 0.
- Reset Value: 0x0.
- Offset Address: 0x94

Table 12-2-30 Fields for Register: SDMAM

Bits	Access	Name	Description & Reset Value
31:1	R	RSVD_SDMAM_31to1	SDMAM 31to1 Reserved bits read as 0. Exists: Always
0	R/W	SDMAM	Shadow DMA Mode. This is a shadow register for the DMA mode bit (FCR[3]). This can be used to remove the burden of having to store the previously written value to the FCR in memory and having to mask this value so that only the DMA Mode bit gets updated. This determines the DMA signalling mode used for the dma_tx_req_n and dma_rx_req_n output signals when additional DMA handshaking signals are not selected (DMA_EXTRA == NO). See section 5.9 on page 54 for details on DMA support. Reset Value: 0x0 Values: 0x0 (MODE_0): Mode 0 0x1 (MODE_1): Mode 1 Exists: Always

12.2.5.2.27 SFE

- Name: ShadowFIFOEnableRegister.
- Description: Shadow FIFO Enable Register. SFE register is valid only when the UART is configured to have additional FIFO registers implemented (FIFO_MODE != None) and additional shadow registers implemented

(SHADOW = YES). If these registers are not implemented, this register does not exist and reading from this register address returns 0.

- Reset Value: 0x0.
- Offset Address: 0x98

Table 12-2-31 Fields for Register: SFE

Bits	Access	Name	Description & Reset Value
31:1	R	RSVD_SFE_31to1	SFE 31to1 Reserved bits read as 0. Exists: Always
0	R/W	SFE	Shadow FIFO Enable. This is a shadow register for the FIFO enable bit (FCR[0]). This can be used to remove the burden of having to store the previously written value to the FCR in memory and having to mask this value so that only the FIFO enable bit gets updated. This enables/disables the transmit (XMIT) and receive (RCVR) FIFO's. If this bit is set to zero (disabled) after being enabled then both the XMIT and RCVR controller portion of FIFO's will be reset. Reset Value: 0x0 Values: 0x0 (DISABLED): FIFOs are disabled 0x1 (ENABLED): FIFOs are enabled Exists: Always

12.2.5.2.28 SRT

- Name: ShadowRCVRTriggerRegister.
- Description: Shadow RCVR Trigger Register. SRT register is valid only when the UART is configured to have additional FIFO registers implemented (FIFO_MODE != None) and additional shadow registers implemented (SHADOW = YES). If these registers are not implemented, this register does not exist and reading from this register address returns 0.
- Reset Value: 0x0.
- Offset Address: 0x9c

Table 12-2-32 Fields for Register: SRT

Bits	Access	Name	Description & Reset Value
31:2	R	RSVD_SRT_31to2	SRT 31to2 Reserved bits read as 0. Exists: Always

Bits	Access	Name	Description & Reset Value
1:0	R/W	SRT	Shadow RCVR Trigger. This is a shadow register for the RCVR trigger bits (FCR[7:6]). This can be used to remove the burden of having to store the previously written value to the FCR in memory and having to mask this value so that only the RCVR trigger bit gets updated. This is used to select the trigger level in the receiver FIFO at which the Received Data Available Interrupt will be generated. It also determines when the dma_rx_req_n signal will be asserted when DMA Mode (FCR[3]) is set to one. Reset Value: 0x0 Values: 0x0 (FIFO_CHAR_1): 1 character in FIFO 0x1 (FIFO_QUARTER_FULL): FIFO 1/4 full 0x2 (FIFO_HALF_FULL): FIFO 1/2 full 0x3 (FIFO_FULL_2): FIFO 2 less than full Exists: Always

12.2.5.2.29 STET

- Name: ShadowTXEmptyTriggerRegister.
- Description: Shadow TX Empty Trigger Register. This register is valid only when the UART is configured to have FIFOs implemented (FIFO_MODE != NONE) and THRE interrupt support implemented (THRE_MODE_USER = Enabled) and additional shadow registers implemented (SHADOW = YES). If FIFOs are not implemented or THRE interrupt support is not implemented or shadow registers are not implemented, this register does not exist and reading from this register address returns 0.
- Reset Value: 0x0.
- Offset Address: 0xa0

Table 12-2-33 Fields for Register: STET

Bits	Access	Name	Description & Reset Value
31:2	R	RSVD_STET_31to2	STET 31to2 Reserved bits read as 0. Exists: Always
1:0	R/W	STET	Shadow TX Empty Trigger. This is a shadow register for the TX empty trigger bits (FCR[5:4]). This can be used to remove the burden of having to store the previously written value to the FCR in memory and having to mask this value so that only the TX empty trigger bit gets updated. Writes will have no effect when THRE_MODE_USER == Disabled. This is used to select the empty threshold level at which the THRE Interrupts will be generated when the mode is active. Reset Value: 0x0 Values: 0x0 (FIFO_EMPTY): FIFO empty 0x1 (FIFO_CHAR_2): 2 characters in FIFO 0x2 (FIFO_QUARTER_FULL): FIFO 1/4 full 0x3 (FIFO_HALF_FULL): FIFO 1/2 full Exists: Always

12.2.5.2.30 HTX

- Name: HaltTX.

- Description: Halt TX.
- Reset Value: 0x0.
- Offset Address: 0xa4

Table 12-2-34 Fields for Register: HTX

Bits	Access	Name	Description & Reset Value
31:1	R	RSVD_HTX_31to1	HTX 31to1 Reserved bits read as 0. Exists: Always
0	*Varies	HTX	Halt TX. Writes will have no effect when FIFO_MODE == NONE, always readable. This register is use to halt transmissions for testing, so that the transmit FIFO can be filled by the master when FIFO's are implemented and enabled. Note, if FIFO's are implemented and not enabled the setting of the halt TX register will have no effect on operation. Reset Value: 0x0 Values: 0x0 (DISABLED): Halt Transmission disabled 0x1 (ENABLED): Halt Transmission enabled Exists: Always Memory Access: (FIFO_MODE==0) ? read-only : read-write

12.2.5.2.31 DMASA

- Name: DMASoftwareAcknowledgeRegister.
- Description: DMA Software Acknowledge Register.
- Reset Value: 0x0.
- Offset Address: 0xa8

Table 12-2-35 Fields for Register: DMASA

Bits	Access	Name	Description & Reset Value
31:1	R	RSVD_DMASA_31to1	DMASA 31to1 Reserved bits read as 0. Exists: Always
0	*Varies	DMASA	DMA Software Acknowledge. Writes will have no effect when DMA_EXTRA == No. This register is use to perform DMA software acknowledge if a transfer needs to be terminated due to an error condition. For example, if the DMA disables the channel, then the DW_apb_uart should clear its request. This will cause the TX request, TX single, RX request and RX single signals to de- assert. Note that this bit is 'self-clearing' and it is not necessary to clear this bit. Reset Value: 0x0 Values: 0x1 (SOFT_ACK): DMA software acknowledge Exists: Always Memory Access: (DMA_EXTRA==1) ? write-only : read-only

12.2.5.2.32 TCR

- Name: TransceiverControlRegister.

- Description: Transceiver Control Register. This register is used to enable or disable RS485 mode and also control the polarity values for Driven enable (de) and Receiver Enable (re) signals.
This register is only valid when the UART is configured to have RS485 interface implemented (UART_RS485_INTERFACE_EN = ENABLED). If RS485 interface is not implemented, this register does not exist and reading from this register address returns zero.
- Reset Value: 0x6.
- Offset Address: 0xac

Table 12-2-36 Fields for Register: TCR

Bits	Access	Name	Description & Reset Value
31:5	R	RSVD_TCR_31to5	TCR 31to5 Reserved bits read as 0. Exists: Always
4:3	*Varies	XFER_MODE	Transfer Mode. 0: In this mode, transmit and receive can happen simultaneously. The user can enable DE_EN, RE_EN at any point of time. Turn around timing as programmed in the TAT register is not applicable in this mode. 1: In this mode, DE and RE are mutually exclusive. Either DE or RE only one of them is expected to be enabled through programming. Hardware will consider the Turn Around timings which are programmed in the TAT register while switching from RE to DE or DE to RE. For transmission Hardware will wait if it is in middle of receiving any transfer, before it starts transmitting. 2: In this mode, DE and RE are mutually exclusive. Once DE_EN/RE_EN is programed - by default 're' will be enabled and DW_apb_uart controller will be ready to receive. If the user programs the TX FIFO with the data then DW_apb_uart, after ensuring no receive is in progress, disable 're' and enable 'de' signal. Once the TX FIFO becomes empty, 're' signal gets enabled and 'de' signal will be disabled. In this mode of operation hardware will consider the Turn Around timings which are programmed in the TAT register while switching from RE to DE or DE to RE. In this mode, 'de' and 're' signals are strictly complementary to each other. Reset Value: 0x0 Exists: Always Memory Access: (UART_RS485_INTERFACE_EN==1) ? read-write : read-only
2	*Varies	DE_POL	Driver Enable Polarity. 1: DE signal is active high 0: DE signal is active low Reset Value: UART_DE_POL Exists: Always Memory Access: (UART_RS485_INTERFACE_EN==1) ? read-write : read-only

Bits	Access	Name	Description & Reset Value
1	*Varies	RE_POL	Receiver Enable Polarity. 1: RE signal is active high 0: RE signal is active low Reset Value: UART_RE_POL Exists: Always Memory Access: (UART_RS485_INTERFACE_EN==1) ? read-write : read-only
0	*Varies	RS485_EN	RS485 Transfer Enable. 0 : In this mode, the transfers are still in the RS232 mode. All other fields in this register are reserved and register DE_EN/RE_EN/TAT are also reserved. 1 : In this mode, the transfers will happen in RS485 mode. All other fields of this register are applicable. Reset Value: 0x0 Exists: Always Memory Access: (UART_RS485_INTERFACE_EN==1) ? read-write : read-only

12.2.5.2.33 DE_EN

- Name: DriverOutputEnableRegister.
- Description: Driver Output Enable Register. The Driver Output Enable Register (DE_EN) is used to control the assertion and de-assertion of the DE signal.
This register is only valid when the UART is configured to have RS485 interface implemented (UART_RS485_INTERFACE_EN = ENABLED). If RS485 interface is not implemented, this register does not exist and reading from this register address will return zero.
- Reset Value: 0x0.
- Offset Address: 0xb0

Table 12-2-37 Fields for Register: DE_EN

Bits	Access	Name	Description & Reset Value
31:1	R	RSVD_DE_EN_31to1	DE_EN 31to1 Reserved bits read as 0. Exists: Always
0	*Varies	DE_Enable	DE Enable control. The 'DE Enable' register bit is used to control assertion and de-assertion of 'de' signal. - 0: De-assert 'de' signal - 1: Assert 'de' signal Reset Value: 0x0 Exists: Always Memory Access: (UART_RS485_INTERFACE_EN==1) ? read-write : read-only

12.2.5.2.34 RE_EN

- Name: ReceiverOutputEnableRegister.
- Description: Receiver Output Enable Register. The Receiver Output Enable Register (RE_EN) is used to control the assertion and de-assertion of the RE signal.
This register is only valid when the UART is configured to have RS485 interface implemented

(UART_RS485_INTERFACE_EN = ENABLED). If the RS485 interface is not implemented, this register does not exist and reading from this register address will return zero.

- Reset Value: 0x0.
- Offset Address: 0xb4

Table 12-2-38 Fields for Register: RE_EN

Bits	Access	Name	Description & Reset Value
31:1	R	RSVD_RE_EN_31to1	RE_EN 31to1 Reserved bits read as 0. Exists: Always
0	*Varies	RE_Enable	RE Enable control. The 'RE Enable' register bit is used to control assertion and de-assertion of 're' signal. 0: De-assert 're' signal 1: Assert 're' signal Reset Value: 0x0 Exists: Always Memory Access: (UART_RS485_INTERFACE_EN==1) ? read-write : read-only

12.2.5.2.35 DET

- Name: DriverOutputEnableTimingRegister.
- Description: Driver Output Enable Timing Register. The Driver Output Enable Timing Register (DET) is used to control the DE assertion and de-assertion timings of 'de' signal.
This register is only valid when the UART is configured to have RS485 interface implemented (UART_RS485_INTERFACE = ENABLED). If RS485 interface is not implemented, this register does not exist and reading from this register address will return zero.
- Reset Value: 0x0.
- Offset Address: 0xb8

Table 12-2-39 Fields for Register: DET

Bits	Access	Name	Description & Reset Value
31:24	R	RSVD_DE_DEAT_31to24	DET 31to24 Reserved bits read as 0. Exists: Always
23:16	*Varies	DE_De-assertion_Time	Driver Enable de-assertion time. This field controls the amount of time (in terms of number of serial clock periods) between the end of stop bit on the sout to the falling edge of Driver output enable signal. Reset Value: 0x0 Exists: Always Memory Access: (UART_RS485_INTERFACE_EN==1) ? read-write : read-only
15:8	R	RSVD_DE_AT_15to8	DET 15to8 Reserved bits read as 0. Exists: Always

Bits	Access	Name	Description & Reset Value
7:0	*Varies	DE_Assertion_Time	Driver Enable assertion time. This field controls the amount of time (in terms of number of serial clock periods) between the assertion of rising edge of Driver output enable signal to serial transmit enable. Any data in transmit buffer, will start on serial output (sout) after the transmit enable. Reset Value: 0x0 Exists: Always Memory Access: (UART_RS485_INTERFACE_EN==1) ? read-write : read-only

12.2.5.2.36 TAT

- Name: TurnAroundTimingRegister.
- Description: TurnAround Timing Register. The TurnAround Timing Register (TAT) is used to hold the turnaround time between switching of 're' and 'de' signals.
This register is only valid when the UART is configured to have the RS485 interface implemented (UART_RS485_INTERFACE_EN = ENABLED). If RS485 interface is not implemented, this register does not exist and reading from this register address will return zero.
- Reset Value: 0x0.
- Offset Address: 0xbc

Table 12-2-40 Fields for Register: TAT

Bits	Access	Name	Description & Reset Value
31:16	*Varies	RE_to_DE	Receiver Enable to Driver Enable TurnAround time. Turnaround time (in terms of serial clock) for RE De- assertion to DE assertion. Note: - If the DE assertion time in the DET register is 0, then the actual value is the programmed value + 3. - If the DE assertion time in the DET register is 1, then the actual value is the programmed value + 2. - If the DE assertion time in the DET register is greater than 1, then the actual value is the programmed value + 1. Reset Value: 0x0 Exists: Always Memory Access: (UART_RS485_INTERFACE_EN==1) ? read-write : read-only
15:0	*Varies	DE_to_RE	Driver Enable to Receiver Enable TurnAround time. Turnaround time (in terms of serial clock) for DE De- assertion to RE assertion. Note: The actual time is the programmed value + 1. Reset Value: 0x0 Exists: Always Memory Access: (UART_RS485_INTERFACE_EN==1) ? read-write : read-only

12.2.5.2.37 DLF

- Name: DivisorLatchFractionRegister.
- Description: Divisor Latch Fraction Register. This register is only valid when the UART is configured to have Fractional Baud rate Divisor implemented (FRACTIONAL_BAUD_DIVISOR_EN = ENABLED). If Fractional Baud rate divisor is not implemented, this register does not exist and reading from this register address will return zero.

- Reset Value: 0x0.
- Offset Address: 0xc0

Table 12-2-41 Fields for Register: DLF

Bits	Access	Name	Description & Reset Value
31:y	R	RSVD_DLF	DLF 31 to DLF_SIZE Reserved bits read as 0. Exists: Always Range Variable[y]: DLF_SIZE
x:0	*Varies	DLF	Fractional part of divisor. The fractional value is added to integer value set by DLH, DLL. Fractional value is determined by (Divisor Fraction value)/(2^DLF_SIZE). For information on DLF values to be programmed for DLF_SIZE=4, see the 'Fractional Baud Rate Support' section in the DW_apb_uart Databook. Reset Value: 0x0 Exists: Always Range Variable[x]: DLF_SIZE - 1 Memory Access: (FRACTIONAL_BAUD_DIVISOR_EN==1) ? read-write : read-only

12.2.5.2.38 RAR

- Name: ReceiveAddressRegister.
- Description: Receive Address Register.
- Reset Value: 0x0.
- Offset Address: 0xc4

Table 12-2-42 Fields for Register: RAR

Bits	Access	Name	Description & Reset Value
31:8	R	RSVD_RAR_31to8	RAR 31to8 Reserved bits read as 0. Exists: Always
7:0	*Varies	RAR	This is an address matching register during receive mode. If the 9-th bit is set in the incoming character then the remaining 8-bits will be checked against this register value. If the match happens then sub-sequent characters with 9-th bit set to 0 will be treated as data byte until the next address byte is received. Note: - This register is applicable only when 'ADDR_MATCH'(LCR_EXT[1] and 'DLS_E' (LCR_EXT[0]) bits are set to 1. - If UART_16550_COMPATIBLE is configured to 0, then RAR should be programmed only when UART is not busy. - If UART_16550_COMPATIBLE is configured to 0, then RAR can be programmed at any point of the time. However, user must not change this register value when any receive is in progress. Reset Value: 0x0 Exists: Always Memory Access: (UART_9BIT_DATA_EN==1) ? read- write : read-only

12.2.5.2.39 TAR

- Name: TransmitAddressRegister.

- Description: Transmit Address Register.
- Reset Value: 0x0.
- Offset Address: 0xc8

Table 12-2-43 Fields for Register: TAR

Bits	Access	Name	Description & Reset Value
31:8	R	RSVD_TAR_31to8	TAR 31to8 Reserved bits read as 0. Exists: Always
7:0	*Varies	TAR	This is an address matching register during transmit mode. If DLS_E (LCR_EXT[0]) bit is enabled, then DW_apb_uart will send the 9-bit character with 9-th bit set to 1 and remaining 8-bit address will be sent from this register provided 'SEND_ADDR' (LCR_EXT[2]) bit is set to 1. Note: - This register is used only to send the address. The normal data should be sent by programming THR register. - Once the address is started to send on the DW_apb_uart serial lane, then 'SEND_ADDR' bit will be auto-cleared by the hardware. Reset Value: 0x0 Exists: Always Memory Access: (UART_9BIT_DATA_EN==1) ? read- write : read-only

12.2.5.2.40 LCR_EXT

- Name: LineExtendedControlRegister.
- Description: Line Extended Control Register.
- Reset Value: 0x0.
- Offset Address: 0xcc

Table 12-2-44 Fields for Register: LCR_EXT

Bits	Access	Name	Description & Reset Value
31:4	R	RSVD_LCR_EXT	LCR_EXT 31to4 Reserved bits read as 0. Exists: Always Volatile: true
3	R/W	TRANSMIT_MODE	Transmit mode control bit. This bit is used to control the type of transmit mode during 9-bit data transfers. 1: In this mode of operation, Transmit Holding Register (THR) and Shadow Transmit Holding Register (STHR) are 9-bit wide. The user needs to ensure that the THR/ STHR register is written correctly for address/data. Address: 9th bit is set to 1, Data : 9th bit is set to 0. Note: Transmit address register (TAR) is not applicable in this mode of operation. 0: In this mode of operation, Transmit Holding Register (THR) and Shadow Transmit Holding register (STHR) are 8-bit wide. The user needs to program the address into Transmit Address Register (TAR) and data into the THR/ STHR register. SEND_ADDR bit is used as a control knob to indicate the DW_apb_uart on when to send the address. Reset Value: 0x0 Exists: Always Volatile: true

Bits	Access	Name	Description & Reset Value
2	R/W	SEND_ADDR	Send address control bit. This bit is used as a control knob for the user to determine when to send the address during transmit mode. 1 = 9-bit character will be transmitted with 9-th bit set to 1 and the remaining 8-bits will match to what is being programmed in 'Transmit Address Register'. 0 = 9-bit character will be transmitted with 9-th bit set to 0 and the remaining 8-bits will be taken from the TXFIFO which is programmed through 8-bit wide THR/STHR register. Note: 1. This bit is auto-cleared by the hardware, after sending out the address character. User is not expected to program this bit to 0. 2. This field is applicable only when DLS_E bit is set to 1 and TRANSMIT_MODE is set to 0. Reset Value: 0x0 Exists: UART_9BIT_DATA_EN == 1 Volatile: true
1	R/W	ADDR_MATCH	Address Match Mode. This bit is used to enable the address match feature during receive. 1 = Address match mode; DW_apb_uart will wait until the incoming character with 9-th bit set to 1. And further checks to see if the address matches with what is programmed in 'Receive Address Match Register'. If match is found, then sub-sequent characters will be treated as valid data and DW_apb_uart starts receiving data. 0 = Normal mode; DW_apb_uart will start to receive the data and 9-bit character will be formed and written into the receive RXFIFO. User is responsible to read the data and differentiate b/n address and data. Note: This field is applicable only when DLS_E is set to 1. Exists: Always Volatile: true
0	R/W	DLS_E	Extension for DLS. This bit is used to enable 9-bit data for transmit and receive transfers. Reset Value: 0x0 Exists: Always Volatile: true

12.2.5.2.41 CPR

- Name: ComponentParameterRegister.
- Description: Component Parameter Register. This register is valid only when UART_ADD_ENCODED_PARAMS = 1. If the UART_ADD_ENCODED_PARAMS parameter is not set, this register does not exist and reading from this register address returns 0.
- Reset Value: Configuration-dependent.
- Offset Address: 0xf4

Table 12-2-45 Fields for Register: CPR

Bits	Access	Name	Description & Reset Value
31:24	R	RSVD_CPR_31to24	CPR 31to24 Reserved bits read as 0. Exists: Always

Bits	Access	Name	Description & Reset Value
23:16	R	FIFO_MODE	Encoding of FIFO_MODE configuration parameter value. Values: 0x0 (FIFO_MODE_0): FIFO mode is 0 0x1 (FIFO_MODE_16): FIFO mode is 16 0x2 (FIFO_MODE_32): FIFO mode is 32 0x4 (FIFO_MODE_64): FIFO mode is 64 0x8 (FIFO_MODE_128): FIFO mode is 128 0x10 (FIFO_MODE_256): FIFO mode is 256 0x20 (FIFO_MODE_512): FIFO mode is 512 0x40 (FIFO_MODE_1024): FIFO mode is 1024 0x80 (FIFO_MODE_2048): FIFO mode is 2048 Exists: Always
15:14	R	RSVD_CPR_15to14	CPR 15to14 Reserved bits read as 0. Exists: Always
13	R	DMA_EXTRA	Encoding of DMA_EXTRA configuration parameter value. Values: 0x0 (DISABLED): DMA_EXTRA disabled 0x1 (ENABLED): DMA_EXTRA enabled Exists: Always
12	R	UART_ADD_ENCODED_PARAMS	Encoding of UART_ADD_ENCODED_PARAMS configuration parameter value. Values: 0x0 (DISABLED): UART_ADD_ENCODED_PARAMS disabled 0x1 (ENABLED): UART_ADD_ENCODED_PARAMS enabled Exists: Always
11	R	SHADOW	Encoding of SHADOW configuration parameter value. Values: 0x0 (DISABLED): SHADOW disabled 0x1 (ENABLED): SHADOW enabled Exists: Always
10	R	FIFO_STAT	Encoding of FIFO_STAT configuration parameter value. Values: 0x0 (DISABLED): FIFO_STAT disabled 0x1 (ENABLED): FIFO_STAT enabled Exists: Always
9	R	FIFO_ACCESS	Encoding of FIFO_ACCESS configuration parameter value. Values: 0x0 (DISABLED): FIFO_ACCESS disabled 0x1 (ENABLED): FIFO ACCESS enabled Exists: Always

Bits	Access	Name	Description & Reset Value
8	R	ADDITIONAL_FEAT	Encoding of ADDITIONAL_FEATURES configuration parameter value. Values: 0x0 (DISABLED): Additional features disabled 0x1 (ENABLED): Additional features enabled Exists: Always
7	R	SIR_LP_MODE	Encoding of SIR_LP_MODE configuration parameter value. Values: 0x0 (DISABLED): SIR_LP mode disabled 0x1 (ENABLED): SIR_LP mode enabled Exists: Always
6	R	SIR_MODE	Encoding of SIR_MODE configuration parameter value. Values: 0x0 (DISABLED): SIR mode disabled 0x1 (ENABLED): SIR mode enabled Exists: Always
5	R	THRE_MODE	Encoding of THRE_MODE configuration parameter value. Values: 0x0 (DISABLED): THRE mode disabled 0x1 (ENABLED): THRE mode enabled Exists: Always
4	R	AFCE_MODE	Encoding of AFCE_MODE configuration parameter value. Values: 0x0 (DISABLED): AFCE mode disabled 0x1 (ENABLED): AFCE mode enabled Exists: Always
3:2	R	RSVD_CPR_3to2	CPR 3to2 Reserved bits read as 0. Exists: Always
1:0	R	APB_DATA_WIDTH	Encoding of APB_DATA_WIDTH configuration parameter value. Values: 0x0 (APB_8BITS): APB data width is 8 bits 0x1 (APB_16BITS): APB data width is 16 bits 0x2 (APB_32BITS): APB data width is 32 bits Exists: Always

12.2.5.2.42 UCV

- Name: UARTComponentVersion.
- Description: UART Component Version. This register is valid only when the UART is configured to have additional features implemented (ADDITIONAL_FEATURES = YES). If additional features are not implemented, this register does not exist and reading from this register address returns 0.
- Reset Value: SeethereleasestableintheAMBA2releasenotes.
- Offset Address: 0xf8

Table 12-2-46 Fields for Register: UCV

Bits	Access	Name	Description & Reset Value
31:0	R	UART_Component_Version	ASCII value for each number in the version, followed by . <i>For example 32_30_31_2A represents the version 2.01</i> Exists: Always

12.2.5.2.43 CTR

- Name: ComponentTypeRegister.
- Description: Component Type Register. This register is valid only when the UART is configured to have additional features implemented (ADDITIONAL_FEATURES = YES). If additional features are not implemented, this register does not exist and reading from this register address returns 0.
- Reset Value: 0x44570110.
- Offset Address: 0xfc

Table 12-2-47 Fields for Register: CTR

Bits	Access	Name	Description & Reset Value
31:0	R	Peripheral_ID	This register contains the peripherals identification code. Reset Value: 0x44570110 Exists: Always

12.2.6 Interface Description

Table 12-2-48 UART Interface Description

Module pin	Direction	PAD name	IOMUX
uart0_sout	O	PAD_IO_38	IO_38_sel == 3'd1;
uart0_sin	I	PAD_IO_39	IO_39_sel == 3'd1;
uart1_sout	O	PAD_IO_40;PAD_IO_9;PAD_IO_3	IO_40_sel == 3'd1;IO_9_sel == 3'd2;IO_3_sel == 3'd3
uart1_sin	I	PAD_IO_41;PAD_IO_10;PAD_IO_4	IO_41_sel == 3'd1;IO_10_sel == 3'd2;IO_4_sel == 3'd3
uart1_rts_n	O	PAD_IO_42	IO_42_sel == 3'd1;
uart1_cts_n	I	PAD_IO_43	IO_43_sel == 3'd1;
uart2_sout	O	PAD_IO_44;PAD_IO_11;PAD_IO_5	IO_44_sel == 3'd1;IO_11_sel == 3'd2;IO_5_sel == 3'd3
uart2_sin	I	PAD_IO_45;PAD_IO_12;PAD_IO_6	IO_45_sel == 3'd1;IO_12_sel == 3'd2;IO_6_sel == 3'd3
uart2_rts_n	O	PAD_IO_46	IO_46_sel == 3'd1;
uart2_cts_n	I	PAD_IO_47	IO_47_sel == 3'd1;
uart3_sout	O	PAD_IO_50;PAD_IO_28;PAD_IO_32	IO_50_sel == 3'd1;IO_28_sel == 3'd2;IO_32_sel == 3'd3
uart3_sin	I	PAD_IO_51;PAD_IO_29;PAD_IO_33	IO_51_sel == 3'd1;IO_29_sel == 3'd2;IO_33_sel == 3'd3

Module pin	Direction	PAD name	IOMUX
uart3_rts_n	O	PAD_IO_52;PAD_IO_30;PAD_IO_34	IO_52_sel == 3'd1;IO_30_sel == 3'd2;IO_34_sel == 3'd3
uart3_cts_n	I	PAD_IO_53;PAD_IO_31;PAD_IO_35	IO_53_sel == 3'd1;IO_31_sel == 3'd2;IO_35_sel == 3'd3
uart3_de	O	PAD_IO_62	IO_62_sel == 3'd2
uart3_re	O	PAD_IO_63	IO_63_sel == 3'd2
uart4_sout	O	PAD_IO_48; PAD_IO_36	IO_48_sel == 3'd1; IO_36_sel == 3'd3
uart4_sin	I	PAD_IO_49; PAD_IO_37	IO_49_sel == 3'd1; IO_37_sel == 3'd3

12.2.7 Application Notes

12.2.7.1 Programming Flow for Fractional Baud Rate

The flow diagram in Figure 12-2-23 shows the programming sequence for setting up the UART for transmission.

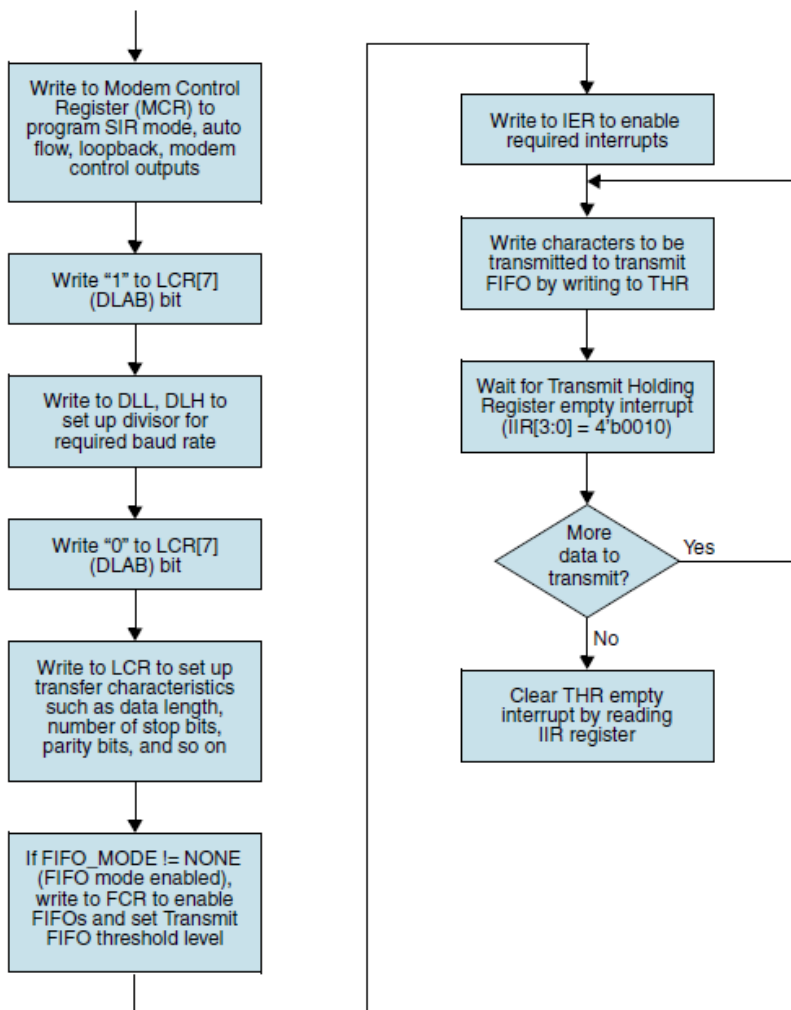


Figure 12-2-23 Flowchart for UART Transmit Programming Example

The flow diagram in Figure 12-2-24 shows the programming sequence for setting up the UART for reception.

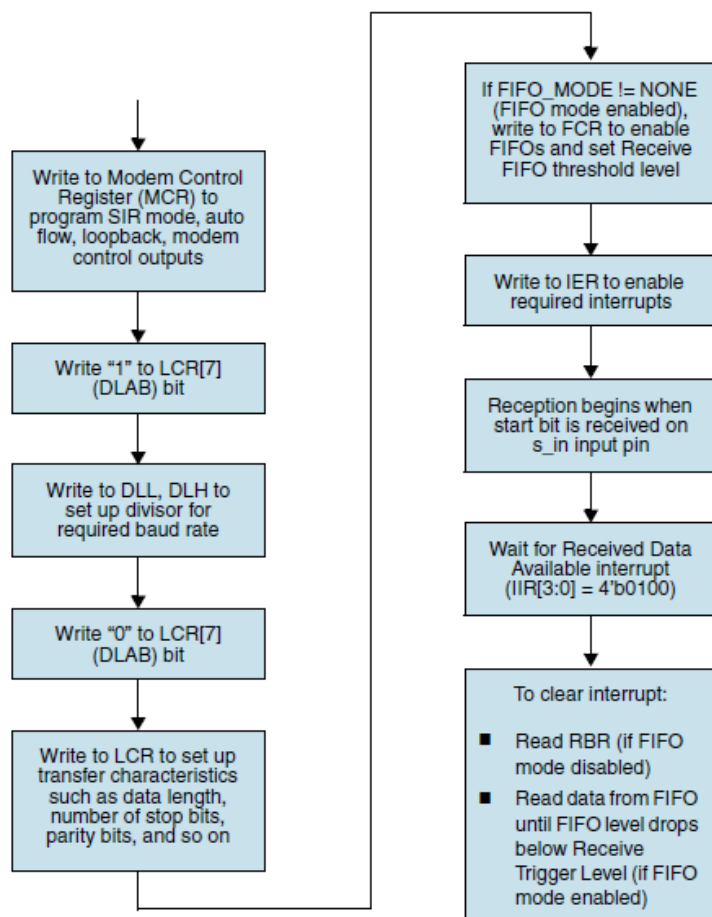


Figure 12-2-24 Flowchart for UART Receive Programming Example

12.2.7.2 Programming Flow in RS485 Mode

1. Full Duplex Mode (XFER_MODE=0)

- Program the TCR register to set the XFER_MODE(0), DE_POL (polarity of 'de'signal) and RE_POL (polarity of 're' signals).
- Program the DE assertion and de-assertion timing in the DET register.
- Program RE_EN and DE_EN register to assert 're' signal and 'de' signal, respectively.
- Perform the data transmission and reception.
- Program RE_EN to de-assert 're' signal.
- 'The de' signal gets de-asserted based on TxFIFO empty. Program DE_EN to '0' if you do not want to transmit further.

2. Software-Enabled Half Duplex Mode (XFER_MODE=1)

Program the TCR register to set the XFER_MODE (1), DE_POL (polarity of 'de' signal) and RE_POL (polarity of 're' signals).

- Program the DE assertion and de-assertion timing in the DET register.
- Program the turnaround times in TAT register.
- Program RE_EN and DE_EN registers to assert 're' signal and 'de' signal, respectively.
- Perform the data transmission/receive.
- Program the RE_EN register to de-assert 're' signal.
- Program the DE_EN register to '0', before programming RE_EN to '1'.
 - 'de' signal de-assertion is based on TxFIFO empty condition and will be taken care by the Hardware. You need to program DE_EN to '0' only in the situation where you want to change the mode.

3. Hardware enabled Half Duplex mode (XFER_MODE=2)

- Program the TCR register to set the XFER_MODE (2), DE_POL (polarity of 'de' signal) and RE_POL (polarity of 're' signals).
- Program the DE assertion and de-assertion timing in the DET register.
- Program the turnaround times in TAT register.
- Program RE_EN and DE_EN register to enable the transmit and receive paths.
- Perform the Data transmission / receive.
- Once the 'RE_EN' and 'DE_EN' is programmed to '1', then by default 're' signal will be asserted and 'de' will be de-asserted. When the software pushes the data into the TX FIFO and if there is no ongoing receive transfer, then the 're' signal will get de-asserted and then the 'de' signal gets asserted until the TX FIFO has data to be transmitted.
- RE_EN and DE_EN will still serve as the software overrides to decide when to shutdown transmit and receive paths.

12.2.7.3 Programming Flow in 9-bit Data Mode

1. Transmit Mode 0

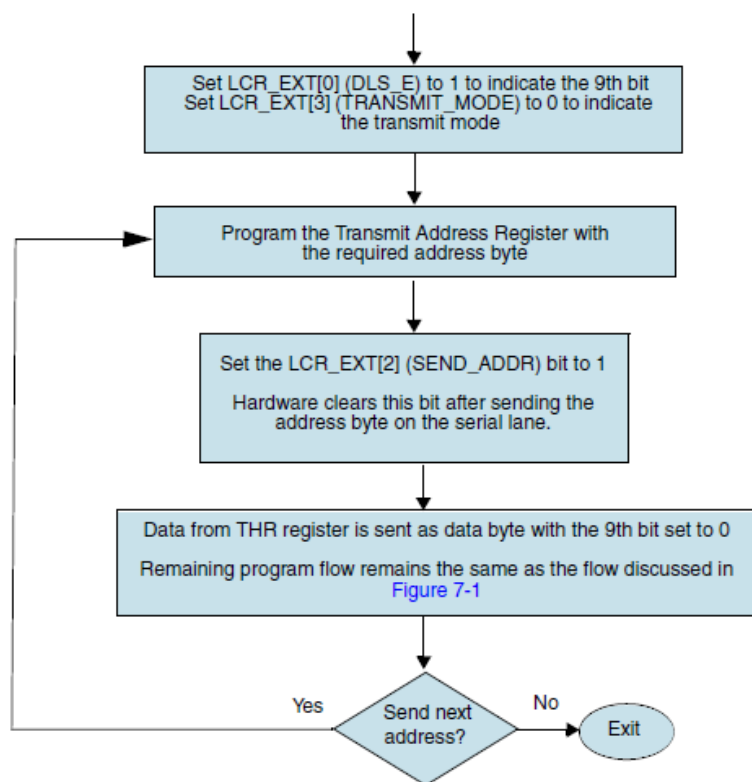


Figure 12-2-25 Auto Address Transmit Mode

2. Transmit Mode 1

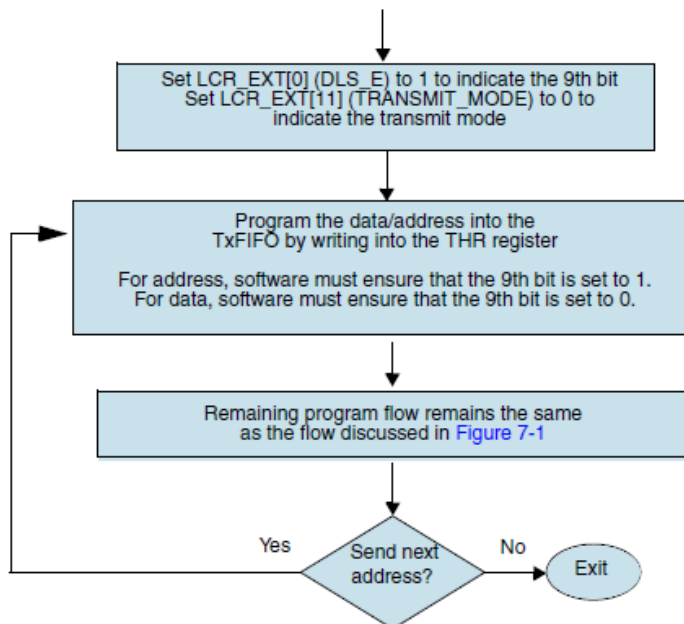


Figure 12-2-26 Normal Transmit Mode Programming Flow

3. Hardware Address Match Receive mode For setting up UART in hardware address match receive mode, follow the programming sequence mentioned in Figure 12-2-24 and additional steps as mentioned in Figure 12-2-27.

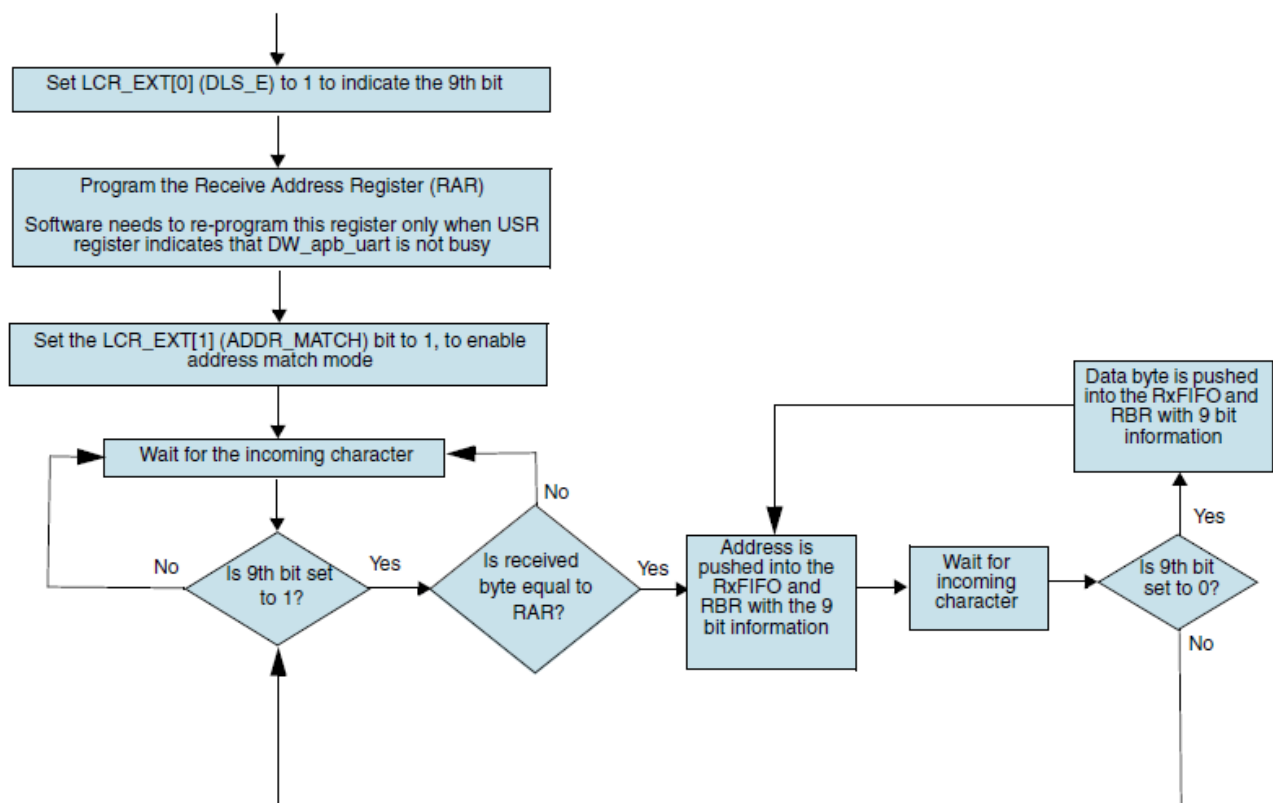


Figure 12-2-27 Hardware Address Match Receive Programming Flow

4. Software Address Match Receive mode

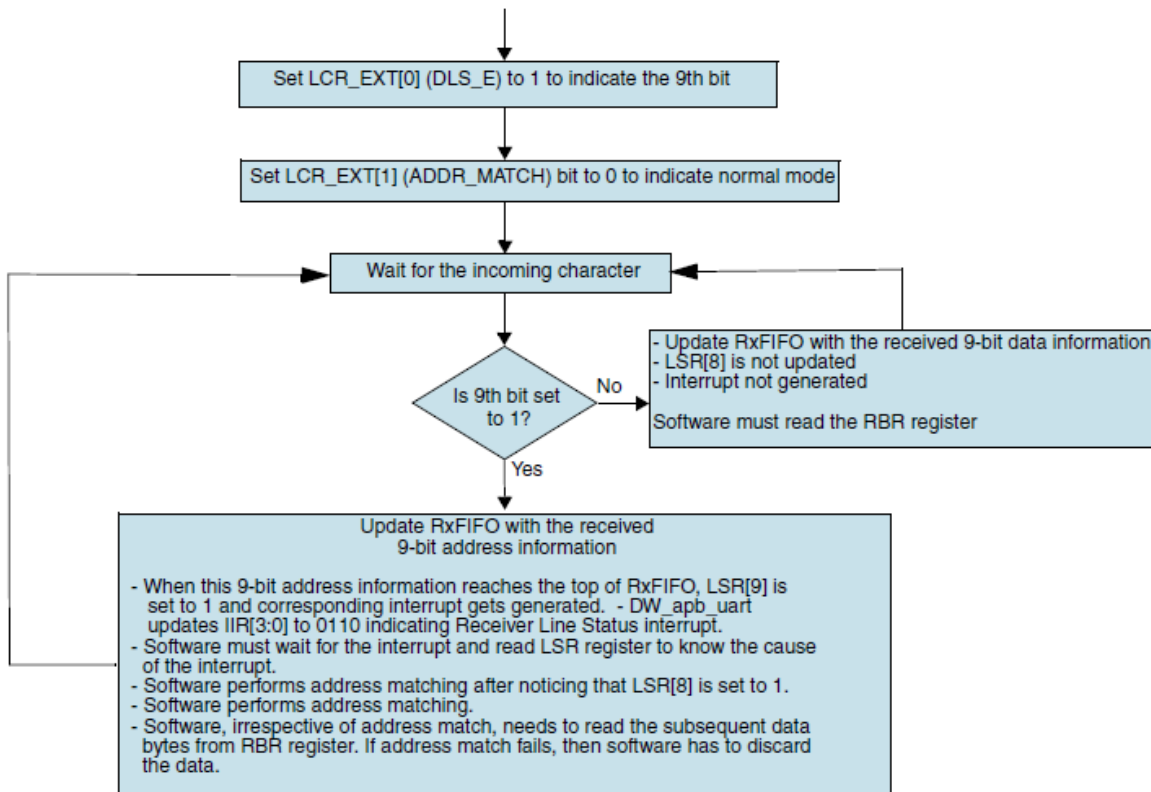


Figure 12-2-28 Software Address Match Receive Programming Mode

12.3 SPI

12.3.1 Overview

12.3.1.1 General Description

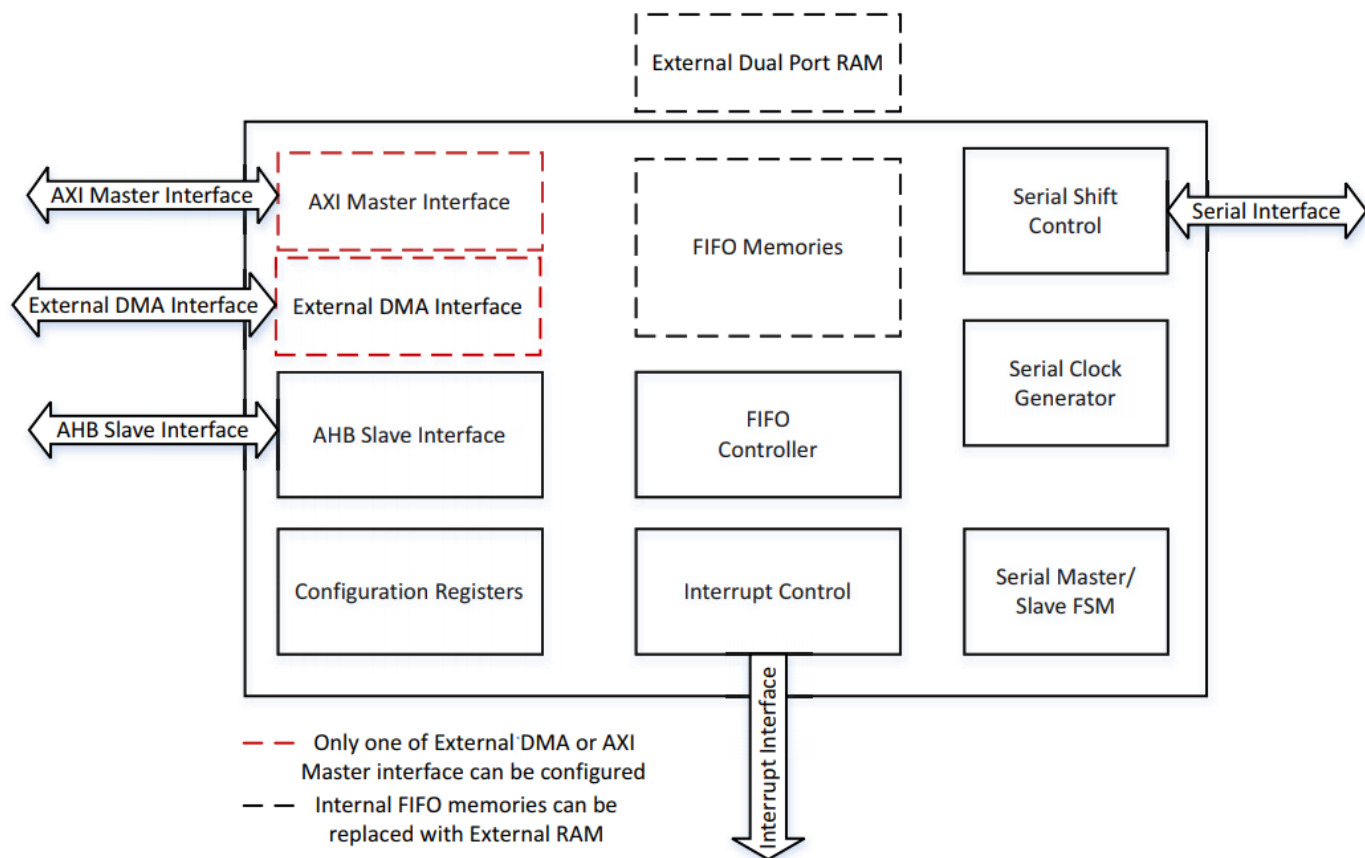
The serial peripheral interface (SPI) controller implements serial-to-parallel conversion and parallel-to-serial conversion, and serves as a master to communicate with peripherals in synchronous and serial modes.

12.3.1.2 Features

- Supports AHB Slave Interface for software programming(AMBA 3)
- Supports AXI Master Interface for Internal DMA transfers
 - AMBA 3 AXI
 - AXI data width : 64
 - AXI address width : 32
- Supports two separate FIFOs. One acts as an RX FIFO and the other one acts as a TX FIFO. Each of them is 256-location deep.
- Enhanced (Dual/Quad) SPI features:
 - Programmable address,instruction,wait cycles,and data frame size
 - Supported for the Master configurations
- Supports programmable serial data frame length: 4 bits to 32 bits.
- Programmable delay on the sample time of received serial data bit(rxd)
- Programmable clock bit rate

12.3.2 Block Diagram

Figure 12-3-1 shows the block diagram for the SPI.



12.3.3 Function Description

12.3.3.1 Peripheral Bus Timings

The clock polarity (SCPOL) configuration parameter determines whether the inactive state of the serial clock is high or low. To transmit data, both SPI peripherals must have identical serial clock phase (SCPH) and clock polarity (SCPOL) values.

When the configuration parameter SCPH = 0, data transmission begins on the falling edge of the slave select signal. The first data bit is captured by the peripherals on the first edge of the serial clock; therefore, valid data must be present on the txd and rxd lines prior to the first serial clock edge. Figure 12-3-2 shows a timing diagram for a single SPI data transfer with SCPH = 0. The serial clock is shown for configuration parameters SCPOL = 0 and SCPOL = 1.

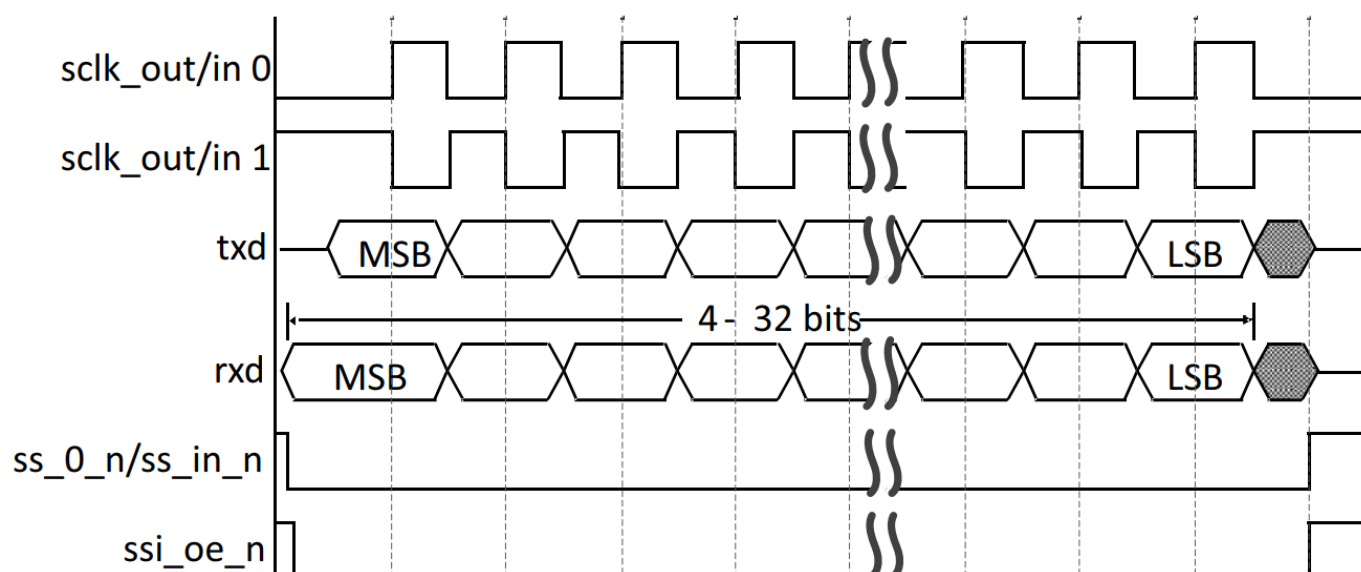


Figure 12-3-2 SPI Serial Format (SCPH)

The meanings of the abbreviations are as follows:

- MSB: most significant bit
- LSB: least significant bit Two different modes of continuous data transfers are supported when $SCPH = 0$; the selection of the desired operation mode is done by configuring CTRL0 field of the SSTE register:
- When CTRL0 field of the SSTE register is programmed to 1, SPI toggles the slave select signal between frames, and the serial clock is held to its default value while the slave select signal is active; this operating mode is illustrated in Figure 12-3-3

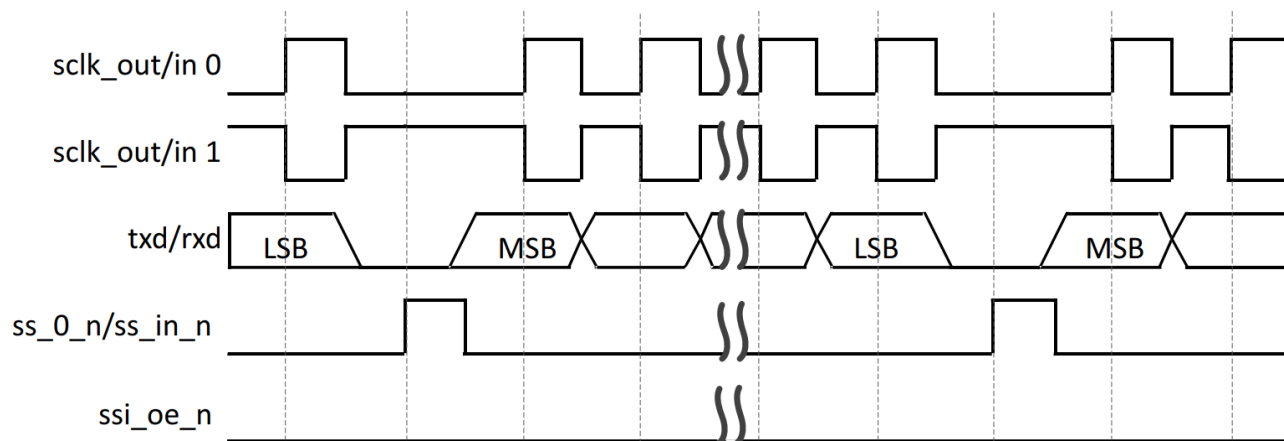


Figure 12-3-3 Serial Format Continuous Transfers ($SCPH = 0$) when CTRL0.SSTE = 1

- When CTRL0 field of the SSTE register is programmed to 0, the slave select signal stays low and the serial clock runs continuously for the duration of the transfer; this operating mode is illustrated in Figure 12-3-4.

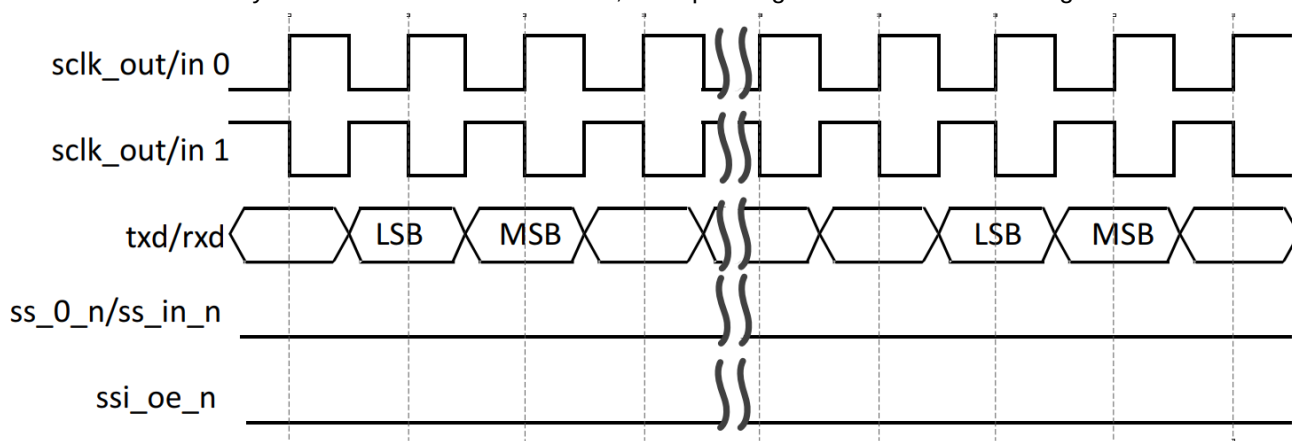


Figure 12-3-4 Serial Format Continuous Transfers ($SCPH = 0$) when CTRL0.SSTE = 0

When the configuration parameter $SCPH = 1$, both master and slave peripherals begin transmitting data on the first serial clock edge after the slave select line is activated. The first data bit is captured on the second (trailing) serial clock edge. Data are propagated by the master and slave peripherals on the leading edge of the serial clock. During continuous data frame transfers, the slave select line may be held active-low until the last bit of the last frame has been captured. Figure 12-3-5 shows the timing diagram for the SPI format when the configuration parameter $SCPH = 1$.

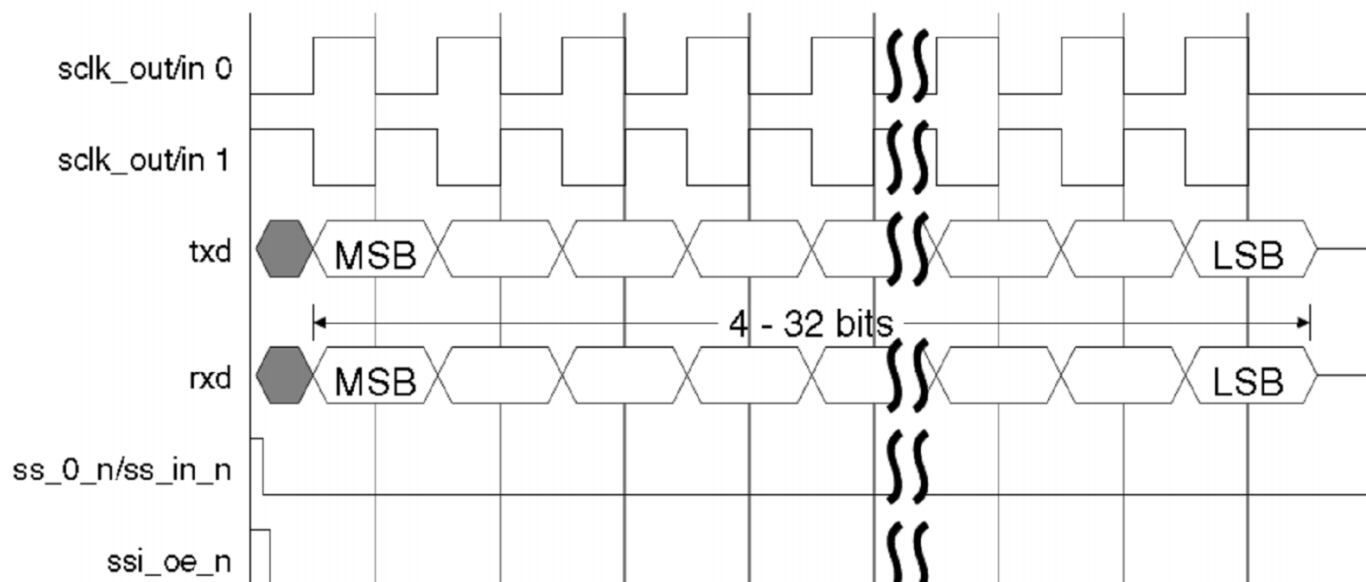


Figure 12-3-5 SPI Serial Format (SCPH = 1)

Continuous data frames are transferred in the same way as single frames, with the MSB of the next frame following directly after the LSB of the current frame. The slave select signal is held active for the duration of the transfer. Figure 12-3-6 shows the timing diagram for continuous SPI transfers when the configuration parameter SCPH = 1.

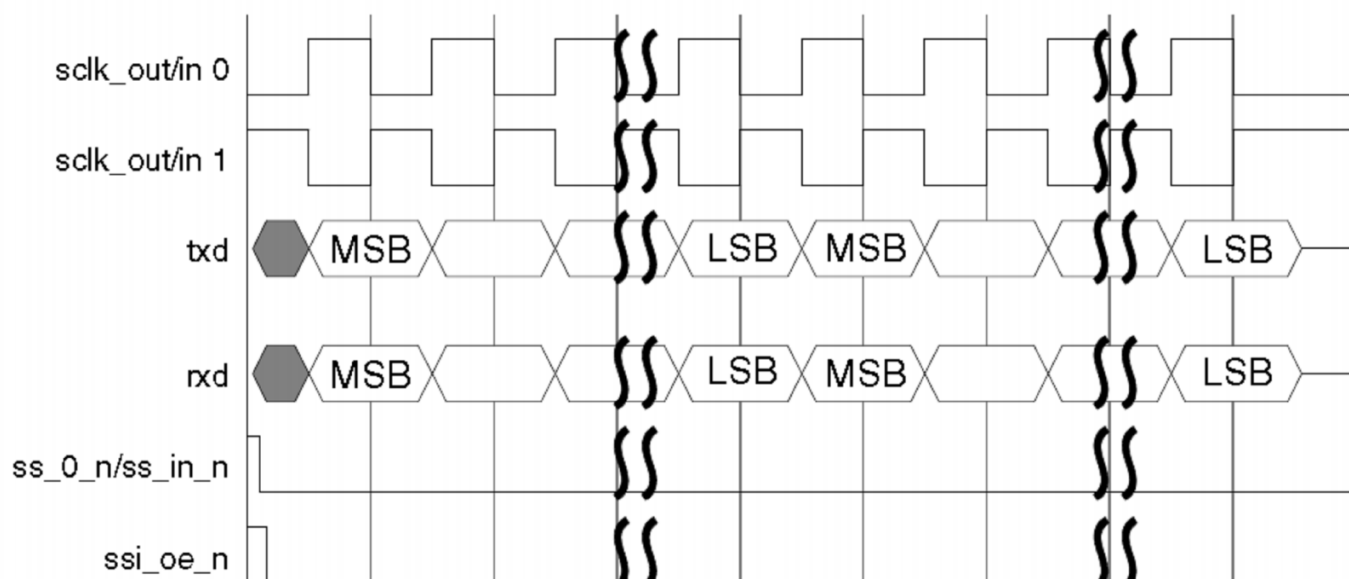


Figure 12-3-6 SPI Serial Format Continuous Transfer (SCPH = 1)

For transmit and receive transfers (transfer mode field (11:10) of the Control Register 0 = 2'b00), data transmitted from the SPI to the external serial device is written into the transmit FIFO. Data received from the external serial device into the SPI is pushed into the receive FIFO. Figure 12-3-7 shows the FIFO levels prior to the beginning of a serial transfer and the FIFO levels on completion of the transfer. In this example, two data words are transmitted from the SPI to the external serial device in a continuous transfer. The external serial device also responds with two data words for the SPI.

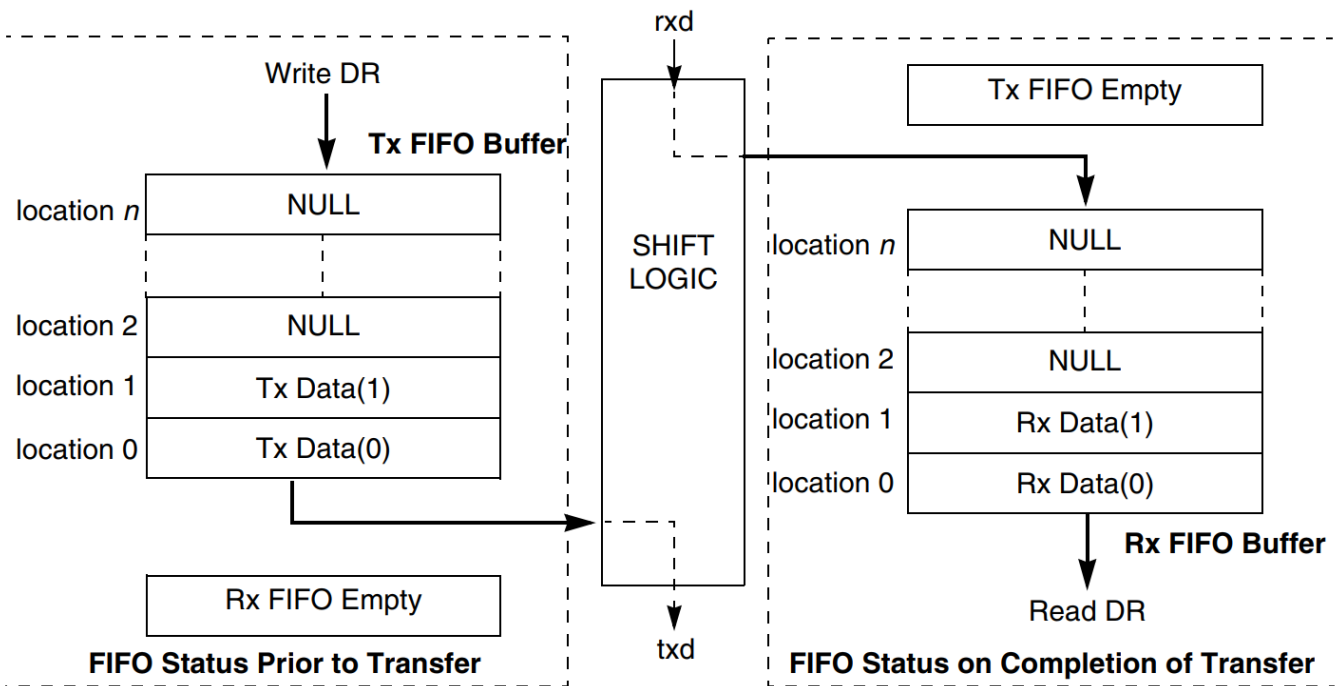


Figure 12-3-7 FIFO Status for Transmit & Receive Transfers]

For transmit only transfers (transfer mode field (11:10) of the Control Register 0 = 2'b01), data transmitted from the SPI to the external serial device is written into the transmit FIFO. As the data received from the external serial device is deemed invalid, it is not stored in the SPI receive FIFO. Figure 12-3-8 shows the FIFO levels prior to the beginning of a serial transfer and the FIFO levels on completion of the transfer. In this example, two data words are transmitted from the SPI to the external serial device in a continuous transfer.

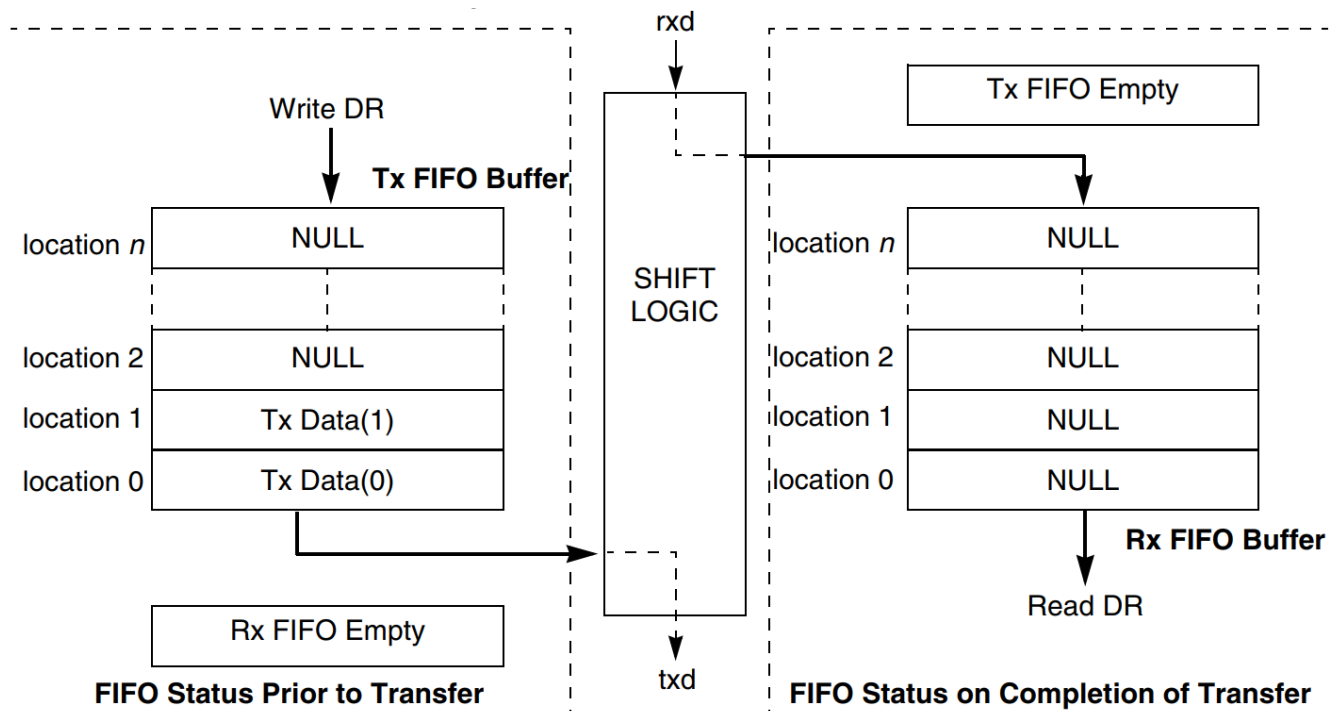


Figure 12-3-8 FIFO Status for Transmit Only SPI Transfers

For receive only transfers (transfer mode field (11:10) of the Control Register 0 = 2'b10), data transmitted from the SPI to the external serial device is invalid, so a single dummy word is written into the transmit FIFO to begin the serial transfer. The txd output from the SPI is held at a constant logic level for the duration of the serial transfer. Data received from the external serial device into the SPI is pushed into the receive FIFO. Figure 12-3-9 shows the FIFO levels prior to the beginning of a serial transfer and the FIFO levels on completion of the transfer. In this example, two data words are received by the SPI from the external serial device in a continuous transfer.

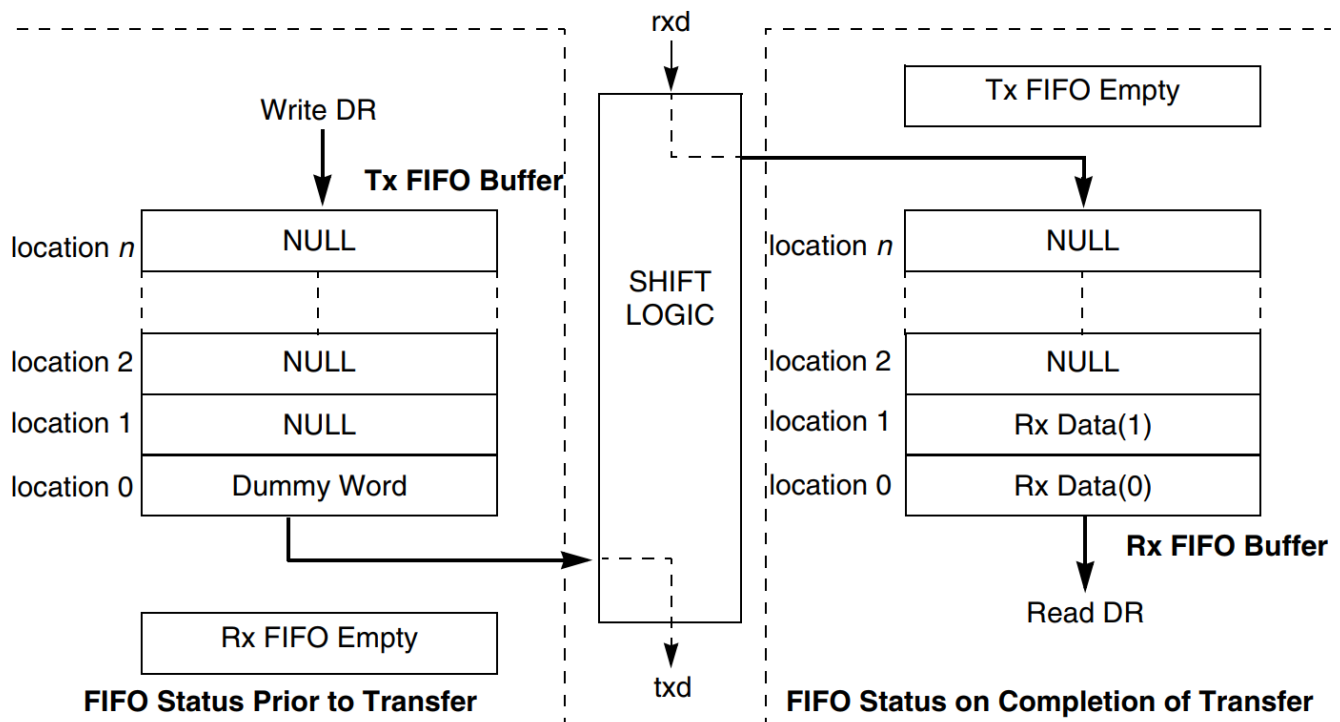


Figure 12-3-9 FIFO Status for Receive Only SPI Transfers

For `eeeprom_read` transfers (transfer mode field [11:10] of the Control Register 0 = 2'b11), opcode and/or EEPROM address are written into the transmit FIFO. During transmission of these control frames, received data is not captured by the SPI master. After the control frames have been transmitted, receive data from the EEPROM device is stored in the receive FIFO. Figure 12-3-10 shows the FIFO levels prior to the beginning of a serial transfer and the FIFO levels on completion of the transfer. In this example, one opcode and an upper and lower address are transmitted to the EEPROM device, and eight data frames are read from the EEPROM device and stored in the receive FIFO of the SPI master.

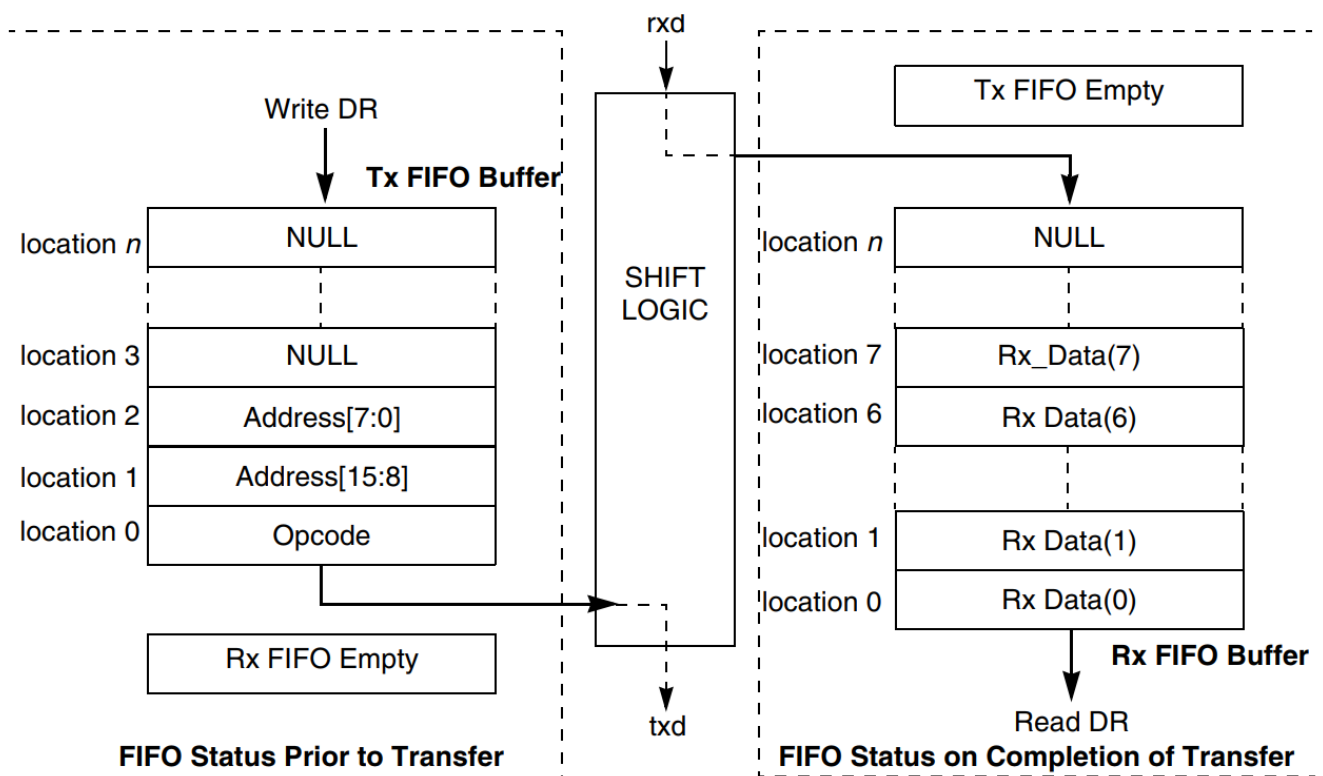


Figure 12-3-10 FIFO Status for EEPROM Read Transfer Mode

12.3.3.2 RXD Sample Delay

Additional logic can be included in the design in order to delay the default sample time of the rxd signal. This additional logic can help to increase the maximum achievable frequency on the serial bus. The SPI uses RX_SAMPLE_DLY register to change the sampling point of rxd signal.

- If RX_SAMPLE_DLY field of the SE register is set to 0, then SPI delays the sampling point by the programmed number of ssi_clk cycles.
- If RX_SAMPLE_DLY field of the SE register is set to 1, then SPI delays sampling point by programmed number of ssi_clk cycles + 0.5 * (ssi_clk period). Thus, the sampling is done on negative edge of ssi_clk signal as described in the figure Figure 12-3-11.

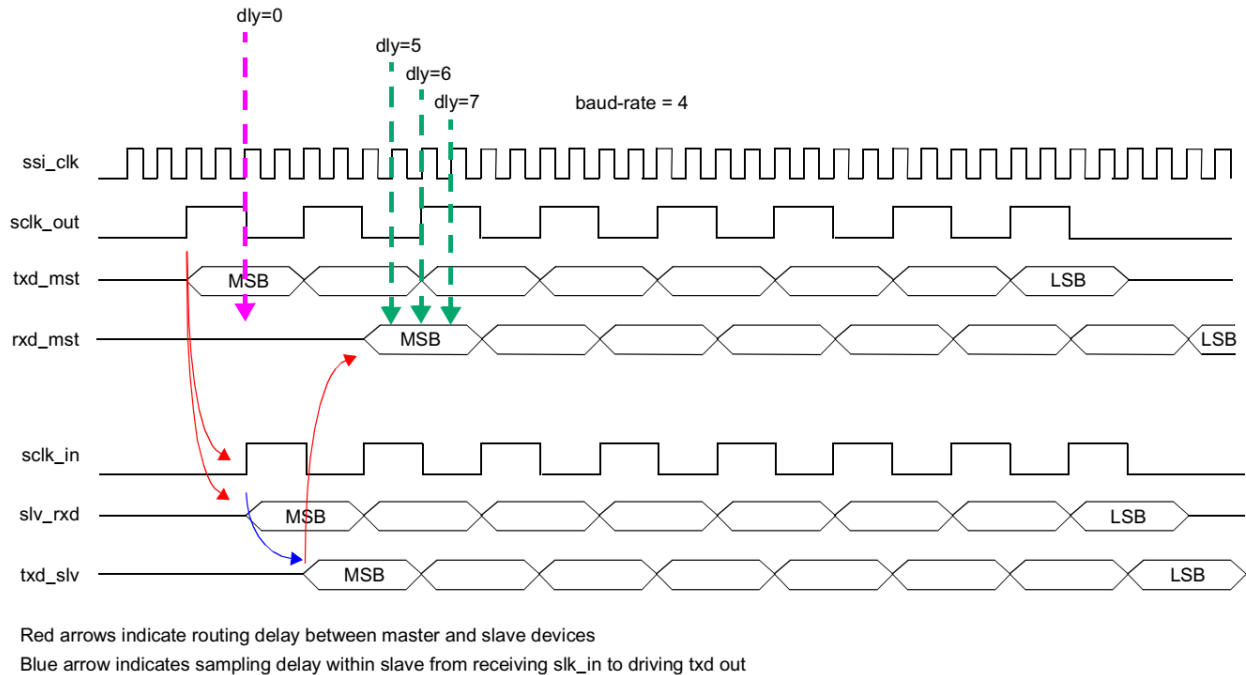


Figure 12-3-11 Effects of Round-Trip Routing Delays on the sclk_out Signal The slave uses the sclk_out signal from the master as a strobe in order to drive rxd signal data onto the serial bus. Routing and sampling delays on the sclk_out signal by the slave device can mean that the rxd bit has not stabilized to the correct value before the master samples the rxd signal. Figure 12-11 shows an example of how a routing delay on the rxd signal can result in an incorrect rxd value at the default time when the master samples the port. Note:

- If RX_SAMPLE_DLY field of the SE register is set to 0, then the SPI delays the sampling point by the programmed number of ssi_clk cycles.
- If RX_SAMPLE_DLY field of the SE register is set to 1, then the SPI delays sampling point by programmed number of ssi_clk cycles + 0.5 * (ssi_clk period). Thus, the sampling is done on negative edge of ssi_clk signal.
- RX_SAMPLE_DLY field of the SE register is available only if SSIC_HAS_RX_SAMPLE_DELAY configuration parameter is set to 2 (Value Use Both Positive and Negative Edges).

12.3.3.3 Clock Ratios

The frequency of sclk_out signal can be derived from the following equation. $F_{sclk\_out} = F_{ssi\_clk} / SCKDV$ Where, SCKDV is a programmable register holding any even value in the range 0 – 65,534. If SCKDV = 0, sclk_out is disabled. The sclk_out line only toggles when an active transfer is in progress. At all other time, it is held in inactive state, as define by the serial protocol under which it operates. Figure 12-3-12 illustrates the maximum ratio between sclk_out signal and ssi_clk signal.

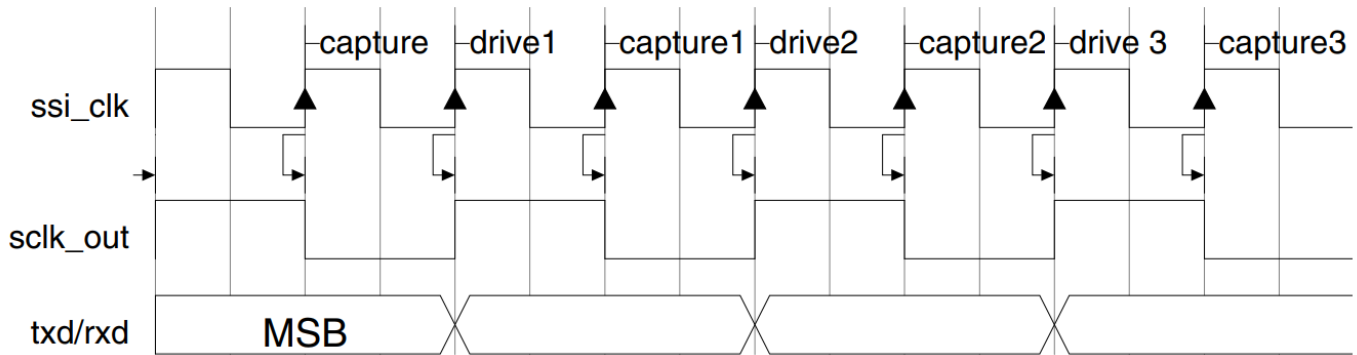


Figure 12-3-12 Maximum sclk_out/ssi_clk Ratio

12.3.3.4 Internal DMA Feature

The SPI internal DMA feature is responsible for data transfer from an internal memory (or on chip memory) to an external SPI device, and vice versa.

- The internal DMA operation is supported only in Dual/Quad SPI mode of operation.
- When internal DMA is enabled, any read or write operation to Data register (DR) results in error response from the AHB slave interface.
- Data Mask feature is not supported in internal DMA mode.
- For internal DMA operation, the address length on SPI interface can be maximum of 32 bits.

Write Operation

In write operation, the SPI device acts as destination and internal memory acts as a source. SPI fetches the data using the AXI master interface and transmits the data on the SPI device. Following is the programming requirement for this operation:

1. Program CTRLR0.TMOD = 1 (SPI write).
2. Program the CTRLR0.SPI_FRF register field with the frame format for SPI transfer.
3. Set transfer characteristics for SPI operation in the CTRLR0, BAUDR, TXFTLR.TXFTHR, and SPI_CTRLR0 registers.
4. Program CTRLR1.NDF with the amount of data to be transmitted to the device.
5. Enable DMA transfers using the DMACR.IDMAE register field.
6. Set the source transfer width in the DMACR.ATW field.
7. Set AXI transfer characteristics using the DMACR.AID, DMACR.AINC, DMACR.APROT, DMACR.ACACHE, and AXIARLEN.ARLN register fields.
8. Set Source address in the AXIAR0, and AXIAR1 registers.
9. Set device destination address in the SPIAR register.
10. Set instruction code in the SPIDR.SPI_INST register field.
11. Enable SPI using the SSIENR register. After SPI is enabled, SPI starts fetching the data from the source using transfer characteristics set for the AXI interface. AXI interface fetches the data using the AXIARLEN.ARLN register field burst length until the space left in FIFO is less than the AXIARLEN.ARLN register field decode value. To get the maximum utilization of transmit FIFO, it is recommended to use the AXIARLEN.ARLN register field value such that the amount of data read in one burst is multiple of FIFO depth. If the DMACR.AINC register field is set to 0, AXI address in the AXIAR0 register increments to the next address after every AXI burst request. After SPI has enough amount of data in the transmit FIFO (specified in TXFTLR.TXFTHR field), it starts the SPI operation using the instruction code set in the SPIDR.SPI_INST register field and device address set in the SPIAR.SDAR register field. The whole block transfer is completed in single block.

- If FIFO goes empty and clock stretching is not enabled, SPI de-selects the slave and asserts transmit FIFO underflow interrupt.
- If the clock stretching is enabled, SPI stretches the clock until the data is available. After SPI transmits the required amount of data to the device, SPI clears the SSIENR register, and interrupt (ssi_done_intr(_n)) is provided to software. Note Before starting the SPI write operation from the controller, ensure that write is enabled in the SPI slave device, and it is not busy; otherwise the transfer may not complete successfully on SPI slave device. After SPI transmits the required amount of data to the device, SPI clears the SSIENR register, and interrupt is provided to software.

Read Operation

In SPI read operation, SPI device acts as source and internal memory acts as a destination device. SPI fetches the data using the SPI interface, and transmits the data on the AXI interface. Following is the programming requirement for this operation:

1. Program CTRLR0.TMOD = 2'b10 (SPI read).
2. Program the CTRLR0.SPI_FRF register field with the frame format for SPI transfer.
3. Set transfer characteristics for SPI operation in the CTRLR0, BAUDR and SPI_CTRLR0 registers.
4. Program the CTRLR1.NDF register field with the amount of data to be transmitted to the device.
5. Enable DMA transfers using the DMACR.IDMAE register field.
6. Set the source transfer width in the DMACR.ATW register field.
7. Set AXI transfer characteristics using the DMACR.AID, DMACR.AINC, DMACR.APROT, DMACR.ACACHE, and AXIAWLEN.AWLEN register fields.
8. Set source address in the AXIAR0, and AXIAR1 registers.
9. Set device destination address in the SPIAR registers.
10. Set instruction code in the SPIDR.SPI_INST register field.
11. Enable SPI using the SSIENR register. SPI starts transfer using instruction code from the SPI_DR.SPI_INST register field, and address set in SPIAR.SDAR register field. This data is stored in the receive FIFO, and then transmitted to the AXI interface. After the receive FIFO receives the data greater than or equal to AXIAWLEN.AWLEN register field, SPI starts the operation on the AXI interface. This process continues until the whole block of data (CTRLR1.NDF) is transmitted to the AXI interface. If DMACR.AINC register field is set to 0, then AXI address in AXIAR0 increments to the next address after every AXI burst request. If Receive FIFO becomes full and clock stretching is not enabled, then SPI de-selects the slave and starts a new read transfer on the next contiguous address. If the clock stretching is enabled, then SPI stretches the clock and continues to receive until FIFO has enough space. After SPI receives the required amount of data from the device, SPI clears the SSIENR register and interrupt is provided to software.

AXI Master Interface

SPI uses AXI interface to perform internal DMA operation. AXI interface consists of all five channels, a separate clock (aclk), and a reset (aresetn) signal. All control signals for AXI interface are derived from the programming registers, the table below describes the mapping of each control signal with internal register field.

Table 2-1 AXI Control Signals Mapping

Signal Name	Register Field
For Read transfers (CTRLR0.TMOD = 10)	
awid/wid	DMACR. AID
awaddr[31:0]	AXIAR0 Note: The Address should be aligned with awsize address boundary.

Signal Name	Register Field
awaddr[63:32]	AXIAR1
awsiz	DMACR. ATW
awburst	DMACR.AINC If AINC=0; INCR burst If AINC=1; FIXED burst Note: WRAP burst type is not supported
awlock	awlock signal is set to 2'b00 as SPI does not support locked or exclusive accesses.
awcache	DMACR.ACACHE
awprot	DMACR.APROT For write transfers (CTRLR0.TMOD = 01)
arid	DMACR. AID
araddr[31:0]	AXIAR0 Note: The Address should be aligned with arsize address boundary.
araddr[63:32]	AXIAR1
arsiz	DMACR. ATW
arburst	DMACR.AINC If AINC=0; INCR burst If AINC=1; FIXED burst Note: WRAP burst type is not supported
arlock	awlock signal is set to 2'b00 as SPI does not support locked or exclusive accesses.
arcache	DMACR.ACACHE
arprot	DMACR.APROT

AXI interface can receive error response from Write Response or Read data channel. In such situation SPI aborts the transfer, immediately by de-asserting the slave-select signal disables SPI, and asserts the ssi_axie_intr(_n) signal. The data present in the internal FIFO is flushed and SPI is disabled. Also, SPI updates the SR.CMPLTD_DF field with the amount of data which was successfully transferred in the last DMA transfer. You can use this information to determine where exactly the error occurred, and the amount of data transferred successfully.

12.3.3.5 Enhanced SPI Modes

SPI supports the dual, quad modes of SPI using the SSIC_SPI_MODE configuration parameter. The possible values for this parameter are Standard, Dual SPI and Quad SPI modes. When dual and quad mode is selected for this parameter, the width of txd, rxd and ssi_oe_n signals change to 2 or 4 respectively. Hence, the data is shifted out/in on more than one line, increasing the overall throughput. Dual SPI or Quad SPI modes function similarly except for the width of txd, rxd and ssi_oe_n signals. The mode of operation (write/read) can be selected using the CTRLR0.TMOD field. The following sections describe the read and write operations in Dual SPI and Quad SPI modes in detail:

Write Operation in Enhanced SPI Modes

The following register fields are used for a write operation:

- CTRLR0.SPI_FRF - Specifies the format in which the transmission happens for the frame.
- SPI_CTRLR0 (SPI Control Register 0 register) – Specifies length of instruction, address, and data.
- SPI_CTRLR0.INST_L – Specifies length of an instruction (possible values for instruction length are 0, 4, 8, or 16 bits.)
- SPI_CTRLR0.ADDR_L – Specifies address length
- CTRLR0.DFS – Specifies data length. Note: An instruction takes one FIFO location and address can take more than one FIFO locations. Both the instruction and address must be programmed in the data register (DR).

Read Operation in Enhanced SPI Modes

Wait Cycles can be programmed using SPIC_CTRLR0.WAIT_CYCLES field. The wait cycles are introduced for target slave to change their mode from input to output and the wait cycles can vary for different devices. For a READ operation, SPI sends instruction and control data once and waits until it receives NDF (CTRLR1 register) number of data frames and then de-asserts slave select signal.

To initiate a dual/quad/octal read operation, CTRLR0.SPI_FRF must be set to 01/10 respectively. This sets the transfer type, now for each read command data is transferred in the format specified in CTRLR0.SPI_FRF field.

12.3.5 Register Summary

This chapter details all possible registers in the controller. They are arranged hierarchically into maps and blocks (banks). Table 4-1 describes the registers.

Offset Address	Register	Description
0x0	CTRLR0	Control Register 0
0x4	CTRLR1	Control Register 1
0x8	SSIENR	SSI Enable Register
0xc	MWCR	Microwire Control Register
0x10	SER	Slave Enable Register
0x14	BAUDR	Baud Rate Select
0x18	TXFTLR	Transmit FIFO Threshold Level
0x1c	RXFTLR	Receive FIFO Threshold Level
0x20	TXFLR	Transmit FIFO Level Register
0x24	RXFLR	Receive FIFO Level Register
0x28	SR	Status Register
0x2c	IMR	Interrupt Mask Register
0x30	ISR	Interrupt Status Register
0x34	RISR	Raw Interrupt Status Register
0x38	TXEICR	Transmit FIFO Error Interrupt Clear Registers
0x3c	RXOICR	Receive FIFO Overflow Interrupt Clear Register
0x40	RXUICR	Receive FIFO Underflow Interrupt Clear Register
0x44	MSTICR	Multi-Master Interrupt Clear Register
0x48	ICR	Interrupt Clear Register

Offset Address	Register	Description
0x4c	DMACR	DMA Control Register
0x50	DMATDLR	DMA Transmit Data Level
0x50	AXIAWLEN	Destination Burst Length
0x54	DMARDLR	DMA Receive Data Level
0x54	AXIARLEN	Source Burst Length
0x58	IDR	Identification Register
0x5c	SSIC_VERSION_ID	Component version Register
0x60+i*0x4	DRx(for i = 0; i <= 35)	Data Register
0x68	SSI_CTRL	Control Register
0xf0	RX_SAMPLE_DELAY	RX Sample Delay Register
0xf4	SPI_CTRLR0	SPI_CTRLR0 - SPI Control Register
0xf8	DDR_DRIVE_EDGE	DDR_DRIVE_EDGE - Transmit Drive Edge Register
0x118	SPI_CTRLR1	SPI Control 1 register
0x11c	SPITECR	SPI Transmit Error Interrupt Clear Register
0x120	SPIDR	SPI Device Register
0x124	SPIAR	SPI Device Address Register
0x128	AXIAR0	AXI Address Register 0
0x12c	AXIAR1	AXI Address Register 1
0x130	AXIECR	AXI Master Error Interrupt Clear Register
0x134	DONECR	Transfer Done Clear Interrupt Clear Register

12.3.4 Register Descriptions

CTRLR0

CTRLR0 is a Control Register 0 Offset Address: 0x0 Total Reset Value:0x00004007

Bits	Access	Name	Description	Reset
31	* Varies	SSI_IS_MST	This field selects if SPI is working in Master or Slave mode Values: - 0x1 (MASTER) - 0x0 (SLAVE)	0x0
30:26	R	RSVD_CTRLR0_26_31	Reserved bits - read as zero	0x0
25	* Varies	SPI_DWS_EN	Enable Dynamic wait states in SPI mode of operation. Values: - 0x0 (DISABLE): Disable SPI Dynamic Wait State - 0x1 (ENABLE): Enable SPI Dynamic Wait State	0x0

Bits	Access	Name	Description	Reset
24	* Varies	SPI_HYPERBUS_EN	SPI Hyperbus Frame format enable: Values: 0x0 (DISABLE): Disable Hyperbus Format 0x1 (ENABLE): Enable Hyperbus Format	0x0
23:22	* Varies	SPI_FRF	SPI Frame Format: Values: - 0x0 (SPI_STANDARD): Standard SPI Format - 0x1 (SPI_DUAL): Dual SPI Format - 0x2 (SPI_QUAD): Quad SPI Format - 0x3 (SPI_OCTAL): Octal SPI Format	0x0
21:20	R	RSVD_CTRLR0_20_21	Reserved bits - read as zero	0x0
19:16	R/W	CFS	Control Frame Size.	0x0
15	R	RSVD_CTRLR0_15	Reserved bits - read as zero	0x0
14	R/W	SSTE	Slave Select Toggle Enable. Values: - 0x1 (TOGGLE_EN): ss_n line will toggle between consecutive data frames, with the serial clock (sclk) being held to its default value while ssn is high 0x0 (TOGGLE_DISABLE): ss*_n will stay low and sclk will run continuously for the duration of the transfer	0x1
13	* Varies	SRL	Shift Register Loop. Used for testing purposes only. Values: 0x1 (TESTING_MODE): Test Mode Operation - 0x0 (NORMAL_MODE): Normal mode operation	0x0
12	R/W	SLV_OE	Slave Output Enable. Values: - 0x1 (DISABLED): Slave Output is disabled -0x0 (ENABLED): Slave Output is enabled	0x0
11:10	R/W	TMOD	Transfer Mode. Values: - 0x0 (TX_AND_RX): Transmit & Receive; Not Applicable in enhanced SPI operating mode or when SSIC_HAS_TX_RX_EN is set to 0 - 0x1 (TX_ONLY): Transmit only mode; Or Write in enhanced SPI operating mode - 0x2 (RX_ONLY): Receive only mode; Or Read in enhanced SPI operating mode - 0x3 (EEPROM_READ): EEPROM Read mode; Not Applicable in enhanced SPI operating mode	0x0
9	* Varies	SCPOL	Serial Clock Polarity. Values: - 0x0 (INACTIVE_HIGH): Inactive state of serial clock is low - 0x1 (INACTIVE_LOW): Inactive state of serial clock is high	0x0

Bits	Access	Name	Description	Reset
8	* Varies	SCPH	Serial Clock Phase. Values: - 0x1 (START_BIT): Serial clock toggles at start of first bit - 0x0 (MIDDLE_BIT): Serial clock toggles in middle of first bit	0x0
7:6	* Varies	FRF	Frame Format: 0x0 (SPI): Motorola SPI Frame Format	0x0
5	R	RSVD_CTRLR0_5	Reserved bits - read as zero	0x0
4:0	R/W	DFS	Data Frame Size.Note: When SSIC_SPI_MODE is set to "Dual", "Quad" or "Octal" mode and SPI_FRF is not set to 2'b00: - DFS value must be a multiple of 2 if SPI_FRF = 01 - DFS value must be multiple of 4 if SPI_FRF = 10 - DFS value must be multiple of 8 if SPI_FRF = 11	0x7

CTRLR1

CTRLR1 is a Control Register 1 Offset Address: 0x4 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:16	R	RSVD_CTRLR1	Reserved bits - read as zero	0x0
15:0	R/W	NDF	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. When SPI_CTRLR0.CLK_STRETCH_EN=1 and TMOD = 01, this register field sets the number of data frames to be continuously transmitted by SPI. If the Transmit FIFO goes empty in-between, SPI masks the serial clock (sclk_out) and wait for rest of the data until the programmed amount of frames are transferred successfully.When the SPI is configured as a serial slave, the transfer continues for as long as the slave is selected. Therefore, this register serves no purpose and is not present when the SPI is configured as a serial slave.	0x0

SSIENR

SSIENR is a SSI Enable Register Offset Address: 0x8 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:1	R	RSVD_SSIENR	Reserved bits - read as zero	0x0

Bits	Access	Name	Description	Reset
0	R/W	SSIC_EN	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 ssi sleep output is set (after delay) to inform the system that it is safe to remove the ssi_clk, thus saving power consumption in the system. Values: - 0x1 (ENABLED): Enables SPI - 0x0 (DISABLE): Disables SPI	0x0

MWCR

MWCR is a Microwire Control Register Offset Address: 0xc Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:3	R	RSVD_MWCR	Reserved bits - read as zero	0x0
2	* Varies	MHS	Microwire Handshaking. Values: - 0x1 (ENABLED): handshaking interface is enabled - 0x0 (DISABLE): handshaking interface is disabled	0x0
1	R/W	MDD	Microwire Control. Values: - 0x1 (TRANSMIT): SSI transmits data - 0x0 (RECEIVE): SSI receives data	0x0
0	R/W	MWMOD	Microwire Transfer Mode. Values: - 0x1 (SEQUENTIAL): Sequential Transfer -0x0 (NON_SEQUENTIAL): Non-Sequential Transfer	0x0

SER

SER is a Slave Enable Register Offset Address: 0x10 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:y	R	RSVD_SER	Reserved bits - read as zero	0x0
x:0	R/W	SER	Slave Select Enable Flag.Each bit in this register corresponds to a slave select line (ss_x_n) from the SPI master. When a bit in this register is set (1), the corresponding slave select line from the master is activated when a serial transfer begins. It should be noted that setting or clearing bits in this register have no effect on the corresponding slave select outputs until a Before beginning a transfer, you should enable the bit in this register that corresponds to the slave device with which the master wants to communicate. When not operating in broadcast mode, only one bit in this field should be set.	0x0

BAUDR

BAUDR is a Baud Rate Select Offset Address: 0x14 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:16	R	RSVD_BAUDR_16_31	Reserved bits - read as zero	0x0
15:1	R/W	SCKDV	SSI Clock Divider.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: $F_{sclk\_out} = \frac{F_{ssi\_clk}}{BAUDR}$ where BAUDR is any even value between 2 and 65534 ($BAUDR = \{SCKDV * 2\}$). For example: for $F_{ssi\_clk} = 3.6864\text{MHz}$ and $BAUDR = 2$ $F_{sclk\_out} = \frac{3.6864}{2} = 1.8432\text{MHz}$	SSIC_DFLT_BAUDR/2(0x0)
0	R	RSVD_BAUDR_0	Reserved bits - read as zero	0x0

TXFTLR

TXFTLR is a Transmit FIFO Threshold Level Offset Address: 0x18 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:y	R	RSVD_TXFTHR	Reserved bits - read as zero	0x0
x:16	* Varies	TXFTHR	Transfer start FIFO level.Used to control the level of entries in transmit FIFO above which transfer will start on serial line. This register can be used to ensure that sufficient data is present in transmit FIFO before starting a write operation on serial line. In Internal DMA mode, this field sets the minimum amount of data frames present in the FIFO after which SPI starts the transfer.	0x0
15:y	R	RSVD_TXFTLR	Reserved bits - read as zero	0x0
x:0	R/W	TFT	Transmit FIFO Threshold.Controls the level of entries (or below) at which the transmit FIFO controller triggers an interrupt. The FIFO depth is configurable in the range 8-256; this register is sized to the number of address bits needed to access the FIFO. 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

RXFTLR

RXFTLR is a Receive FIFO Threshold Level Offset Address: 0x1c Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:y	R	RSVD_RXFTLR	Reserved bits - read as zero	0x0

Bits	Access	Name	Description	Reset
x:0	R/W	RFT	Receive FIFO Threshold.Controls the level of entries (or above) at which the receive FIFO controller triggers an interrupt. The FIFO depth is configurable in the range 8-256. This register is sized to the number of address bits needed to access the FIFO. 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

TXFLR

TXFLR is a Transmit FIFO Level Register Offset Address: 0x20 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:y	R	RSVD_TXFLR	Reserved bits - read as zero	0x0
x:0	R	TXTFL	Transmit FIFO Level.Contains the number of valid data entries in the transmit FIFO.	0x0

RXFLR

RXFLR is a Receive FIFO Level Register
Offset Address: 0x24 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:y	R	RSVD_RXFLR	Reserved bits - read as zero	0x0
x:0	R	RXTFL	Receive FIFO Level.Contains the number of valid data entries in the receive FIFO.	0x0

SR

SR is a Status Register Offset Address: 0x28 Total Reset Value:0x00000006

Bits	Name	Access	Description	Reset
31:15	R	CMPLTD_DF	Completed Data frames This field indicates total data frames transferred in the previous internal DMA transfer	0x0
14:7	R	RSVD_SR *	Reserved bits - read as zero	0x0
6	R	DCOL	Data Collision Error.This bit will be set if ss_in_n input is asserted by other master, when the SPI master is in the middle of the transfer. This informs the processor that the last data transfer was halted before completion. This bit is cleared when read. Values: - 0x1 (TX_COLLISION_ERROR): Transmit Data Collision Error - 0x0 (NO_ERROR_CONDITION): No Error	0x0
5	R	TXE	Transmission Error. Values: - 0x1 (TX_ERROR): Transmission Error - 0x0 (NO_ERROR): No Error	0x0

Bits	Name	Access	Description	Reset
4	R	RFF	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. Values: - 0x1 (FULL): Receive FIFO is full - 0x0 (NOT_FULL): Receive FIFO is not full	0x0
3	R	RFNE	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. Values: - 0x1 (NOT_EMPTY): Receive FIFO is not empty - 0x0 (EMPTY): Receive FIFO is empty	0x0
2	R	TFE	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. Values: - 0x1 (EMPTY): Transmit FIFO is empty - 0x0 (NOT_EMPTY): Transmit FIFO is not empty	0x1
1	R	TFNF	Transmit FIFO Not Full. Set when the transmit FIFO contains one or more empty locations, and is cleared when the FIFO is full. Values: - 0x1 (NOT_FULL): Tx FIFO is not Full - 0x0 (FULL): Tx FIFO is full	0x1
0	R	BUSY	SSI Busy Flag. When set, indicates that a serial transfer is in progress; when cleared indicates that the SPI is idle or disabled. Values: - 0x1 (ACTIVE): SPI is actively transferring data - 0x0 (INACTIVE): SPI is idle or disabled	0x0

IMR

IMR is an Interrupt Mask Register Offset Address: 0x2c Total Reset Value: 0x0000001F

Bits	Access	Name	Description	Reset
31:12	R	RSVD_12_32_IMR	Reserved bits - read as zero	0x0
11	* Varies	DONEM	SSI Done Interrupt Mask Values: - 0x1 (UNMASKED): ssi_done_intr interrupt is not masked - 0x0 (MASKED): ssi_done_intr interrupt is masked	0x0
10	* Varies	SPITEM	SPI Transmit Error Interrupt Mask Values: - 0x1 (UNMASKED): ssi_spite_intr interrupt is not masked - 0x0 (MASKED): ssi_spite_intr interrupt is masked	0x0
9	R	RSVD_9_IMR	Reserved bits - read as zero	0x0

Bits	Access	Name	Description	Reset
8	* Varies	AXIEM	AXI Error Interrupt Mask Values: - 0x1 (UNMASKED): ssi_axie_intr interrupt is not masked - 0x0 (MASKED): ssi_axie_intr interrupt is masked	0x0
7	* Varies	TXUIM	Transmit FIFO Underflow Interrupt Mask Values: - 0x1 (UNMASKED): ssi_txu_intr interrupt is not masked - 0x0 (MASKED): ssi_txu_intr interrupt is masked	0x0
6	* Varies	XR XOIM	XIP Receive FIFO Overflow Interrupt Mask Values: - 0x1 (UNMASKED): ssi_xrxo_intr interrupt is not masked - 0x0 (MASKED): ssi_xrxo_intr interrupt is masked	0x0
5	* Varies	MSTIM	Multi-Master Contention Interrupt Mask. This bit field is not present if SPI is configured as a serial-master device Values: - 0x1 (UNMASKED): ssi_mst_intr interrupt is not masked - 0x0 (MASKED): ssi_mst_intr interrupt is masked	0x1
4	R/W	RXFIM	Receive FIFO Full Interrupt Mask Values: - 0x1 (UNMASKED): ssi_rxf_intr interrupt is not masked - 0x0 (MASKED): ssi_rxf_intr interrupt is masked	0x1
3	R/W	RXOIM	Receive FIFO Overflow Interrupt Mask Values: - 0x1 (UNMASKED): ssi_rxo_intr interrupt is not masked - 0x0 (MASKED): ssi_rxo_intr interrupt is masked	0x1
2	R/W	RXUIM	Receive FIFO Underflow Interrupt Mask Values: 0x1 (UNMASKED): ssi_rxu_intr interrupt is not masked 0x0 (MASKED): ssi_rxu_intr interrupt is masked	0x1
1	R/W	TXOIM	Transmit FIFO Overflow Interrupt Mask Values: - 0x1 (UNMASKED): ssi_txo_intr interrupt is not masked - 0x0 (MASKED): ssi_txo_intr interrupt is masked	0x1
0	R/W	TXEIM	Transmit FIFO Empty Interrupt Mask Values: - 0x1 (UNMASKED): ssi_txe_intr interrupt is not masked - 0x0 (MASKED): ssi_txe_intr interrupt is masked	0x1

ISR

ISR is an Interrupt Status Register

Offset Address: 0x30 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:12	R	RSVD_12_32_RISR	Reserved bits - read as zero	0x0

Bits	Access	Name	Description	Reset
11	R	DONES	SSI Done Interrupt Status Values: - 0x1 (ACTIVE): ssi_done_intr interrupt is active after masking - 0x0 (INACTIVE): ssi_done_intr interrupt is not active after masking	0x0
10	R	SPITES	SPI Transmit Error Interrupt Values: - 0x1 (ACTIVE): ssi_spite_intr interrupt is active after masking - 0x0 (INACTIVE): ssi_spite_intr interrupt is not active after masking	0x0
9	R	RSVD_9_RISR	Reserved bits - read as zero	0x0
8	R	AXIES	AXI Error Interrupt Status Values: - 0x1 (ACTIVE): ssi_axie_intr interrupt is active after masking - 0x0 (INACTIVE): ssi_axie_intr interrupt is not active after masking	0x0
7	R	TXUIS	Transmit FIFO Underflow Interrupt Status Values: - 0x1 (ACTIVE): ssi_txu_intr interrupt is active after masking - 0x0 (INACTIVE): ssi_txu_intr interrupt is not active after masking	0x0
6	R	XR XOIS	XIP Receive FIFO Overflow Interrupt Status Values: - 0x1 (ACTIVE): ssi_xrxo_intr interrupt is active after masking - 0x0 (INACTIVE): ssi_xrxo_intr interrupt is not active after masking	0x0
5	R	MSTIS	Multi-Master Contention Interrupt Status. This bit field is not present if the SPI is configured as a serial-slave device. Values: - 0x1 (ACTIVE): ssi_mst_intr interrupt is active after masking - 0x0 (INACTIVE): ssi_mst_intr interrupt is not active after masking	0x0
4	R	RXFIS	Receive FIFO Full Interrupt Status Values: - 0x1 (ACTIVE): ssi_rxf_intr interrupt is active after masking - 0x0 (INACTIVE): ssi_rxf_intr interrupt is not active after masking	0x0
3	R	RXOIS	Receive FIFO Overflow Interrupt Status Values: - 0x1 (ACTIVE): ssi_rxo_intr interrupt is active after masking - 0x0 (INACTIVE): ssi_rxo_intr interrupt is not active after masking	0x0

Bits	Access	Name	Description	Reset
2	R	RXUIS	Receive FIFO Underflow Interrupt Status Values: - 0x1 (ACTIVE): ssi_rxu_intr interrupt is active after masking - 0x0 (INACTIVE): ssi_rxu_intr interrupt is not active after masking	0x0
1	R	TXOIS	Transmit FIFO Overflow Interrupt Status Values: - 0x1 (ACTIVE): ssi_txo_intr interrupt is active after masking - 0x0 (INACTIVE): ssi_txo_intr interrupt is not active after masking	0x0
0	R	TXEIS	Transmit FIFO Empty Interrupt Status Values: - 0x1 (ACTIVE): ssi_txe_intr interrupt is active after masking - 0x0 (INACTIVE): ssi_txe_intr interrupt is not active after masking	0x0

RISR

RISR is a Raw Interrupt Status Register

Offset Address: 0x34 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:12	R	RSVD_12_32_RISR	Reserved bits - read as zero	0x0
11	R	DONER	SSI Done Interrupt Raw Status Values: - 0x1 (ACTIVE): ssi_done_intr interrupt is active prior to masking - 0x0 (INACTIVE): ssi_done_intr interrupt is not active prior to masking	0x0
10	R	SPITER	SPI Transmit Error Interrupt status.This bit gets set, If SPI Master fails to get a READY status from the slave until the amount of time defined in SPI_CTRLR1.MAX_WS field, then it will stop the SPI transfer and the FIFO is flushed (in case of write operation). Values: - 0x1 (ACTIVE): ssi_spite_intr interrupt is active prior to masking - 0x0 (INACTIVE): ssi_spite_intr interrupt is not active prior to masking	0x0
9	R	RSVD_9_RISR	Reserved bits - read as zero	0x0
8	R	AXIER	AXI Error Interrupt Raw Status Values: -0x1 (ACTIVE): ssi_axie_intr interrupt is active prior to masking - 0x0 (INACTIVE): ssi_axie_intr interrupt is not active prior to masking	0x0

Bits	Access	Name	Description	Reset
7	R	TXUIR	Transmit FIFO Underflow Interrupt Raw Status Values: - 0x1 (ACTIVE): ssi_txu_intr interrupt is active prior to masking - 0x0 (INACTIVE): ssi_txu_intr interrupt is not active prior masking	0x0
6	R	XR XOIR	XIP Receive FIFO Overflow Raw Interrupt Status Values: - 0x1 (ACTIVE): ssi_xrxo_intr interrupt is active prior to masking - 0x0 (INACTIVE): ssi_xrxo_intr interrupt is not active prior masking	0x0
5	R	MSTIR	Multi-Master Contention Raw Interrupt Status. Values: - 0x1 (ACTIVE): ssi_mst_intr interrupt is active prior to masking - 0x0 (INACTIVE): ssi_mst_intr interrupt is not active prior masking	0x0
4	R	RXFIR	Receive FIFO Full Raw Interrupt Status Values: - 0x1 (ACTIVE): ssi_rxf_intr interrupt is active prior to masking - 0x0 (INACTIVE): ssi_rxf_intr interrupt is not active prior masking	0x0
3	R	RXOIR	Receive FIFO Overflow Raw Interrupt Status Values: - 0x1 (ACTIVE): ssi_rxo_intr interrupt is active prior to masking - 0x0 (INACTIVE): ssi_rxo_intr interrupt is not active prior masking	0x0
2	R	RXUIR	Receive FIFO Underflow Raw Interrupt Status Values: - 0x1 (ACTIVE): ssi_rxu_intr interrupt is active prior to masking - 0x0 (INACTIVE): ssi_rxu_intr interrupt is not active prior masking	0x0
1	R	TXOIR	Transmit FIFO Overflow Raw Interrupt Status Values: - 0x1 (ACTIVE): ssi_txo_intr interrupt is active prior to masking - 0x0 (INACTIVE): ssi_txo_intr interrupt is not active prior masking	0x0
0	R	TXEIR	Transmit FIFO Empty Raw Interrupt Status Values: - 0x1 (ACTIVE): ssi_txe_intr interrupt is active prior to masking - 0x0 (INACTIVE): ssi_txe_intr interrupt is not active prior masking	0x0

TXEICR

TXEICR is a Transmit FIFO Error Interrupt Clear Registers Offset Address: 0x38 Total Reset Value:0x0

Bits	Access	Name	Description	Reset

Bits	Access	Name	Description	Reset
31:1	R	RSVD_TXEICR	Reserved bits - read as zero	0x0
0	RC	TXEICR	Clear Transmit FIFO Overflow/Underflow Interrupt. This register reflects the status of the interrupt. A read from this register clears the ssi_txo_intr/ssi_txu_intr interrupt; writing has no effect.	0x0

RXOICR

RXOICR is a Receive FIFO Overflow Interrupt Clear Register Offset Address: 0x3c Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:1	R	RSVD_RXOICR	Reserved bits - read as zero	0x0
0	RC	RXOICR	Clear Receive FIFO Overflow Interrupt. This register reflects the status of the interrupt. A read from this register clears the ssi_rxo_intr interrupt; writing has no effect.	0x0

RXUICR

RXUICR is a Receive FIFO Underflow Interrupt Clear Register Offset Address: 0x40 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:1	R	RSVD_RXUICR	Reserved bits - read as zero	0x0
0	RC	RXUICR	Clear Receive FIFO Underflow Interrupt. This register reflects the status of the interrupt. A read from this register clears the ssi_rxu_intr interrupt; writing has no effect.	0x0

MSTICR

MSTICR is a Multi-Master Interrupt Clear Register Offset Address: 0x44 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:1	R	RSVD_MSTICR	Reserved bits - read as zero	0x0
0	RC	MSTICR	Clear Multi-Master Contention Interrupt. This register reflects the status of the interrupt. A read from this register clears the ssi_mst_intr interrupt; writing has no effect.	0x0

ICR

ICR is an Interrupt Clear Register Offset Address: 0x48 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:1	R	RSVD_ICR	Reserved bits - read as zero	0x0
0	RC	ICR	Clear Interrupts. This register is set if any of the interrupts below are active. A read clears the ssi_txo_intr, ssi_rxu_intr, ssi_rxo_intr, and the ssi_mst_intr interrupts. Writing to this register has no effect.	0x0

DMACR

DMACR is a DMA Control Register Offset Address: 0x4c Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:y	R	RSVD_DMCCR	Reserved bits - read as zero	0x0
x:15	* Varies	AID	AXI awid/arid signal value.	0x0
14:12	* Varies	APROT	AXI arprot/awprot signal value.	0x0
11:8	* Varies	ACACHE	AXI arcache/awcache signal value.	0x0
7	R	RSVD_DMCCR7	Reserved bits - read as zero	0x0
6	* Varies	AINC	Address Increment. Indicates whether to increment the AXI address on every transfer. 1 = Increment 0 = No Change Note: Increment aligns the address to the next DMCCR.ATW boundary	0x0
5	R	RSVD_DMCCR5	Reserved bits - read as zero	0x0
4:3	* Varies	ATW	AXI transfer width for DMA transfers mapped to arsize/awsize. This value must be less than or equal to SSIC_AXI_DW. Values: 0x0: 1 byte 0x1: 2 bytes 0x2: 4 bytes 0x3: 8 bytesNote: When SSIC_AXI_DW is set to 32 bits, if user programs this field to 0x8(3 bytes). SPI will use 4 bytes as transfer size for the AXI transfers.	0x0
4:3	* Varies	ATW	AXI transfer width for DMA transfers mapped to arsize/awsize. This value must be less than or equal to SSIC_AXI_DW. Values: 0x0: 1 byte 0x1: 2 bytes 0x2: 4 bytes 0x3: 8 bytes Note: When SSIC_AXI_DW is set to 32 bits, if user programs this field to 0x8(3 bytes). SPI will use 4 bytes as transfer size for the AXI transfers.	0x0
2	* Varies	IDMAE	Internal DMA Enable. This bit should be enabled only when CTRLR0.FRF = 0 (Motorola SPI) and CTRLR0.SPI_FRF > 0.	0x0
1	* Varies	TDMAE	Transmit DMA Enable. This bit enables/disables the transmit FIFO DMA channel. Values: - 0x1 (ENABLED): Transmit DMA enabled - 0x0 (DISABLE): Transmit DMA disabled	0x0
0	* Varies	RDMAE	Receive DMA Enable. This bit enables/disables the receive FIFO DMA channel. Values: - 0x1 (ENABLED): Receive DMA enabled - 0x0 (DISABLE): Receive DMA disabled	0x0

DMATDLR is a DMA Transmit Data Level Register Offset Address: 0x50 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:y	R	RSVD_DMATDLR	Reserved bits - read as zero	0x0
x:0	R/W	DMATDL	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

AXIAWLEN

AXIAWLEN is a Destination Burst Length Register Offset Address: 0x50 Total Reset Value:0x00000700

Bits	Access	Name	Description	Reset
31:y	R	RSVD_AXIAWLEN2	Reserved bits - read as zero	0x0
x:8	R/W	AWLEN	Destination Burst Length	0x7
7:0	R	RSVD_AXIAWLEN	Reserved bits - read as zero	0x0

DMARDLR

DMARDLR is a DMA Receive Data Level Register Offset Address: 0x54 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:y	R	RSVD_DMARDLR	Reserved bits - read as zero	0x0
x:0	R/W	DMARDL	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

AXIARLEN

AXIARLEN is an Source Burst Length Register Offset Address: 0x54 Total Reset Value:0x00000700

Bits	Access	Name	Description	Reset
31:y	R	RSVD_AXIARLEN2	Reserved bits - read as zero	0x0
x:8	R/W	ARLEN	Source Burst Length	0x7
7:0	R	RSVD_AXIARLEN	Reserved bits - read as zero	0x0

IDR

IDR is an Identification Register Offset Address: 0x58 Total Reset Value:0xha1b2c3d5

Bits	Access	Name	Description	Reset
31:0	R	IDCODE	Identification code. The register contains the peripheral's identification code, which is written into the register at configuration time using CoreConsultant.	0xha1b2c3d5

SSIC_VERSION_ID

SSIC_VERSION_ID is a component version Register Offset Address: 0x5c Total Reset Value:0xh3130332a

Bits	Access	Name	Description	Reset
31:0	R	SSIC_COMP_VERSION	Contains the hex representation of the Synopsys component version. Consists of ASCII value for each number in the version, followed by . <i>For example 31_30_33_2A represents the version 1.03.</i>	0xh3130332a

DRx(for i = 0; i <= 35)

DRx is a Data Register Offset Address:0x60 +i*0x4 Total Reset Value:0x0

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

SSI_CTRL

SSI_CTRL is a control Register Offset Address:0x68 Total Reset Value:0x00004000

Bits	Access	Name	Description	Reset
17:13	R/W	rxds_delay_num	rxds delay line number bit[0]:0:negedge 1:posedge bit[3:1]:delay number	0x2
17:13	RO	ssi2_spi_mode		0x0
10	RO	ssi2_ssi_sleep		0x0
9:8	RO	ssi1_spi_mode		0x0
7	RO	ssi1_ssi_sleep		0x0
6:5	RO	ssi0_spi_mode		0x0
4	RO	ssi0_ssi_sleep		0x0
3:2	R		Reserved bits - read as zero	0x0
0	R/W	ssi0_xip_en		0x0

RX_SAMPLE_DELAY

RX_SAMPLE_DELAY is a RX Sample Delay Register Offset Address: 0xf0 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:17	R	RSVD1_RX_SAMPLE_DLY	Reserved bits - read as zero	0x0

Bits	Access	Name	Description	Reset
16	* Varies	SE	Receive Data (rxd) Sampling Edge. This register is used to decide the sampling edge for RXD signal with ssi_clk. Then this bit is set to 1 then negative edge of ssi_clk will be used to sample the incoming data, otherwise positive edge will be used for sampling.	0x0
15:8	R	RSVD0_RX_SAMPLE_DLY	Reserved bits - read as zero	0x0
7:0	R/W	RSD	Receive Data (rxd) Sample Delay. This register is used to delay the sample of the rxd input port. Each value represents a single ssi_clk delay on the sample of rxd. Note; If this register is programmed with a value that exceeds the depth of the internal shift registers (SSIC_RX_DLY_SR_DEPTH) zero delay will be applied to the rxd sample.	0x0

SPI_CTRLR0

SPI_CTRLR0 is a SPI Control Register Offset Address: 0xf4 Total Reset Value:0x04000200

Bits	Access	Name	Description	Reset
31	R	RSVD_SPI_CTRLR0	Reserved bits - read as zero	0x0
30	* Varies	CLK_STRETCH_EN	Enables clock stretching capability in SPI transfers. In case of write, if the FIFO becomes empty SP will stretch the clock until FIFO has enough data to continue the transfer. In case of read, if the receive FIFO becomes full SPI will stop the clock until data has been read from the FIFO.	0x0
29	* Varies	XIP_PREFETCH_EN	Enables XIP pre-fetch functionality in SPI. Once enabled SPI will pre-fetch data frames from next contiguous location, to reduce the latency for the upcoming contiguous transfer. If the next XIP request is not contiguous then pre-fetched bits will be discarded.	0x0
28	R	RSVD_SPI_CTRLR0_28	Reserved bits - read as zero	0x0
27:26	* Varies	XIP_MBL	XIP Mode bits length. Sets the length of mode bits in XIP mode of operation. These bits are valid only when SPI_CTRLR0.XIP_MD_BIT_EN is set to 1. Values: - 0x0 (MBL_2): Mode bits length equal to 2- 0x1 (MBL_4): Mode bits length equal to 4- 0x2 (MBL_8): Mode bits length equal to 8- 0x3 (MBL_16): Mode bits length equal to 16	0x1

Bits	Access	Name	Description	Reset
25	* Varies	SPI_RXDS_SIG_EN	Enable rxds signaling during address and command phase of Hypebus transfer. This bit enables rxds signaling by Hyperbus slave devices during Command-Address (CA) phase. If the rxds signal is set to 1 during the CA phase of transfer, SPI transmits (2*SPI_CTRLR0.WAIT_CYCLES-1) wait cycles after the address phase is complete.	0x0
24	* Varies	SPI_DM_EN	SPI data mask enable bit. When this bit is enabled, the txd_dmsignal is used to mask the data on the txd data line. This bit is enabled only when the SSIC_DM_EN parameter is set to 1.	0x0
23:22	R	RSVD_SPI_CTRLR0_22_23	Reserved bits - read as zero	0x0
21	* Varies	SSIC_XIP_CONT_XFER_EN	Enable continuous transfer in XIP mode. If this bit is set to 1 then continuous transfer mode in XIP will be enabled, in this mode SPI will keep slave selected until a non-XIP transfer is detected on the AHB interface.	0x0
20	* Varies	XIP_INST_EN	XIP instruction enable bit. If this bit is set to 1 then XIP transfers will also have instruction phase. The instruction op-codes will be chosen from XIP_INCR_INST or XIP_WRAP_INST registers bases on AHB transfer type.	0x0
19	* Varies	XIP_DFS_HC	Fix DFS for XIP transfers. If this bit is set to 1 then data frame size for XIP transfers will be fixed to the programmed value in CTRLR0.DFS. The number of data frames to fetch will be determined by HSIZE and HBURST signals. If this bit is set to 0 then data frame size and number of data frames to fetch will be determined by HSIZE and HBURST signals	0x0
18	* Varies	SPI_RXDS_EN	Read data strobe enable bit. Once this bit is set to 1 SPI will use Read data strobe (rxds) to capture read data in DDR mode.	0x0
17	* Varies	INST_DDR_EN	Instruction DDR Enable bit. This will enable Dual-data rate transfer for Instruction phase.	0x0
16	* Varies	SPI_DDR_EN	SPI DDR Enable bit. This will enable Dual-data rate transfers in Dual/Quad/Octal frame formats of SPI.	0x0
15:11	R/W	WAIT_CYCLES	Wait cycles in Dual/Quad/Octal mode between control frames transmit and data reception. Specified as number of SPI clock cycles.	0x0
10	R	RSVD_SPI_CTRLR0_10	Reserved bits - read as zero	0x0

Bits	Access	Name	Description	Reset
9:8	R/W	INST_L	Dual/Quad/Octal mode instruction length in bits. Values: - 0x0 (INST_L0): No Instruction - 0x1 (INST_L4): 4 bit Instruction length - 0x2 (INST_L8): 8 bit Instruction length - 0x3 (INST_L16): 16 bit Instruction length	0x2
7	* Varies	XIP_MD_BIT_EN	Mode bits enable in XIP mode. If this bit is set to 1, then in XIP mode of operation SPI will insert mode bits after the address phase. These bits are set in register XIP_MODE_BITS register. The length of mode bits is always set to 8 bits.	0x0
6	R	RSVD_SPI_CTRLR0_6	Reserved bits - read as zero	0x0
5:2	R/W	ADDR_L	This bit defines Length of Address to be transmitted. Only after this much bits are programmed in to the FIFO the transfer can begin. Values: - 0x0 (ADDR_L0): No Address - 0x1 (ADDR_L4): 4 bit Address length - 0x2 (ADDR_L8): 8 bit Address length - 0x3 (ADDR_L12): 12 bit Address length - 0x4 (ADDR_L16): 16 bit Address length - 0x5 (ADDR_L20): 20 bit Address length - 0x6 (ADDR_L24): 24 bit Address length - 0x7 (ADDR_L28): 28 bit Address length - 0x8 (ADDR_L32): 32 bit Address length - 0x9 (ADDR_L36): 36 bit Address length - 0xa (ADDR_L40): 40 bit Address length - 0xb (ADDR_L44): 44 bit Address length - 0xc (ADDR_L48): 48 bit Address length - 0xd (ADDR_L52): 52 bit Address length - 0xe (ADDR_L56): 56 bit Address length - 0xf (ADDR_L60): 60 bit Address length	0x0
1:0	R/W	TRANS_TYPE	Address and instruction transfer format. Selects whether SPI will transmit instruction/address either in Standard SPI mode or the SPI mode selected in CTRLR0.SPI_FRF field. Values: - 0x0 (TT0): Instruction and Address will be sent in Standard SPI Mode. - 0x1 (TT1): Instruction will be sent in Standard SPI Mode and Address will be sent in the mode specified by CTRLR0.SPI_FRF. - 0x2 (TT2): Both Instruction and Address will be sent in the mode specified by SPI_FRF. - 0x3 (TT3): Reserved.	0x0

DDR_DRIVE_EDGE is a Transmit Drive Edge Register Offset Address: 0xf8 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:8	R	RSVD_DDR_DRIVE_EDGE	Reserved bits - read as zero	0x0
7:0	R/W	TDE	TXD Drive edge register which decided the driving edge of transmit data.The maximum value of this register is = (BAUDR/2) -1.	0x0

SPI_CTRLR1

SPI_CTRLR1 is a SPI Control 1 register Offset Address: 0x118 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:12	R	RSVD_SPI_CTRLR1_12_32	Reserved bits - Read Only	0x0
11:8	R/W	MAX_WS	Maximum wait cycles allowed per transaction.This field indicate, up to how many times SPI slave could insert the wait states. The internal counter is incremented every time when SPI master checks for the status from the slave and receives wait response.	0x0
7:3	R	RSVD_SPI_CTRLR1_3_7	Reserved bits - Read Only	0x0
2:0	R/W	DYN_WS	SPI Dynamic Wait states field.This field is used to set the value for wait states which will be introduced when SPI slave sends BUSY status to the Master.The programmed value of wait States will be introduced before checking status again.Number of wait states = DYN_WS+1	0x0

SPITECR

SPITECR is a SPI Transmit Error Interrupt Clear Register Offset Address: 0x11c Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:1	R	RSVD_SPITECR	Reserved bits - Read Only	0x0
0	R	SPITECR	Clear SPI Transmit Error interrupt.This register will reflect the status of the interrupt.A read from this register clears the ssi_spite_intr interrupt. Writing to this register has no effect.	0x0

SPIDR

SPIDR is a SPI Device Register Offset Address: 0x120 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:16	R	RSVD_SPIDR	Reserved bits - read as zero	0x0
15:0	R/W	SPI_INST	SPI Instruction code. This instruction code will be used for SPI operation	0x0

SPIAR

SPIAR is a SPI Device Address Register
Offset Address: 0x124 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:0	R/W	SDAR	SPI Device Address Register.This address will be used during read/write on SPI interface for DMA transfer	0x0

AXIAR0

AXIAR0 is an AXI Address Register 0 Offset Address: 0x128 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:0	R/W	AXIAR_0_31	LSB for AXI address for DMA operation.	0x0

AXIAR1

AXIAR1 is an AXI Address Register 1 Offset Address: 0x12c Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:y	R	RSVD_AXIAR1	Reserved bits - read as zero	0x0
x:0	R/W	AXIAR_32_63	MSB for source address for DMA operation.	0x0

AXIECR

AXIECR is an AXI Master Error Interrupt Clear Register Offset Address: 0x130 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:1	R	RSVD_AXIECR	Reserved bit	0x0
0	RC	AXIECR	Clear AXI Error Interrupt. This register will reflect the status of the interrupt. A read from this register clears the ssi_axie_intr interrupt. Writing to this register has no effect.	0x0

DONECR

DONECR is an Transfer Done Clear Interrupt Clear Register Offset Address: 0x134 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:1	R	RSVD_DONECR	Reserved bit	0x0
0	RC	DONECR	Clear Transfer Done Interrupt. This register will reflect the status of the interrupt. A read from this register clears the ssi_done_intr interrupt.Writing to this register has no effect.	0x0

12.4 eMMC/SD/SDIO Controller

12.4.1 Overview

12.4.1.1 General Description

The eMMC/SD/SDIO controller is designed considering the future evolution of SD and eMMC standards, and also considering the scalability towards high speed requirements.

12.4.1.2 Features

- Supports SD memory and SD Input/Output (SDIO) digital interface protocol, and compliant with SD HCI specification
- Uses the same SD-HCI register set for eMMC transfers
- Supports SD memory UHS-II interface protocol
- Supports eMMC protocols including eMMC 5.1
- Supports SD-HCI Host version 4 mode or less
- Supports the following data transfer types for SD, eMMC, and UHS-II modes:
 - CPU
 - SDMA
 - ADMA2
 - ADMA3
- Supports independent controller, Slave Interface and Master Interface clocks
- Supports gating of controller base clock if Host Controller is inactive
- Support context aware functional clock gates
- Applications can gate the slave interface clock if Host Controller is inactive
- Data Buffering
 - Configurable buffer depth. Configurable in coreConsultant/coreAssembler.
 - Automatic packing/unpacking of data to fit buffer width
- Interrupt Outputs
 - Combined and separate interrupt outputs
 - Supports interrupt enabling and masking
- Supports Command Queuing Engine (CQE) and compliant with eMMC CQ HCI
 - Programmable scheduler algorithm selection of task execution
 - Supports data prefetch for back-to-back WRITE operations
- Supports Slave-Only mode as a configuration option for better area and power efficiency
- Supports tuning:
 - SD/eMMC Tuning using CMD19 (SD) or CMD21 (eMMC)
 - Mode 1 Re-Tuning – Host driver maintains the re-tune timer
 - Fully Software driven Tuning/Re-tuning operations
 - Auto-tuning or Mode 3 Re-tuning
- Support for both Flip-flop based and Latch based clock gating.
- Support for debug sideband signals **db_task_id**, **db_pyld_descr** in CQE mode.

12.4.1.3 Supported Features for Interfaces

- SD
 - Supports 4-bit interface
 - Supports UHS-I mode
 - Supports Default Speed (DS), high-speed (HS), SDR12, SDR25, SDR50 and SDR104 speed modes
 - Supports SDIO read wait
 - Supports SDIO card interrupts in both 1-bit and 4-bit modes
 - Wake up on card interrupt
- eMMC
 - Supports 4-bit/8-bit interface
 - Supports legacy, high-speed SDR, high-speed DDR, HS200, and HS400 speed modes
 - Supports boot operation and alternative boot operation

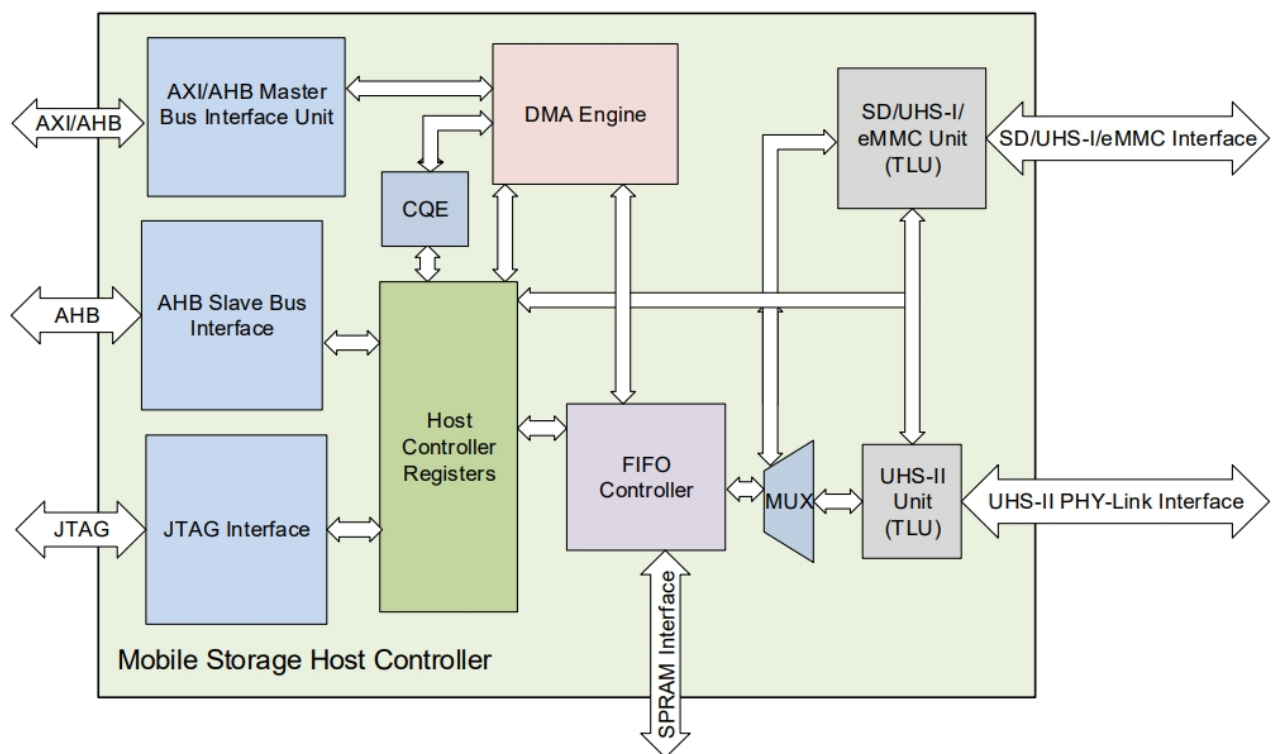
- UHS-II
 - Supports 8-bit/16-bit data width for UHS-II Link-PHY interface
 - Supports UHS-II FD156 (Full Duplex) and UHS-II HD312 (Half Duplex) modes
- AHB Slave Interface
 - Supports 32-bit data width and address width
 - Transfer size (width) used for slave interface can be less than data bus width
- AXI/AHB Master Interface
 - Supports 32-bit and 64-bit address and data width
 - Complies with the AMBA 3 AXI for Master Port specification
 - Complies with the AMBA 2 AHB for Master Port specification

12.4.2 Function Description

12.4.2.1 Overview of Architecture

This section describes the main functional building blocks of the controller, represented in Figure 12-4-1. Controller supports the following:

- Backward compatible with previous versions of the SD standards.
- Supports 4-bit data bus width and the UHS-I interface towards the physical layer in the legacy SD modes, in addition to supporting the UHS-II interface with SD-TRAN.
- Supports 4-bit and 8-bit data bus widths for the eMMC interface. The controller supports non-DMA mode and also various DMA options, such as SDMA, ADMA2, and ADMA3, as specified in the SD standards.



12.4.2.2 Programming Steps

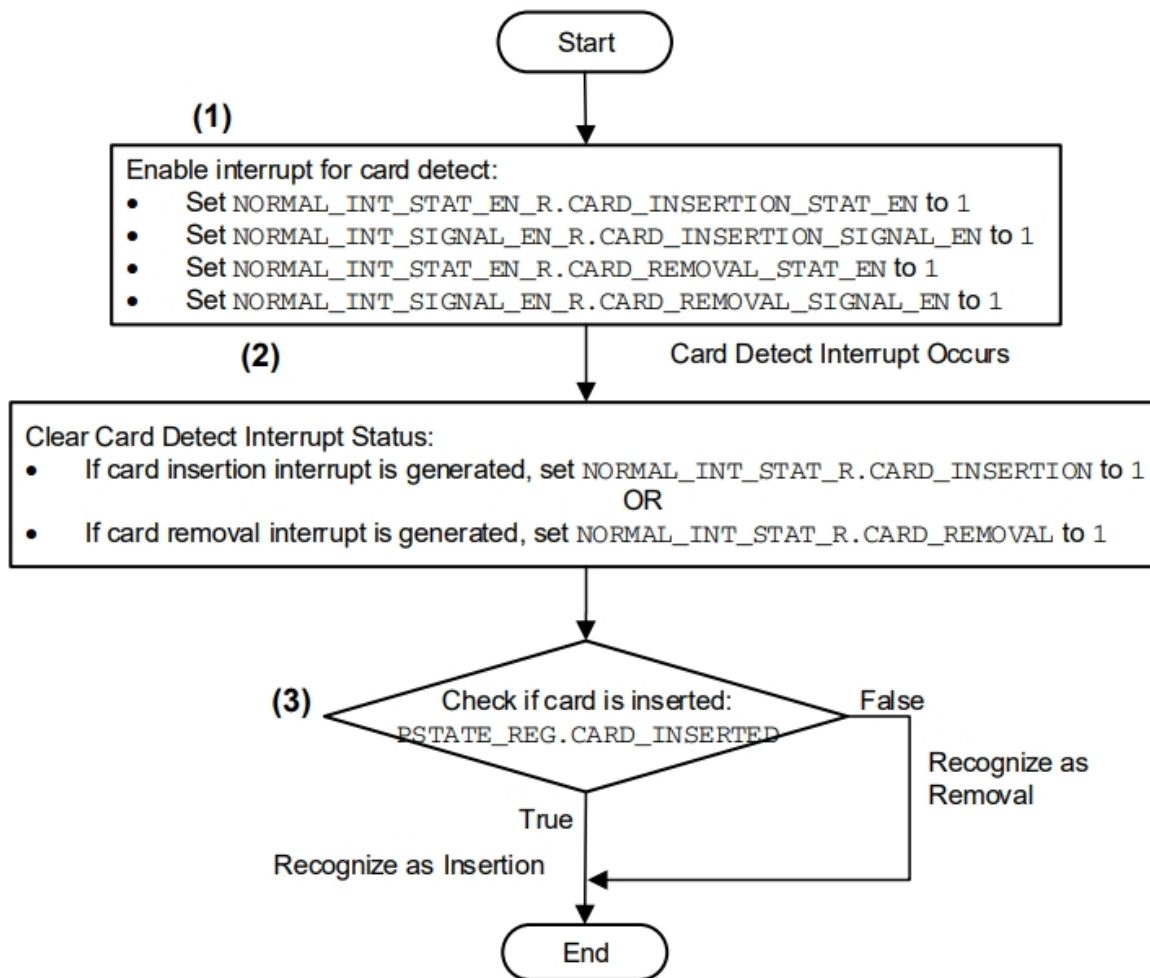
Following is the programming sequence required before starting the data transfer with the card:

1. Start providing hclk to the controller and apply its reset hresetn. This clock is required for card detection logic and for accessing most of the registers in the controller.
2. Check if the card is already inserted by reading Present State Register (PSTATE_REG). This step is required for a removable card. If the card is already inserted, then goto Step 4.

3. Wait for the card insertion interrupt. This step is required for removable card. Card detection sequence is described in “[Card Detection](#Card Detection)” .
4. Setup basic settings for the controller as described in “Host Controller Setup Sequence” on page XXX. It involves settings such as the bus power voltage level for the card, timeout counter value, setting clock generation parameters. Different setup sequences for SD/UHS-II card (“SD/UHS-II Interface” on page XXX) and eMMC card (“eMMC Device” on page XXX) are provided. This step can also be executed immediately after Step 1.
5. Enable input clocks (other than hclk) that are available in the controller. Internal clocks (aclk, bclk, tmclk and cqetmclk) are enabled followed by the card clocks (cclk_tx/cclk_rx or pclk) as discussed in the programming sequence in “Host Controller Clock Setup Sequence” on page XXX. At this stage, the controller is ready to communicate with the card.
6. Identify the type of card (SD or UHS-II card) that is connected to the controller as explained in the programming sequence in “SD/UHS-II Card Interface Detection” on page XXX. If the controller is connected to an eMMC card, then programming sequence described in “eMMC Card Interface Setup” on page XXX must be executed. This is a simplified sequence especially for an eMMC card as the type of card is always known. Both the programming sequences provide power and clock to the card.
7. After powering up the card, the card is initialized. Appropriate sequence as shown in Table 4-1 must to be executed based on type of the connected card.
8. As both host controller and card are initialized, the host can start sending commands to the card to perform the data transfer. Control and data commands are two types of commands that can be issued to the card. The control command is used to read or write any register in the card. The data command is used for writing or reading data to or from the card. Table 4-2 lists the control and data command sequences for SD, eMMC, and UHS-II cards.

12.4.2.3 Card Detection Sequence

Figure 12-4-2 shows the sequence for detecting a card. The procedure outlined in Figure 12-4-2 applies to both SD, SDIO, and UHS-II cards. This programming sequence is not required for an eMMC device as it is non-removable.



12.4.2.4 Host Controller Setup Sequence

This section discusses the host controller setup sequence for SD and UHS-II interfaces, and an eMMC device. This section discusses the following sequences:

- “SD/UHS-II Interface” on page 109
- “eMMC Device” on page 111

12.4.2.4.1 SD/UHS-II Interface

Figure 12-4-3 shows the host controller setup sequence. The register parameters set in this sequence retain their values irrespective of bus speed mode changes. Figure 12-4-3 Host Controller Setup Sequence for SD and UHS-II

12.4.2.4.2 eMMC Device

Figure 12-4-4 shows the host controller setup sequence for an eMMC device. In addition to completing the programming sequence in Figure 12-4-4, you must set the parameter `EMMC_CTRL_R.CARD_IS_EMMC = 1`.

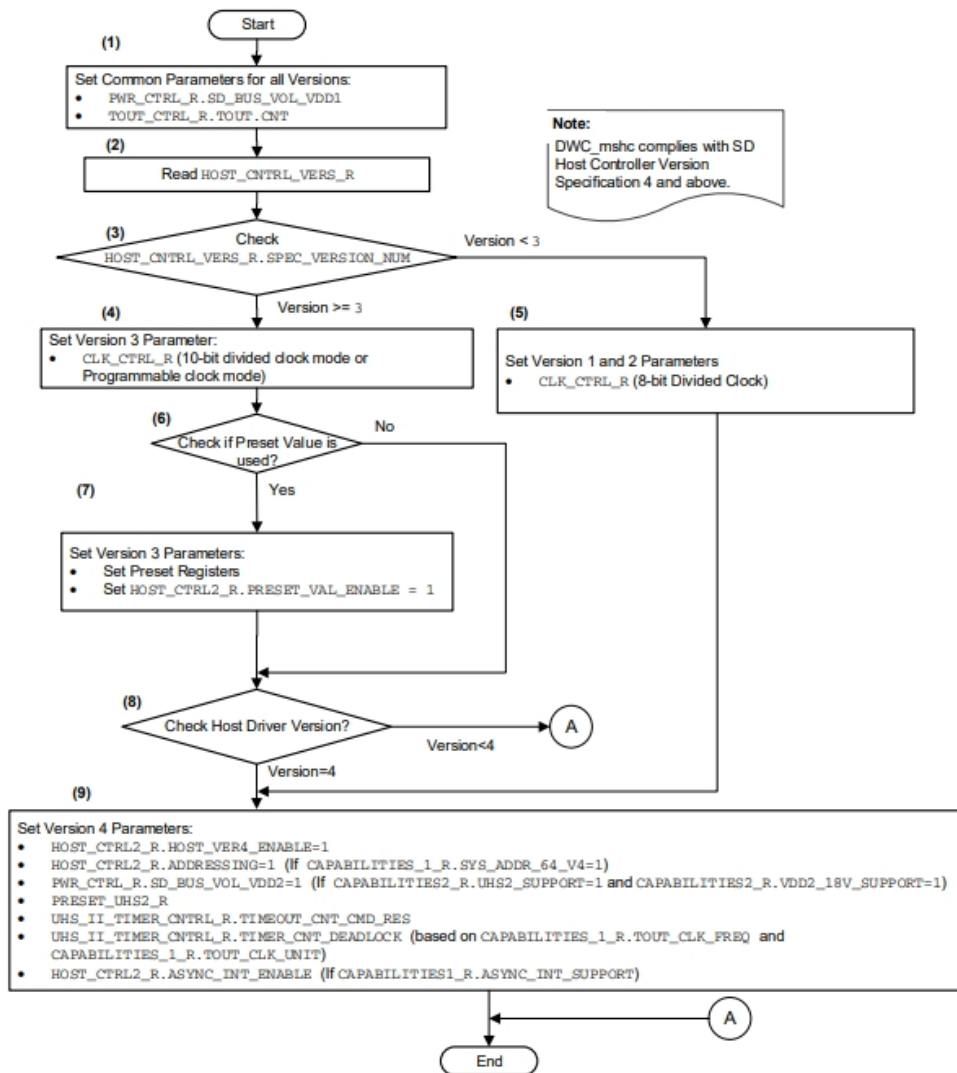


Figure 4-3 Host Controller

Setup Sequence for eMMC Interface

12.4.2.5 Clock Control

12.4.2.5.1 Host Controller Clock Setup Sequence

Figure 12-4-5 shows the sequence for setting up clocks for the controller. The controller requires number of clock signals from the system. All the input clocks are asynchronous to each other. Based on controls available for enabling clocks required for the controller, they are categorized into two types, namely Internal clock and Card clock.

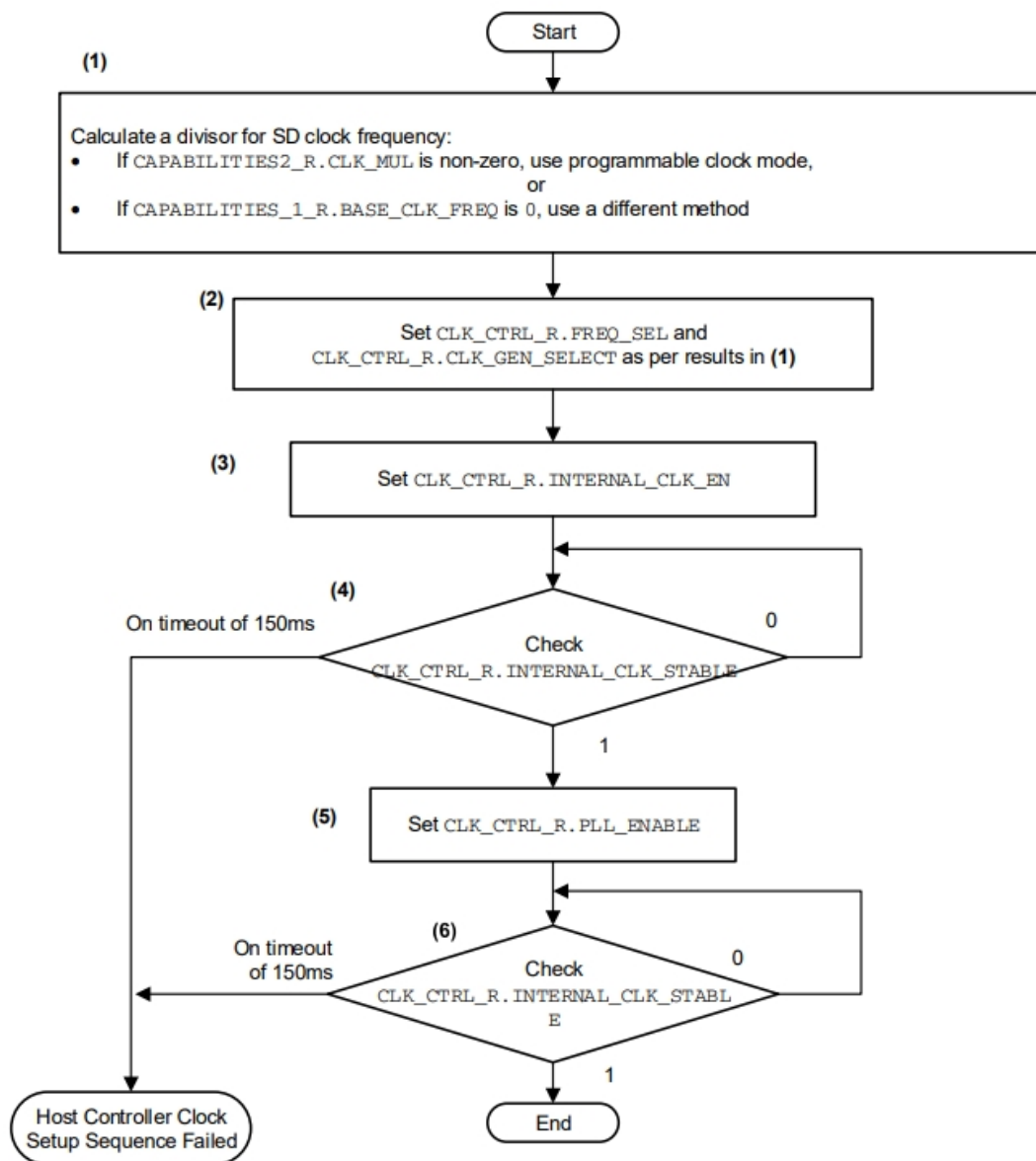
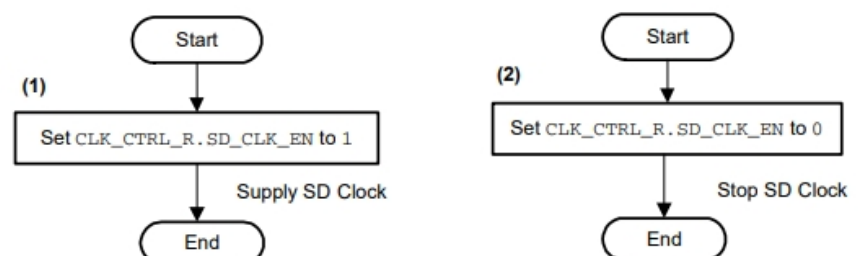


Figure 12-4-5 Host

Controller Clock Setup Sequence

12.4.2.5.2 Card Clock Supply and Stop Sequence

Figure 12-4-6 shows the flow chart for stopping the clock to card. The procedure outlined in Figure 4-5 applies to



both SD/eMMC and UHS-II cards.

Figure 12-4-6 Card Clock Supply and Stop Sequence

12.4.2.5.3 SD Clock Frequency Change Sequence

Figure 12-4-7 shows the sequence for changing SD/UHS-II clock frequency.

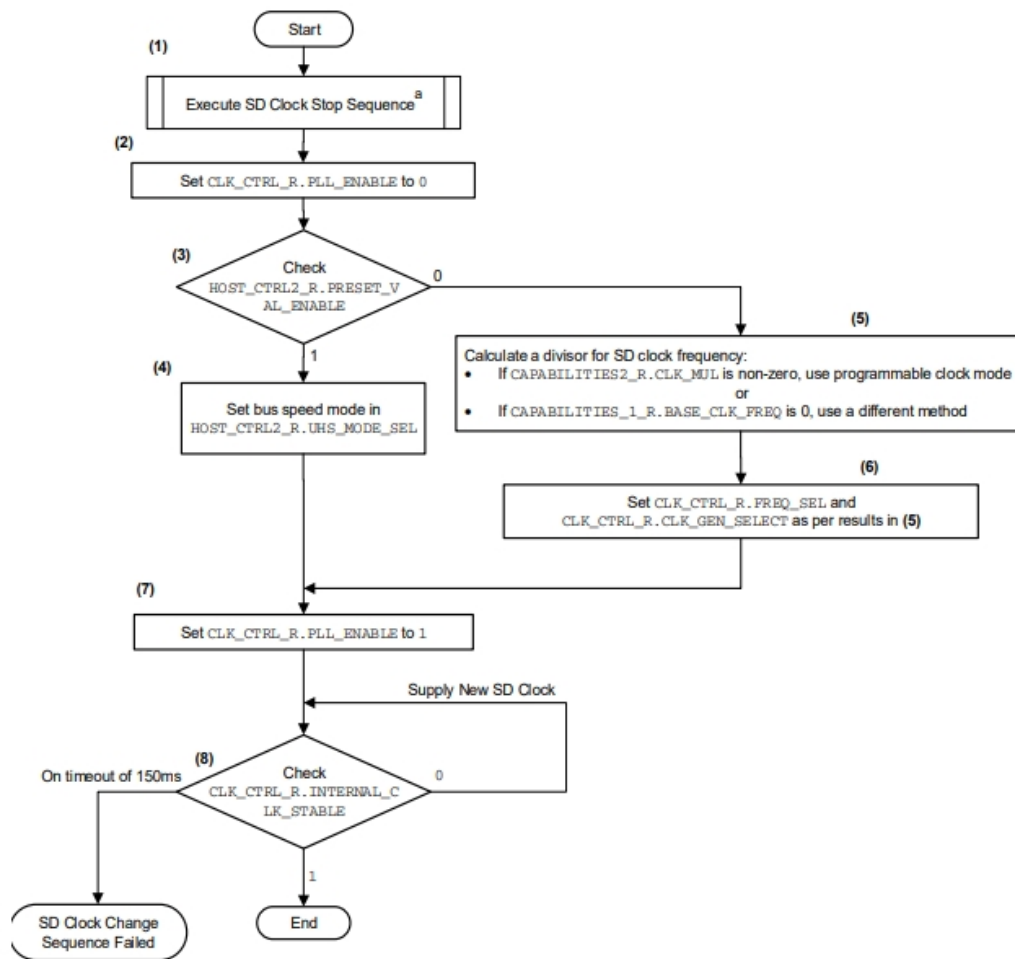


Figure 4-6 SD/UHS-II Clock

Frequency Change Sequence a:Refer “Card Clock Supply and Stop Sequence”

12.4.2.6 Card Interface Setup Sequence

12.4.2.6.1 SD/UHS-II Card Interface Detection

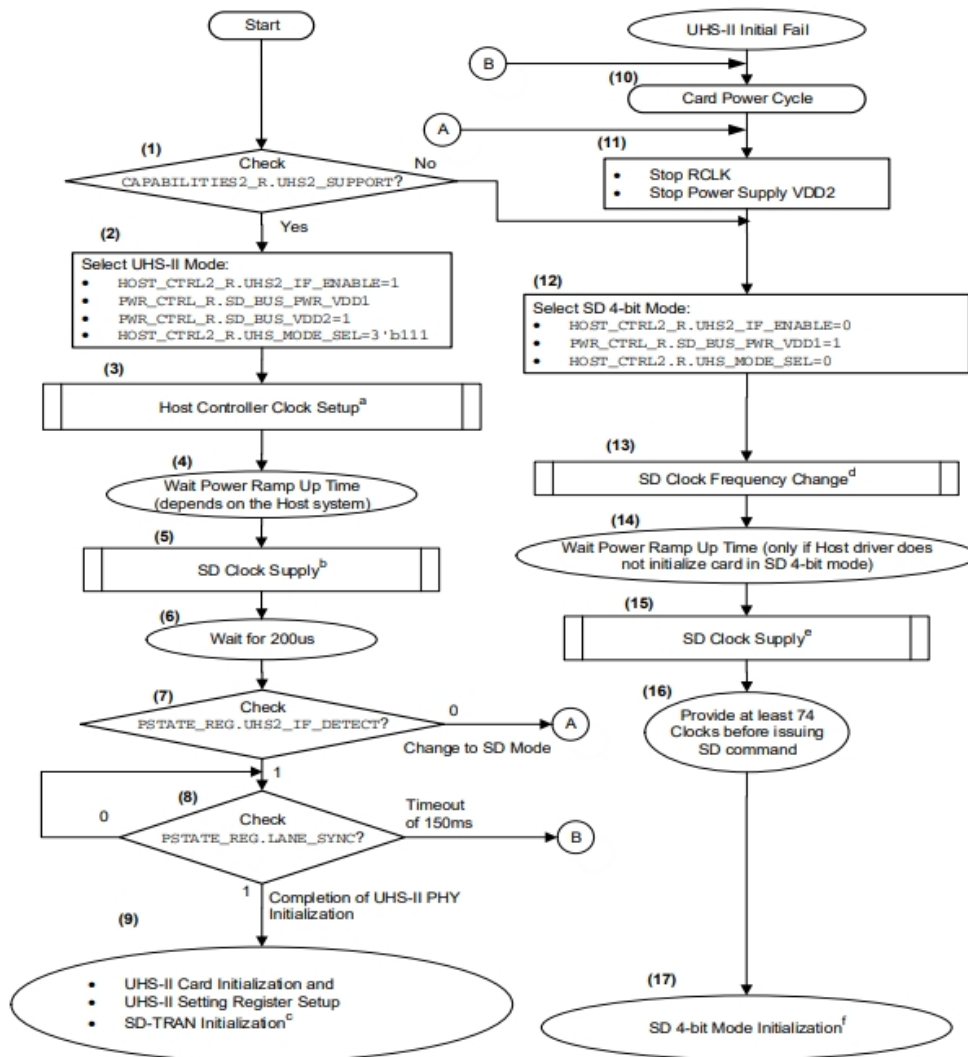


Figure 12-4-8 SD/UHS-II Card

Interface Detection Sequence

a Refer “Host Controller Clock Setup Sequence”.

b e Refer “Card Clock Supply and Stop Sequence”.

c f Refer “SD Card Initialization and Identification”.

d Refer “SD Changing Bus Speed Mode”.

Figure 12-4-9 shows timing of latency compensation implemented by the controller for reading UHS-II interface detection in the Present State register, after setting the SD Clock Enable in Step 5 in . The latency introduced between setting SD Clock Enable and STB.L on D0 line is compensated by delaying the read operation of the Present State register.

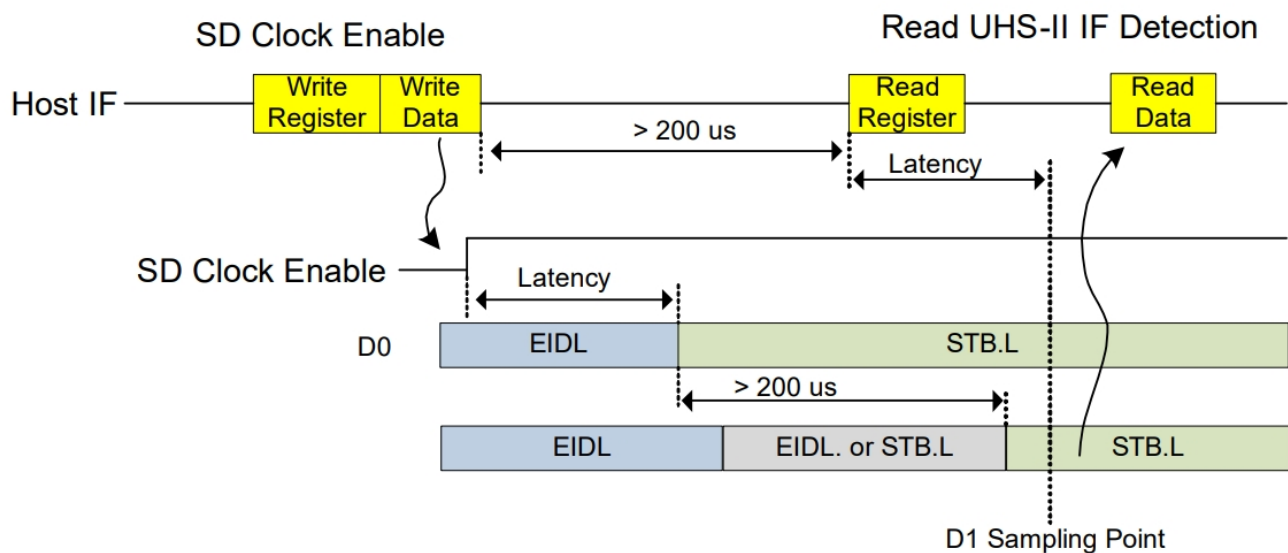


Figure 12-4-9 Latency Compensation for Reading UHS-II IF Detection

12.4.2.6.2 eMMC Card Interface Setup

Figure 12-4-10 shows the programming sequence to set up an eMMC device.

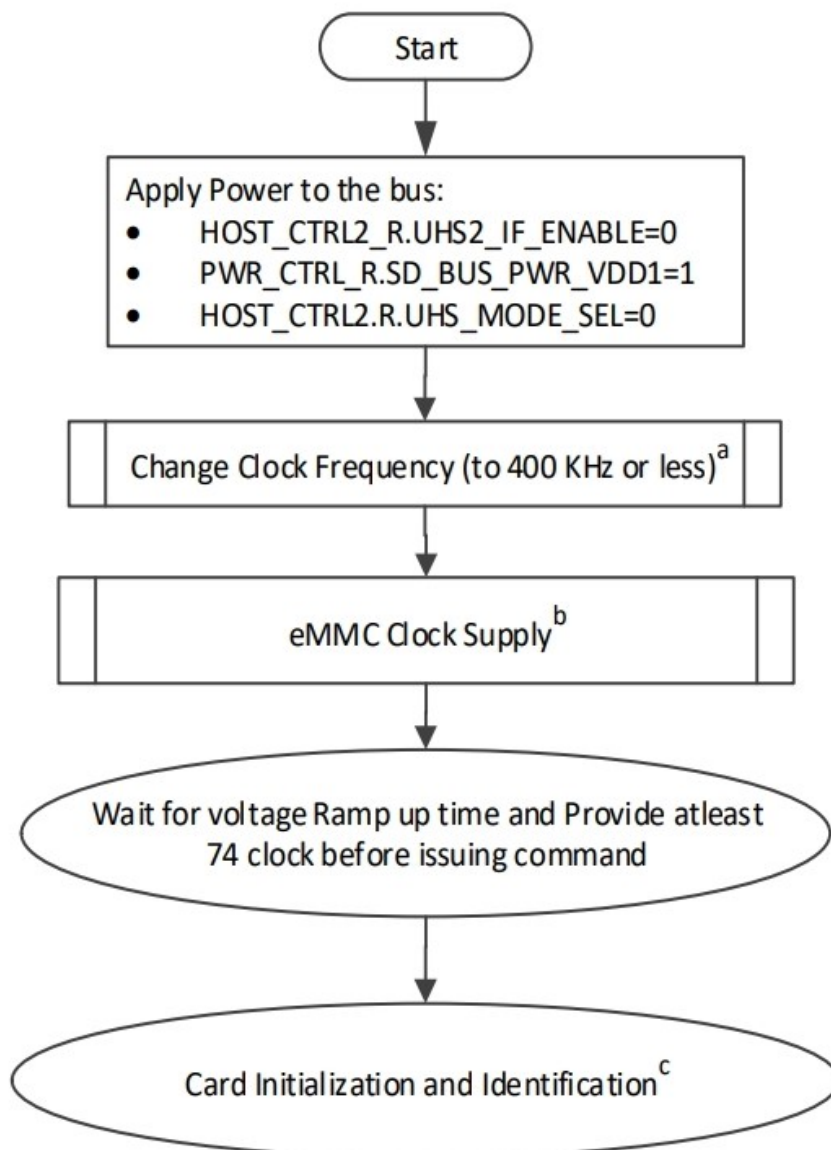


Figure 12-4-10 eMMC Card Setup

Sequence a Refer SD Changing Bus Speed Mode. b Refer Card Clock Supply and Stop Sequence. c Refer SD Card Initialization and Identification.

12.4.2.7 Timeout Setting

12.4.2.7.1 Timeout Setting for an SD/eMMC Bus

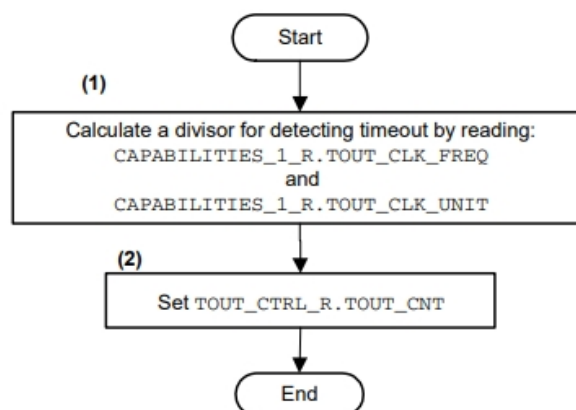


Figure 12-4-11 shows the timeout setting for an SD/eMMC bus.

Figure 12-4-11 SD/eMMC Bus Timeout Setting

12.4.2.7.2 Timeout Setting for a UHS-II Bus

Figure 12-4-12 shows the timeout setting on the UHS-II bus.

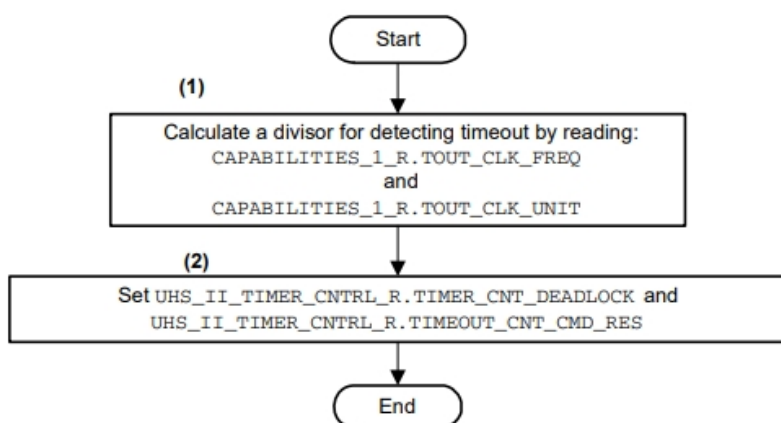


Figure 12-4-12 UHS-II Bus Timeout

Setting

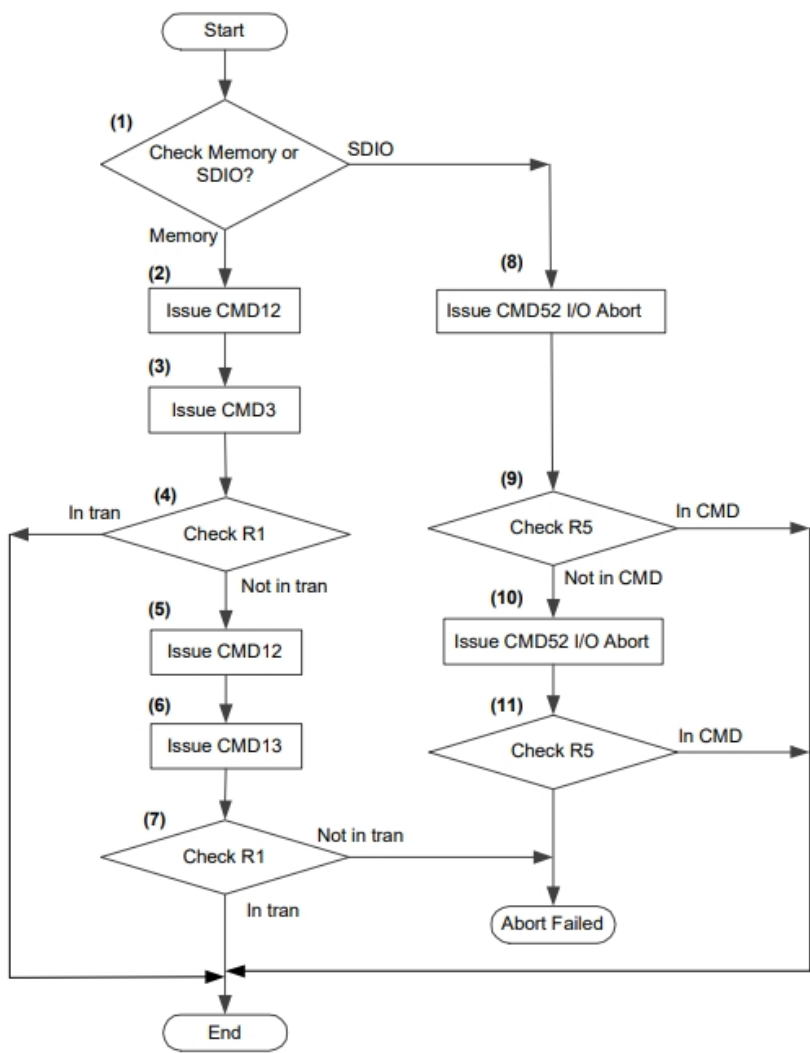
12.4.2.8 Abort Transaction

An abort command to the card is issued to abort an ongoing data transaction. To abort a transaction on an SD/UHS-II/eMMC card, CMD12 is issued, whereas to abort a transaction on an SDIO card, CMD52 is issued. In case of UHS-II, CMD12 needs to be encapsulated using UHS-II CCMD packet. Following are the instances when an abort command must be issued:

- To stop an infinite block transfer
- To stop transfer due to a request from an application
- To stop a transfer due to an error detection The sequence to issue an abort command is shown in “Abort Command Sequence” on page XXX. An abort command can be issued synchronously or asynchronously during data transfer. In asynchronous abort, abort command is issued at any time irrespective of the state of data transfer as a subcommand. Synchronous abort ensures that the data transfer is stopped (at block gap) before issuing an abort command. Both methods are explained in “Asynchronous Abort” on page XXX and “Synchronous Abort” on page XXX respectively.

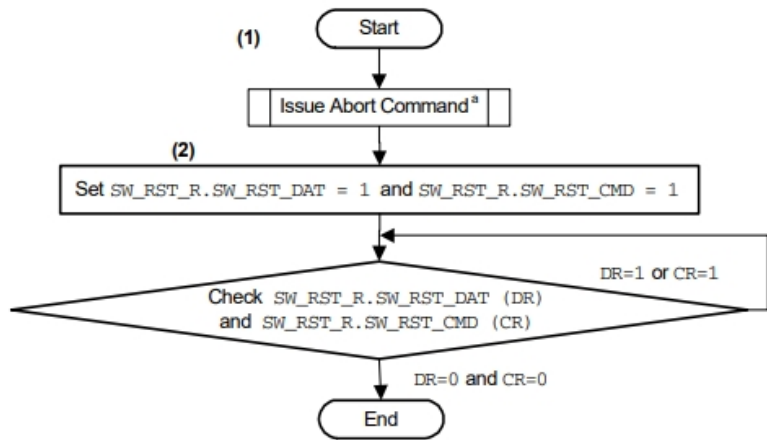
12.4.2.8.1 Abort Command Sequence

Figure 12-4-13 Abort Command Sequence



12.4.2.8.2 Asynchronous Abort

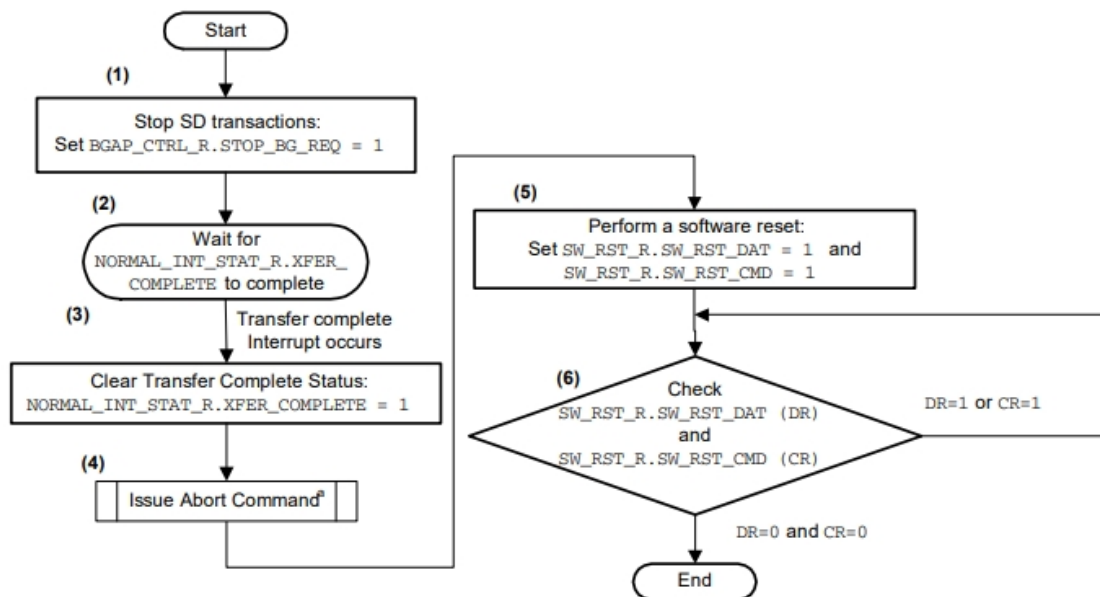
Figure 12-4-14 Asynchronous Abort Sequence



^a Refer Abort Command Sequence.

12.4.2.8.3 Synchronous Abort

Figure 12-4-15 Synchronous Abort Sequence



12.4.2.9 SD/SDIO Transaction Mode

12.4.2.9.1 SD Card Initialization and Identification

Figure 12-4-16 shows the sequence for SD card initialization and identification.

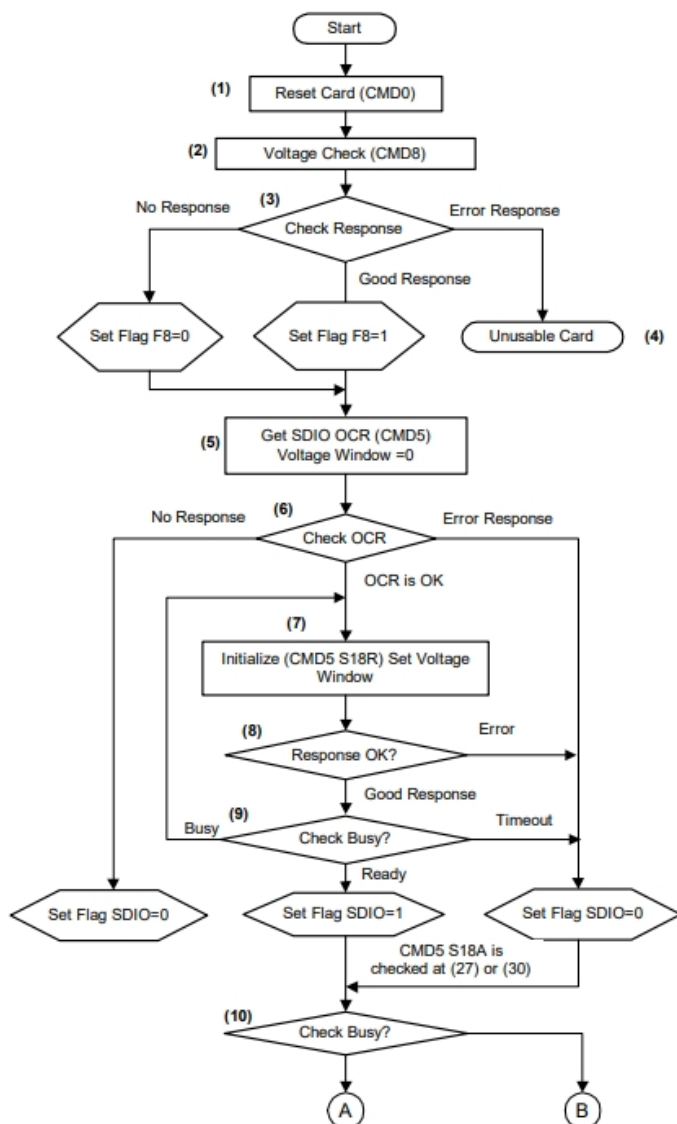
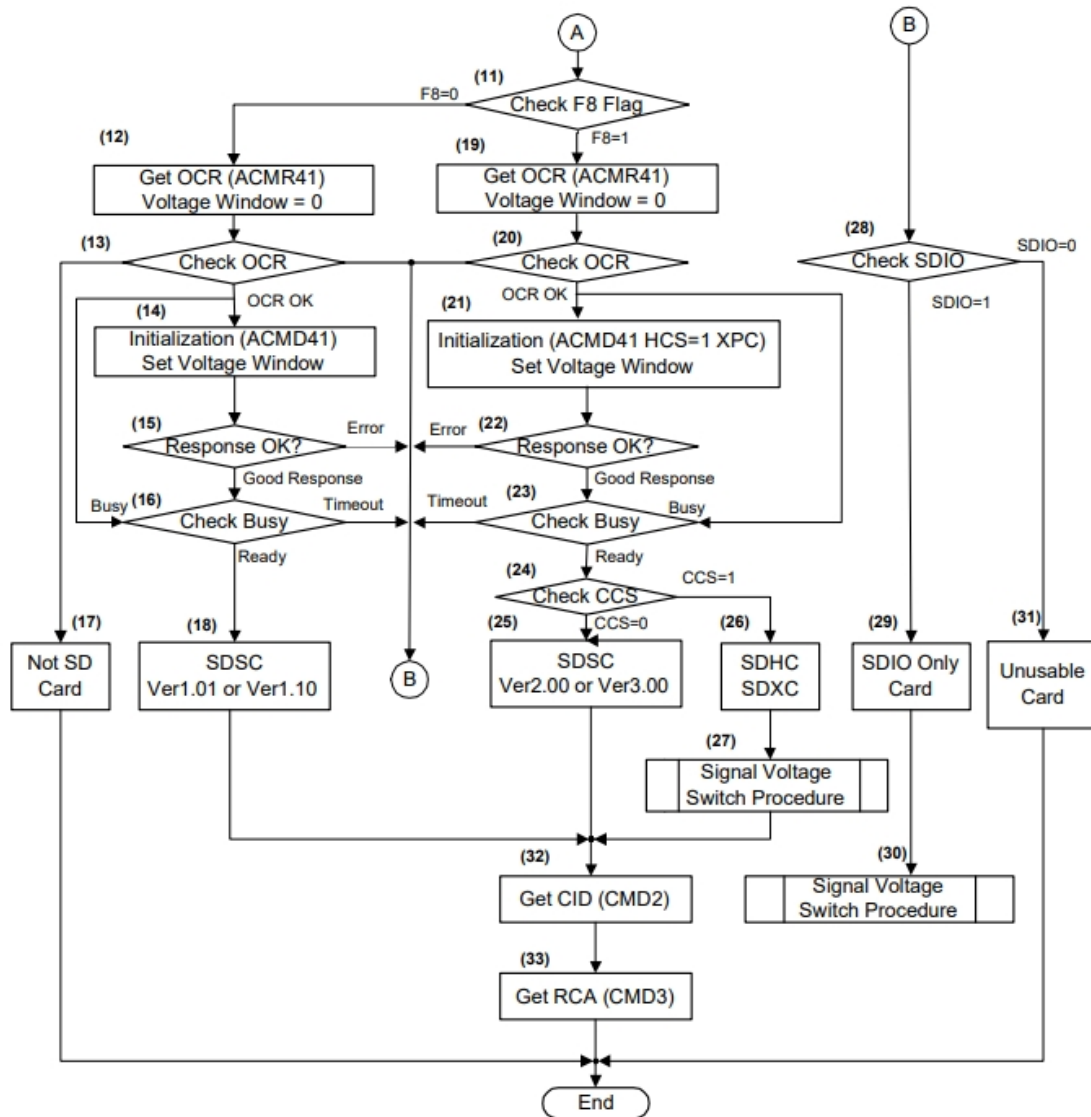


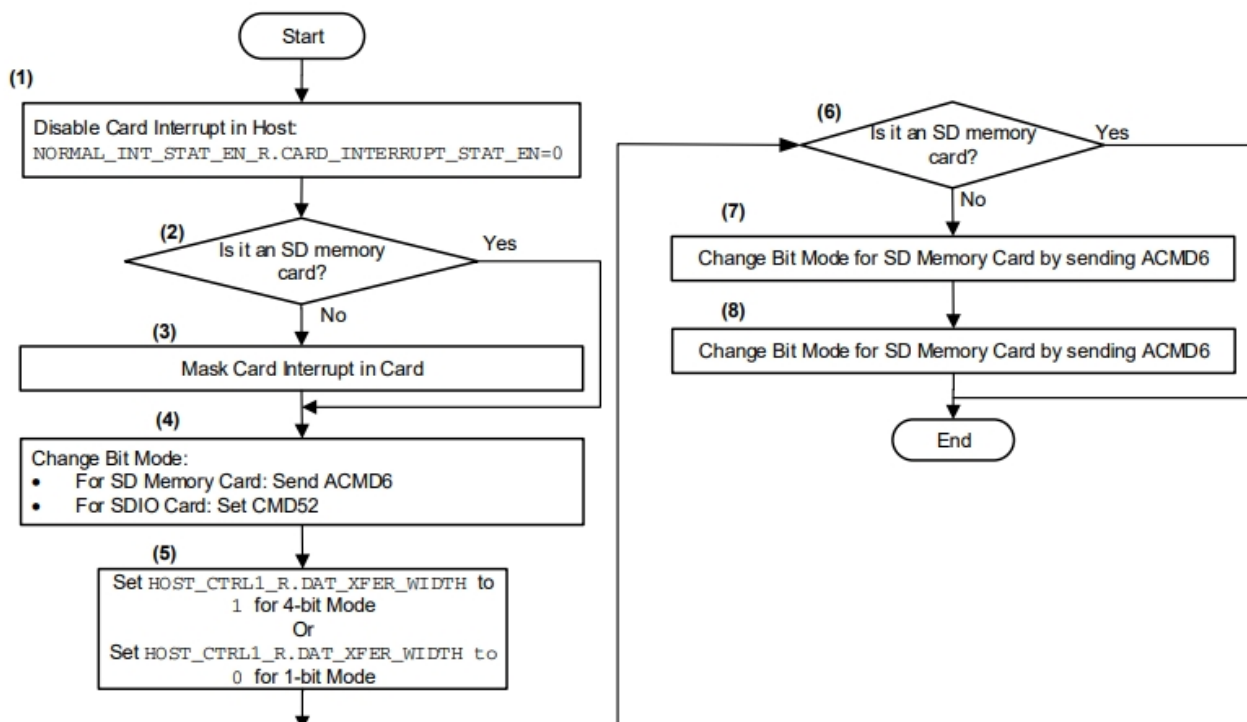
Figure 12-4-17 SD Card Initialization and



Identification

12.4.2.9.2 SD Changing Bus Width

Figure 12-4-18 shows the sequence for changing bit mode on SD Bus.

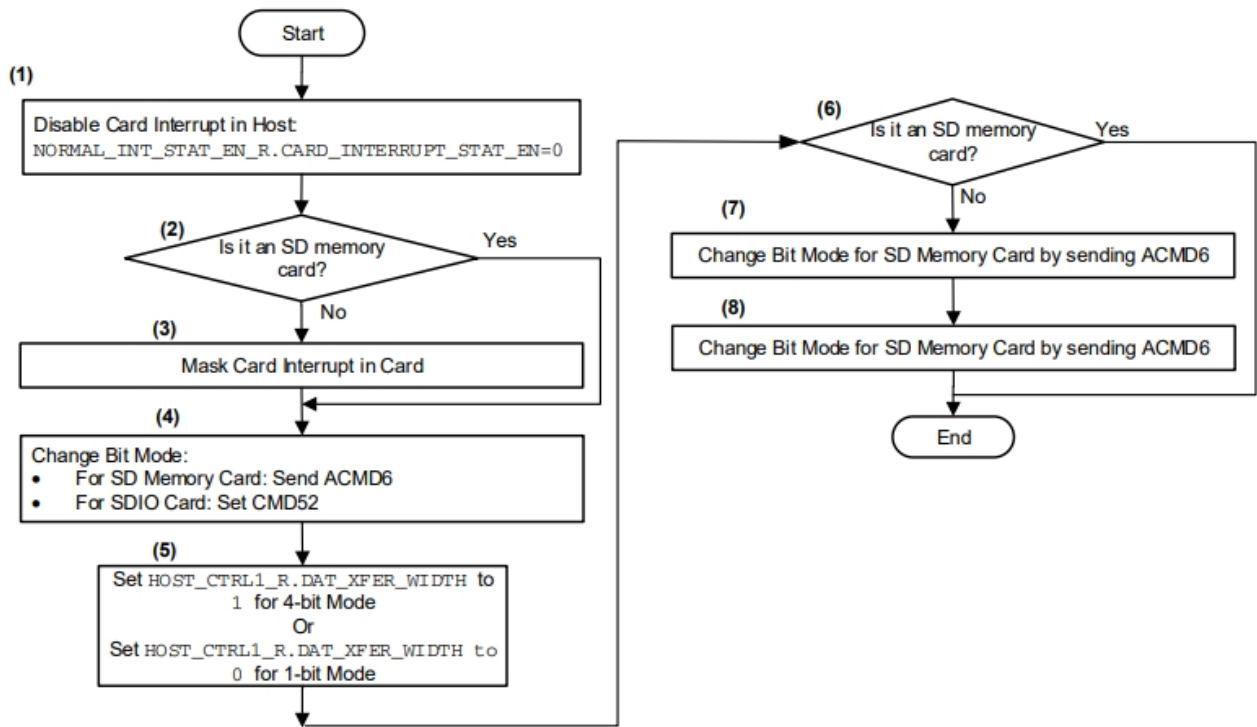


Figure

12-4-18 Change Bus Width Sequence

12.4.2.9.3 SD Bus Power Control

Figure 12-4-19 shows the sequence for controlling the SD Bus Power.



Figure

12-4-19 SD Bus Power Control Sequence

12.4.2.9.4 Issuing CMD without Data Transfer

Figure 12-4-20 shows the sequence on how to issue and complete an SD command. An SD command may or may not use a data line (DAT) (which is used for busy signaling). Figure 4-23 shows the procedure to issue commands for both

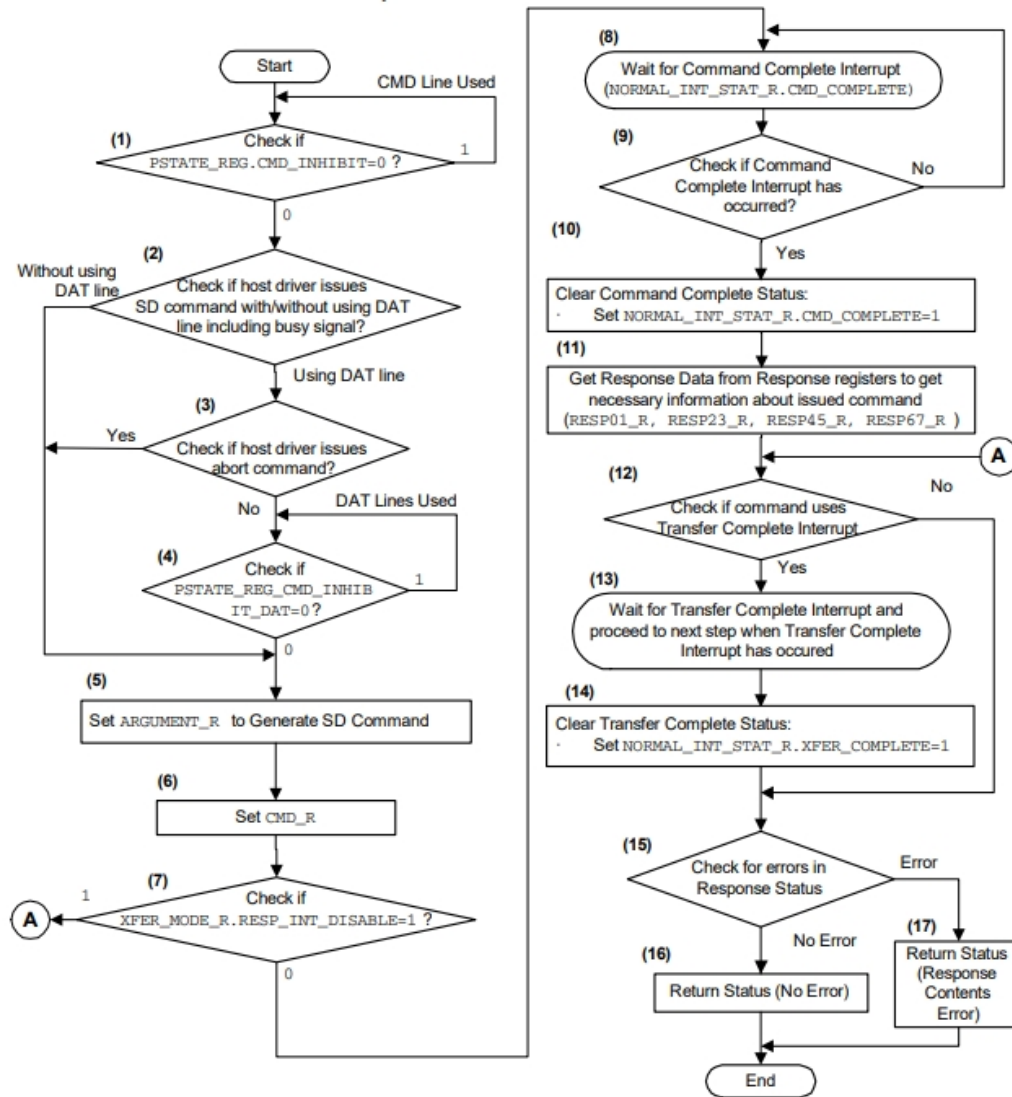


Figure 12-4-

scenarios.

20 SD Command Issue and Complete

12.4.2.9.5 Issuing CMD with Data Transfer

This section includes the sequence to issues a command with data transfer without using DMA and by using SDMA, ADMA, and ADMA3 modes. The ADMA3 ID descriptor, Command Descriptor, and ADMA2 descriptor are organized in a location in the host memory. The location address shall comply based on Table 4-3. Table 4-3 Host Memory Location

Addressing Mode	DWC_MSHC_MBIU_DW (bits)	ADMA3 ID address	Command Descriptor address	ADMA2 descriptor address
64-bit	64	Aligned to MBIU DW	Aligned to MBIU DW	Aligned to 4-byte boundary
64-bit	32	Aligned to MBIU DW	Aligned to MBIU DW	Aligned to MBIU DW
32-bit	64	Aligned to MBIU DW	Aligned to MBIU DW	Aligned to MBIU DW
32-bit	32	Aligned to MBIU DW	Aligned to MBIU DW	Aligned to MBIU DW

12.4.2.9.5.1 Issuing CMD with Data Transfer (Not Using DMA/PIO)

Figure 12-4-21 shows the transaction that does not use DMA for data transfer using the DAT line.

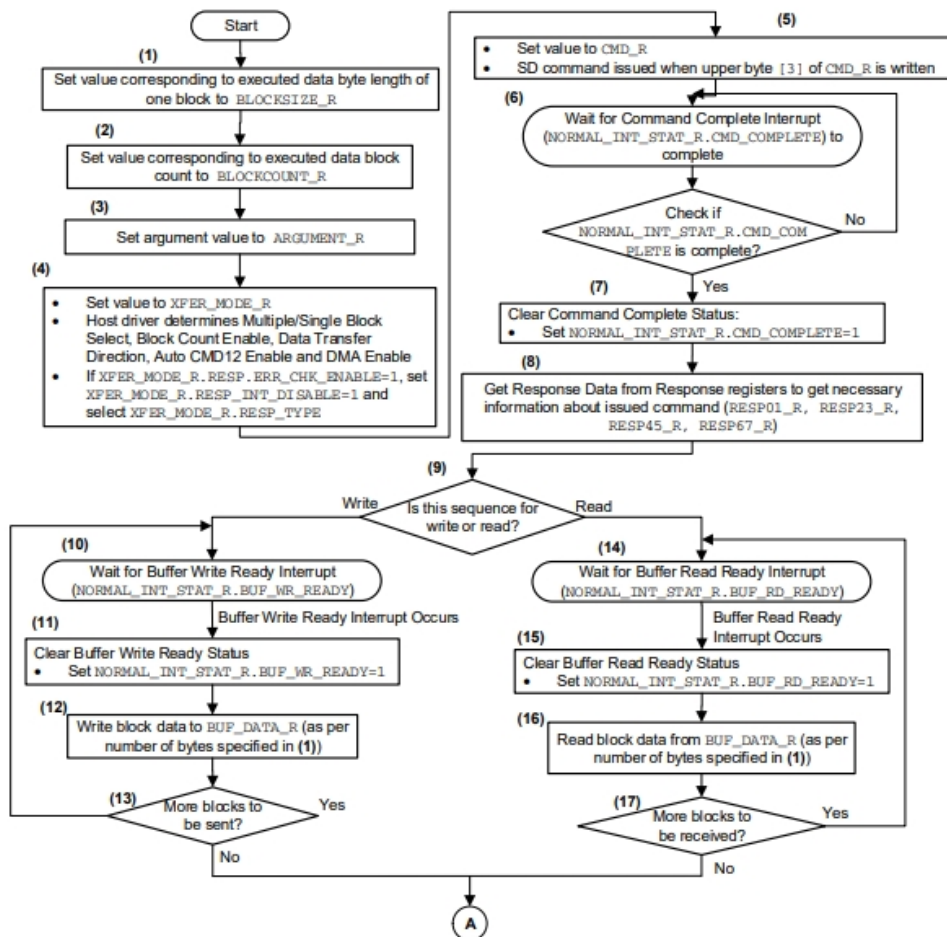


Figure 12-4-21 Transaction

Control with Data Transfer Using DAT Line (Not Using DMA)

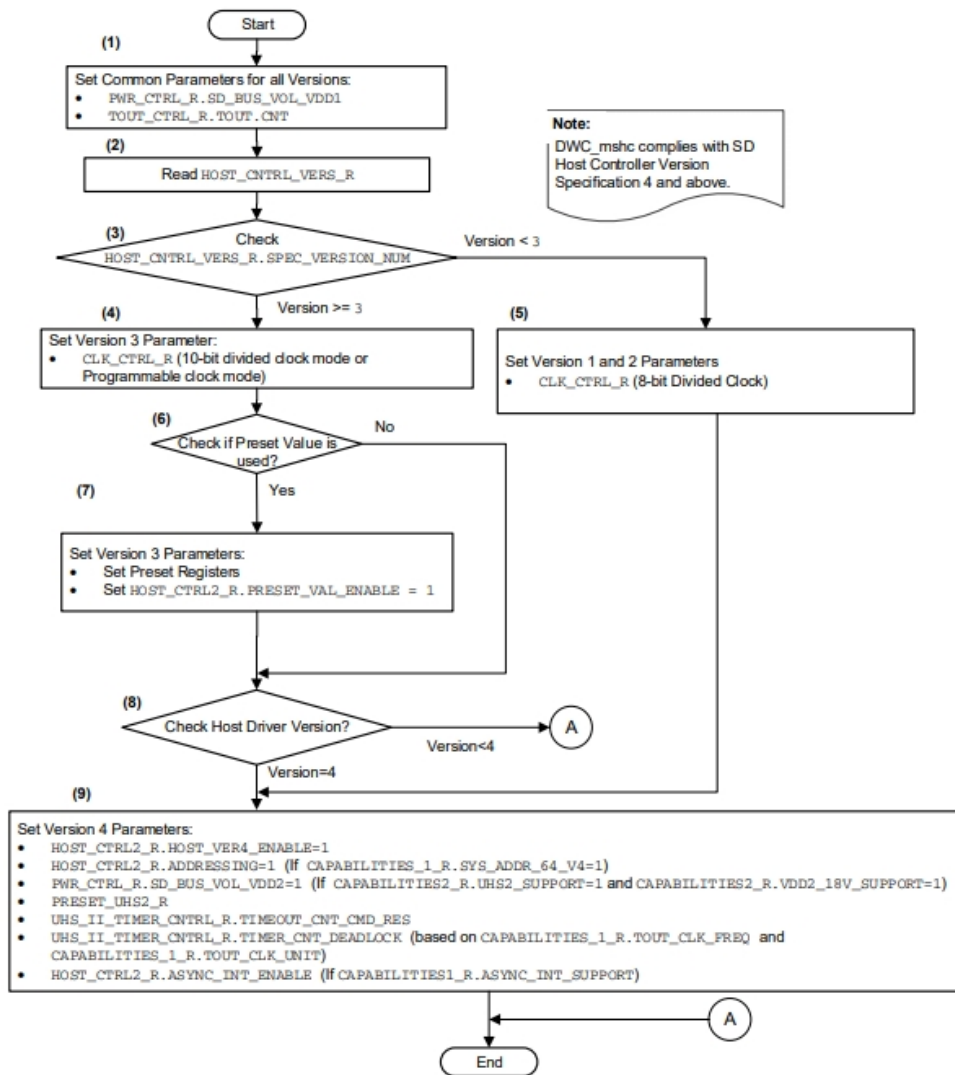


Figure 12-4-22 Transaction

Control with Data Transfer Using DAT Line (Not Using DMA)

12.4.2.9.5.2 Issuing CMD with Data Transfer (Using SDMA)

Figure 12-4-23 shows the transaction that uses SDMA for data transfer using the DAT line.

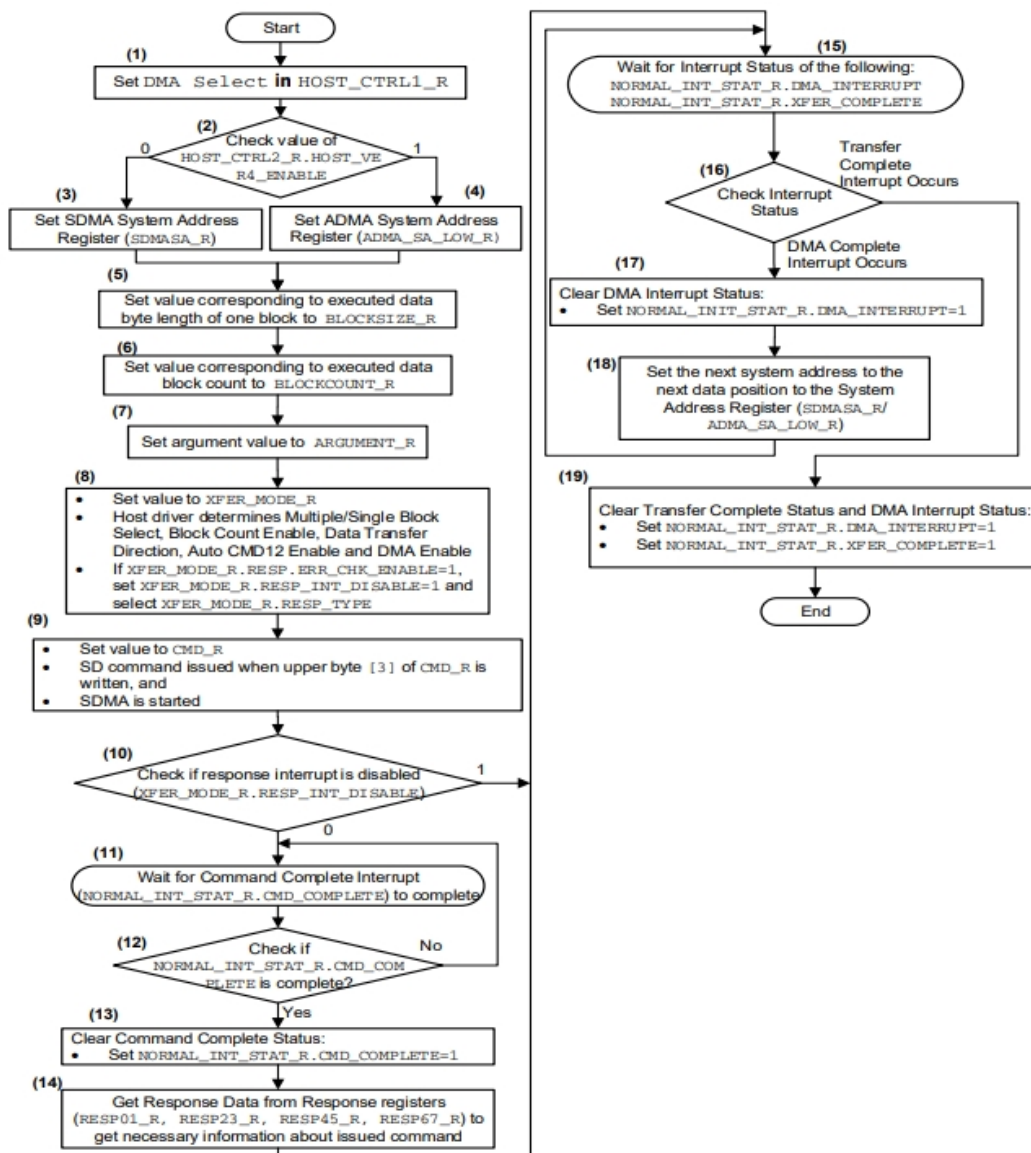


Figure 12-4-23

Transaction Control with Data Transfer using SDMA Note:ERROR_INT_STAT_R .ADMA_ERR bit is set for all DMA (SDMA, ADMA2 and ADMA3) transfers for AXI error response. (In case of 64-bit addressing, program the ADMA_SA_HIGH_R register tool.)

12.4.2.9.5.3 Issuing CMD with Data Transfer (Using ADMA2)

Figure 12-4-24 and Figure 4-28 show the transaction that uses ADMA2 for data transfer using the DAT line.

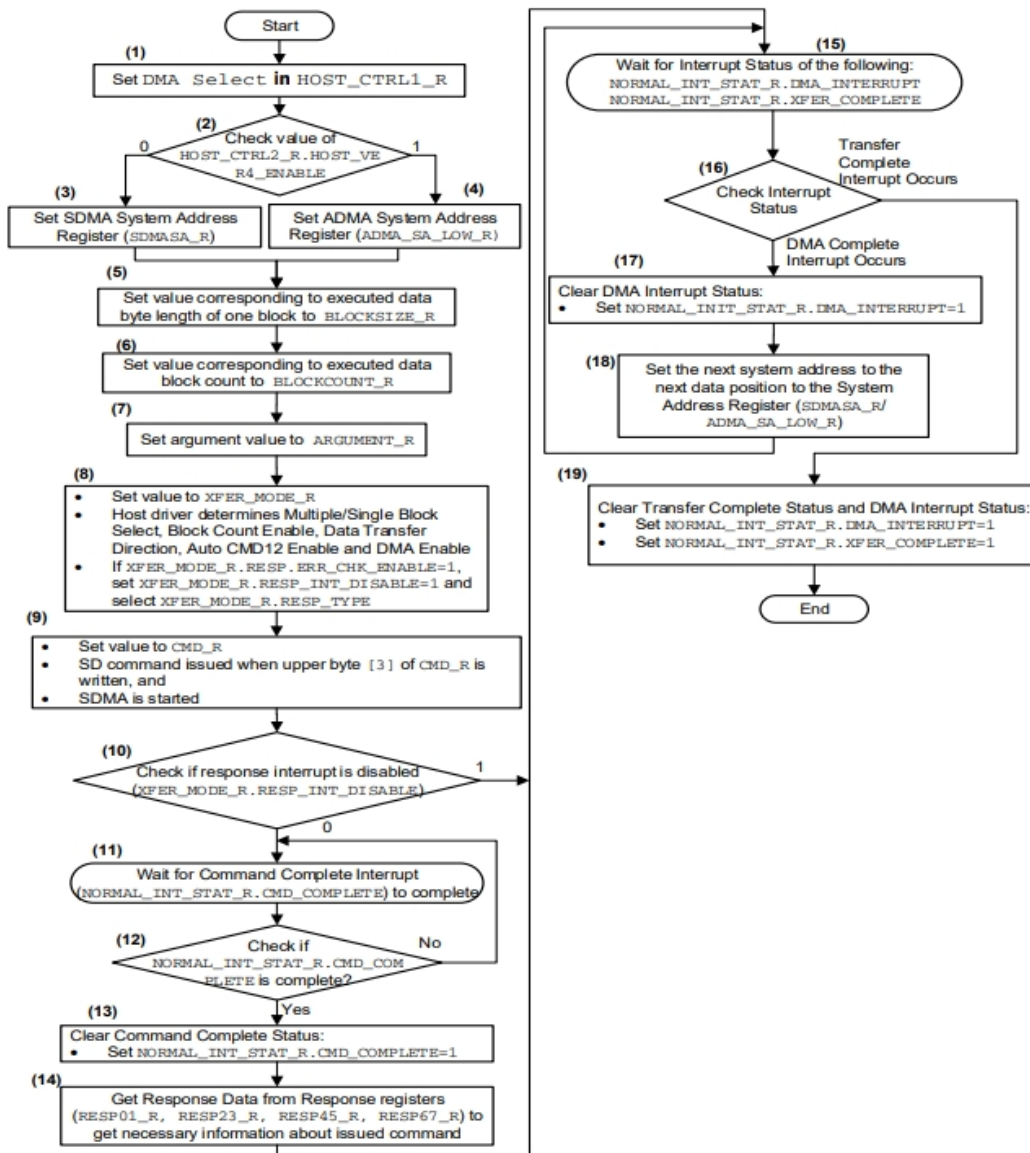


Figure 12-4-24

Transaction Control with Data Transfer Using DAT Line (Using ADMA2)

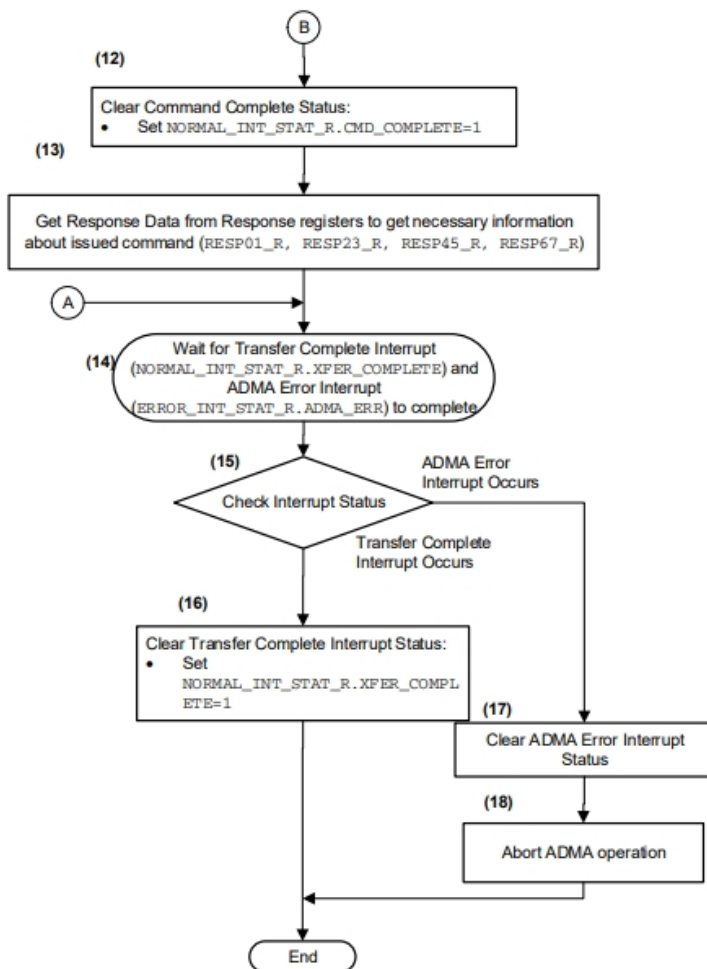


Figure 12-4-25 Transaction Control with Data

Transfer Using DAT Line (Using ADMA) (Contd) During Card Read/Write DMA operations, it is recommended that the very last transfer descriptor also has the 'End' bit set for marginal performance improvement. End bit should always be set on a descriptor of type 'TRAN'. Possible descriptor types are provided in Table 4-4 for quick reference. Table 4-4 Descriptor Types

Act2	Act1	Act0	Symbol	Comment	ADMA2 descriptor address
0	0	0	Nop	No Operation	Do not execute currentline and go to the next line.
0	1	0	rsv	Reserved	Same as Nop. Do not execure current line and go to next line.
1	0	0	Tran	Transfer Data	Transfer data of one descriptor line
1	1	0	Link	Link Descriptor	Link to another descriptor

12.4.2.9.5.4 Issuing CMD with Data Transfer (Using ADMA3)

Figure 12-4-26 shows the transaction that uses ADMA3 for data transfer using the DAT line.

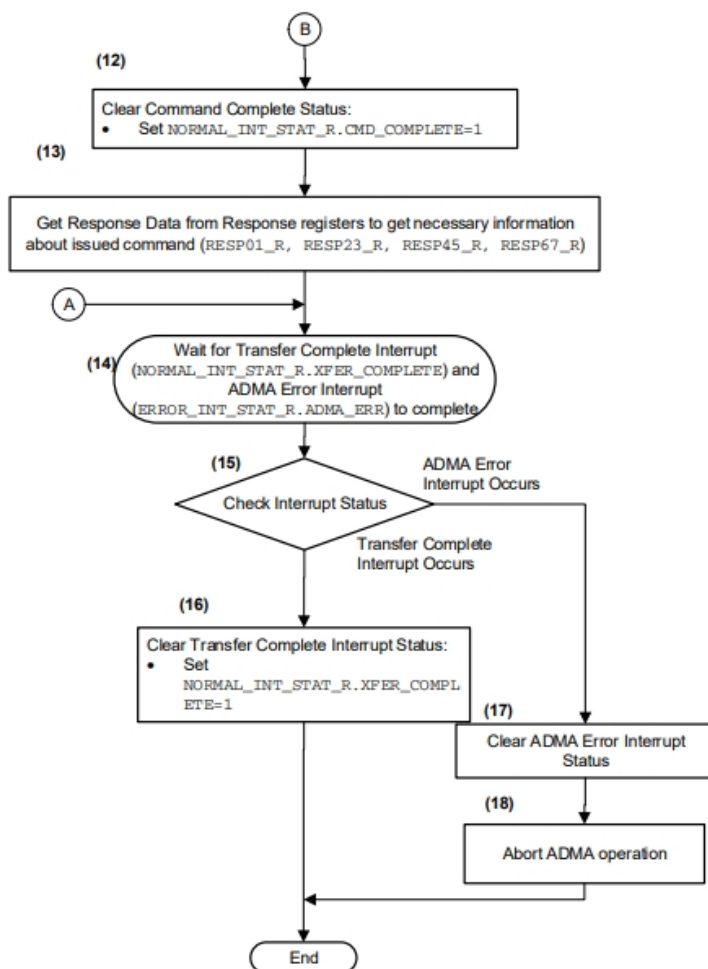


Figure 12-4-26 Transaction Control with Data

Transfer Using DAT Line (Using ADMA3) Following are the descriptor requirements for an ADMA3 CMD:

- Setting infinite data transfer (Block Count Enable = 0) is not allowed except for single block transfer (Multi/Single Block Select=0)
- As ADMA3 only supports 32-bit Block Count mode, Block Count Enable is set to 1, 16-bit Block Count is set to 0000h. Setting Stop Count (32-bit Block Count =0) is not allowed except for single block transfer (Multi/Single Block Select = 0).
- Setting no data transfer (Block Size=0) is not allowed, except for no data transfer command in the Command Descriptor (CDEnd = 1).
- In case of memory data transfer command, "Auto CMD Auto Select" of the Auto CMD Enable in the Transfer Mode register is used so that Command Descriptor is independent to whether card support CMD23. For another command (including SDIO), "Auto Command Disabled" of auto CMD Enable is used. During Card Read/Write DMA operations, it is recommended that the very last transfer descriptor also has the 'End' bit set for marginal performance improvement. End bit should always be set on a descriptor of type 'TRAN'. Possible descriptor types are provided in Table 4-4 for quick reference. Table 4-5 Descriptor Types

Act2	Act1	Act0	Symbol	Comment	ADMA2 descriptor address
0	0	0	Nop	No Operation	Do not execute current line and go to the next line.
0	1	0	rsv	Reserved	Same as Nop. Do not execute current line and go to next line.
1	0	0	Tran	Transfer Data	Transfer data of one descriptor line
1	1	0	Link	Link Descriptor	Link to another descriptor

12.4.2.9.6 SD Changing Bus Speed Mode

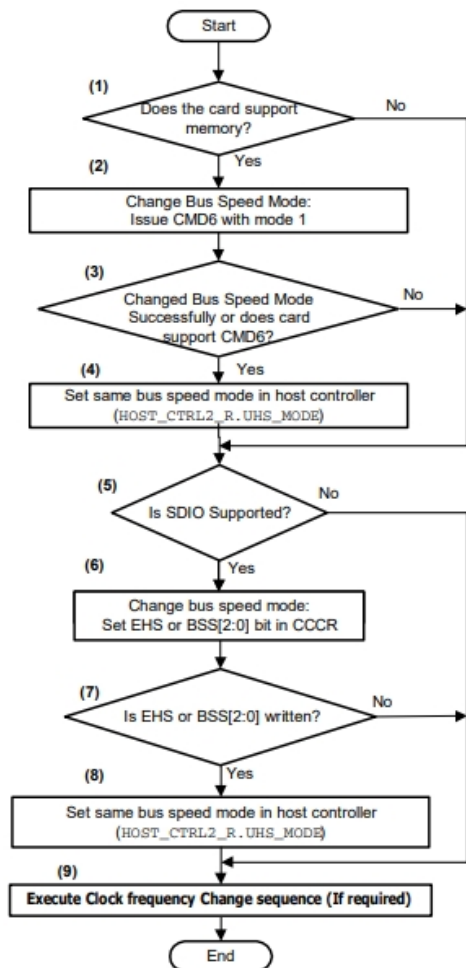


Figure 12-4-27 SD Changing Bus Speed Mode

12.4.2.9.7 SDIO Card Interrupt

This section discusses the sequence for including an SDIO card interrupt.

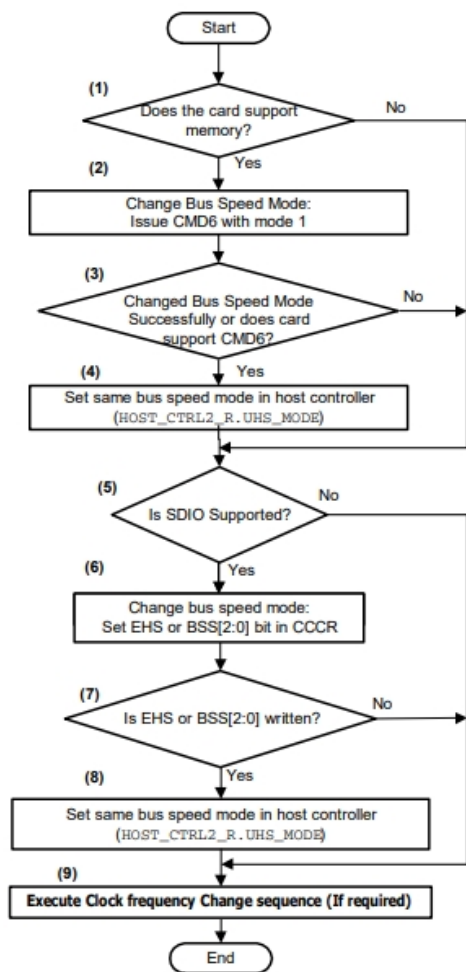


Figure 12-4-28 SDIO Card Interrupt Sequence

12.4.2.10 UHS-II Transaction Mode

This section discusses UHS-II programming sequences for setting up the host controller, detecting the card interface, initializing an UHS-II card, setting the register setup, issuing an UHS-II CCMD and DCMD packet, entering a dormant or hibernate mode, issuing an SD-TRAN reset, and issuing a complete host reset.

12.4.2.10.1 UHS-II Card Initialization

Figure 12-4-29 shows the programming sequence for initializing an UHS-II card.

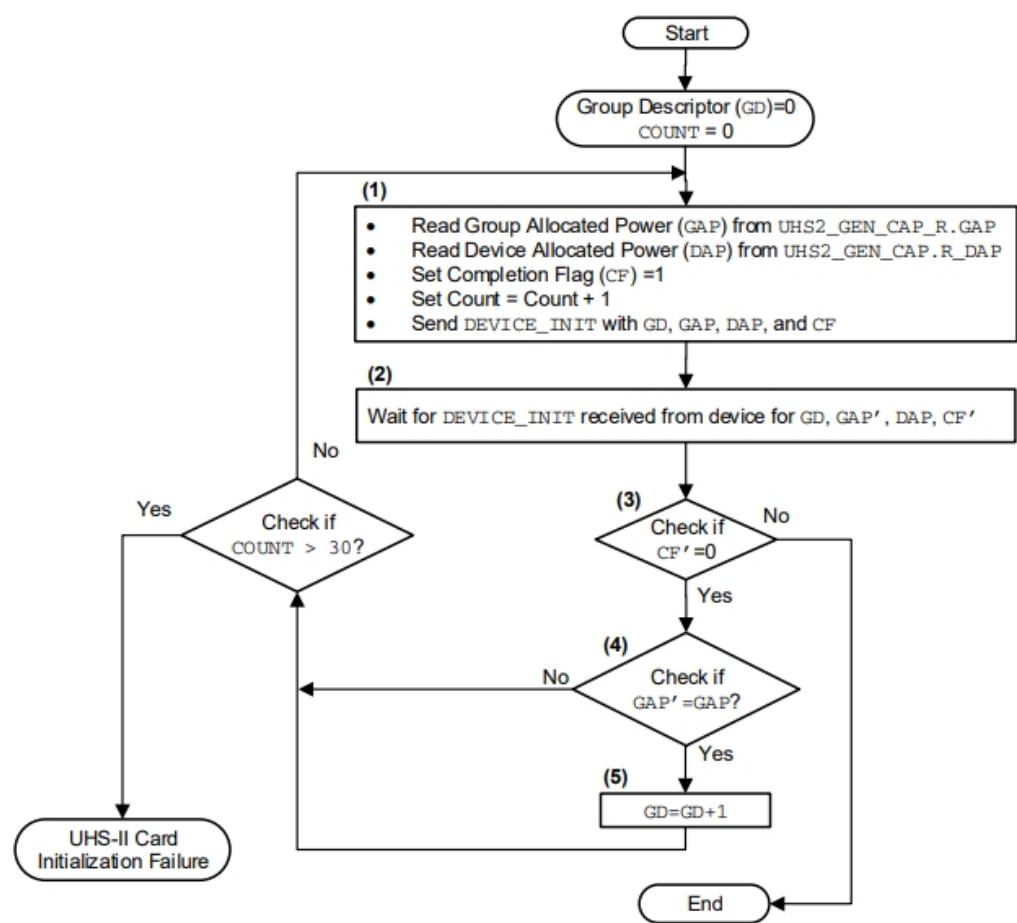


Figure 12-4-29 UHS-II Card

Initialization

12.4.2.10.2 UHS-II Settings Registers' Setup

Figure 12-4-30 shows the sequence for setting up the UHS-II register.

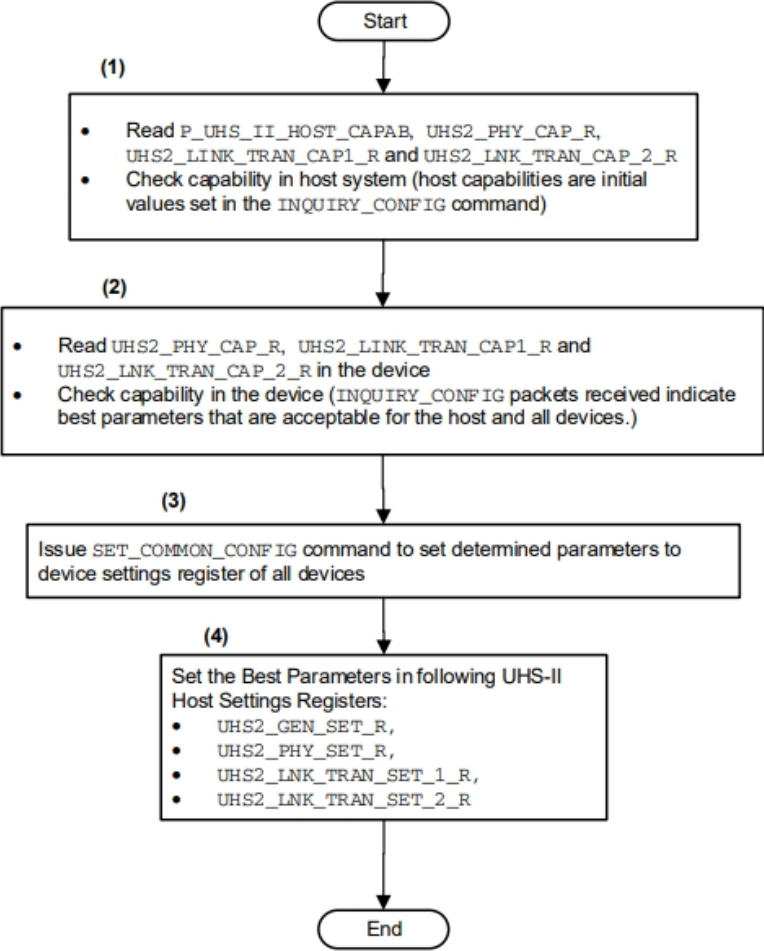


Figure 12-4-30 UHS-II Settings Register

Setup

12.4.2.10.3 UHS-II CCMD Packet Issuing

Figure 12-4-31 shows the sequence for issuing a UHS-II CCMD packet.

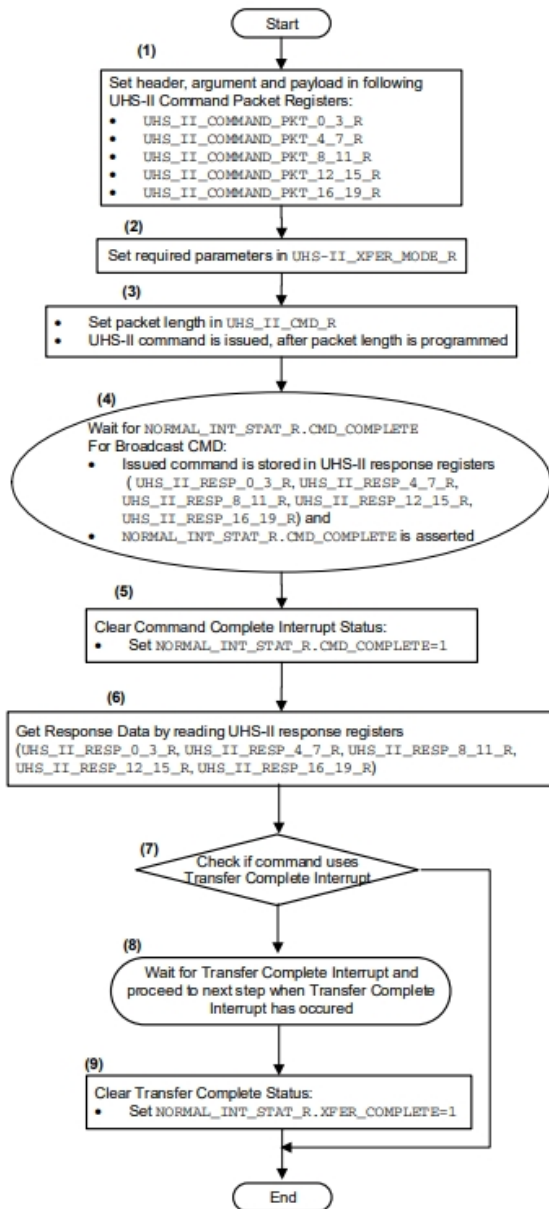


Figure 12-4-31 UHS-II CCMD Packet Issuing

12.4.2.10.4 UHS-II DCMD Packet Issuing

Figure 12-4-32 shows the sequence for issuing a UHS-II DCMD packet.

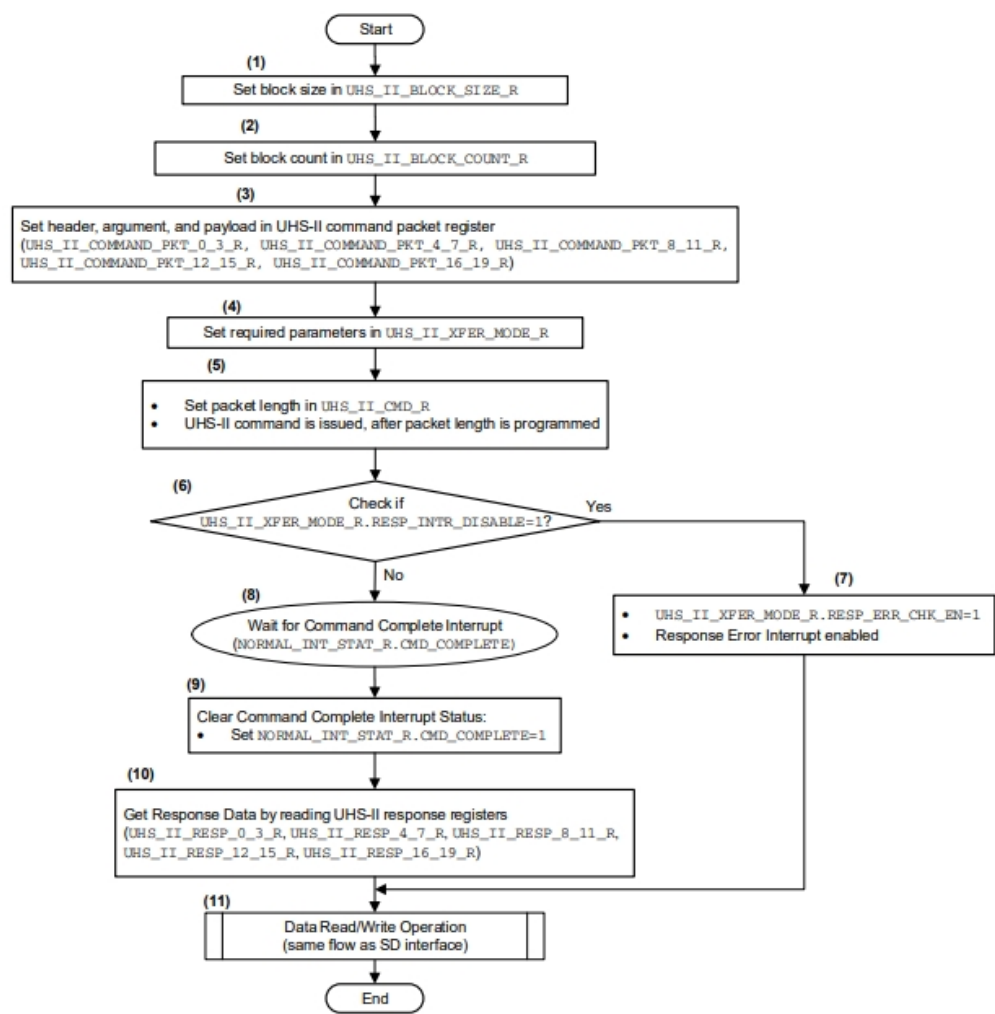


Figure 12-4-32 UHS-II DCMD

Packet Issuing

12.4.2.10.5 Entering Dormant or Hibernate Mode

Figure 12-4-33 shows the sequence for entering a dormant or hibernate mode.

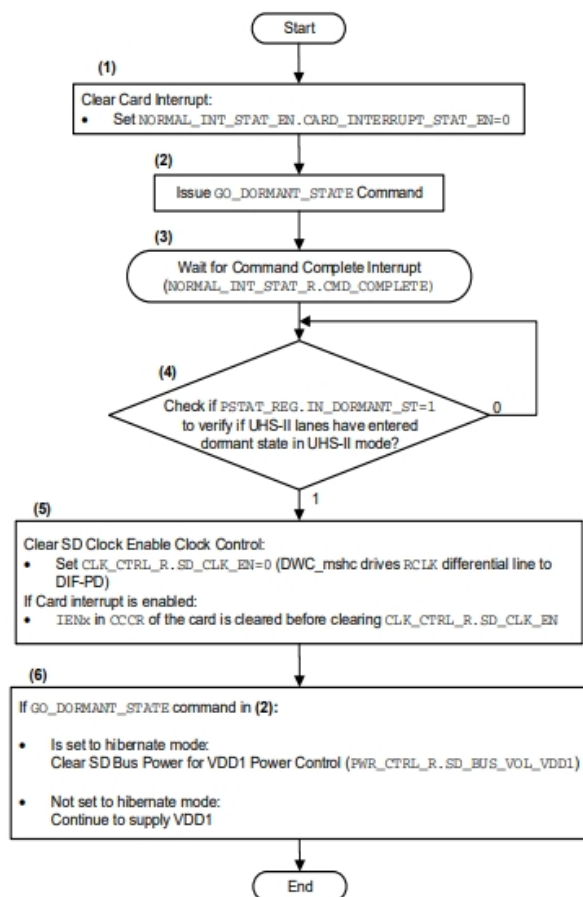


Figure 12-4-33 Entering Dormant or Hibernate Mode

12.4.2.10.6 SD-TRAN Reset Issuing

Figure 12-4-34 shows the sequence for issuing an SD-TRAN reset.

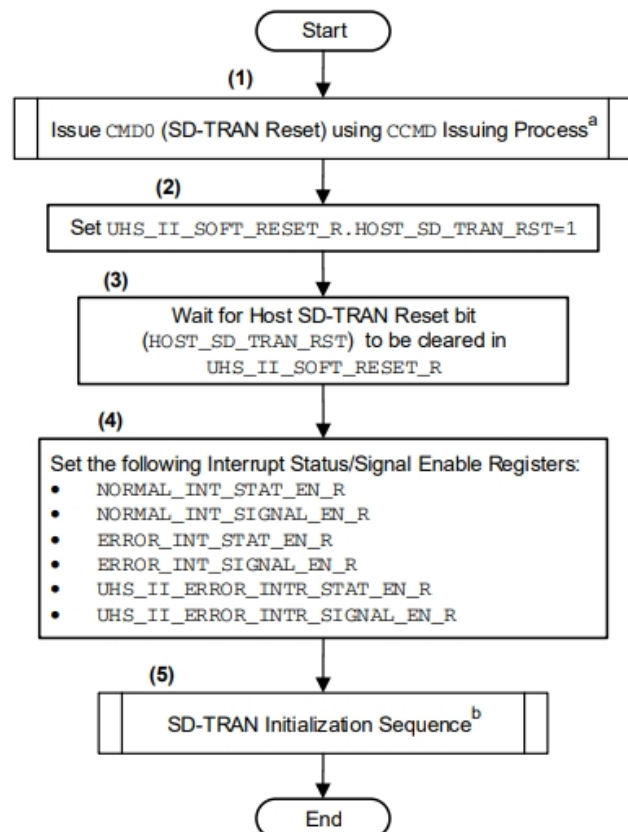


Figure 12-4-34 SD-TRAN Reset Issuing

12.4.2.10.7 Host Full Reset Issuing

Figure 12-4-34 shows the sequence for issuing a host full reset.

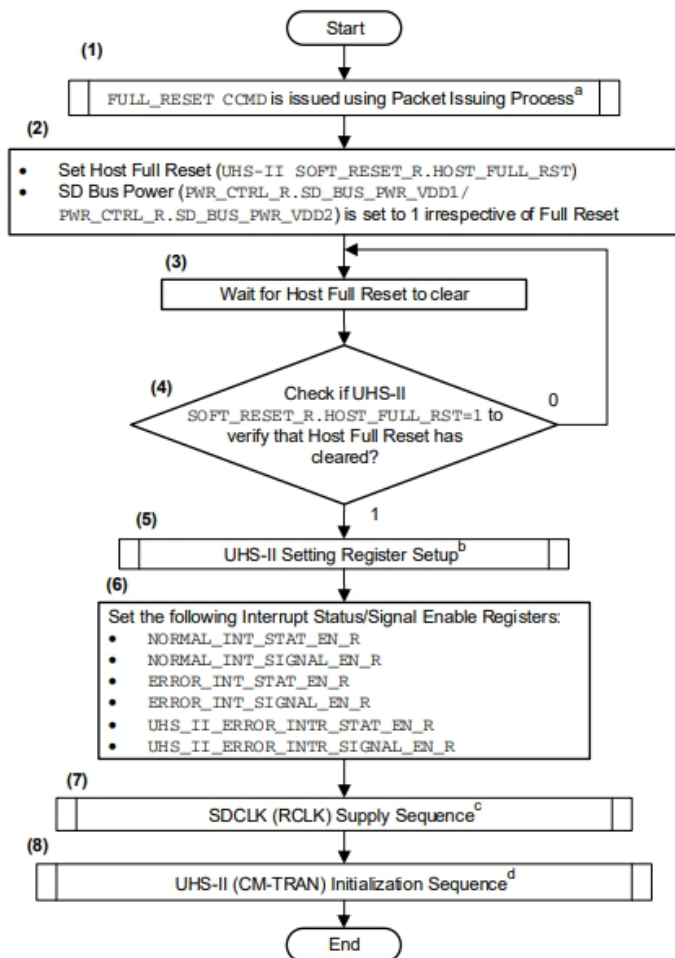


Figure 12-4-34 Host Full Reset Issuing

Note: UHS_II_CMD_R.UHS_II_CMD_TYPE should be set to 0x3 while issuing UHS-II FULL_RESET CCMD.

12.4.2.11 eMMC Transaction Mode

This section discusses the programming sequence for setting the host controller for an eMMC interface, the power sequence, initializing and identify the card, issuing a command with/without data transfer using or not using a DAT line, switching to various speed modes and to change the data bus width. This section also discusses how to program, initiate, and abort different boot speed modes.

12.4.2.11.1 Initializing and Identifying an eMMC Device

Figure 12-4-35 shows the programming sequence to initialize and identify an eMMC device.

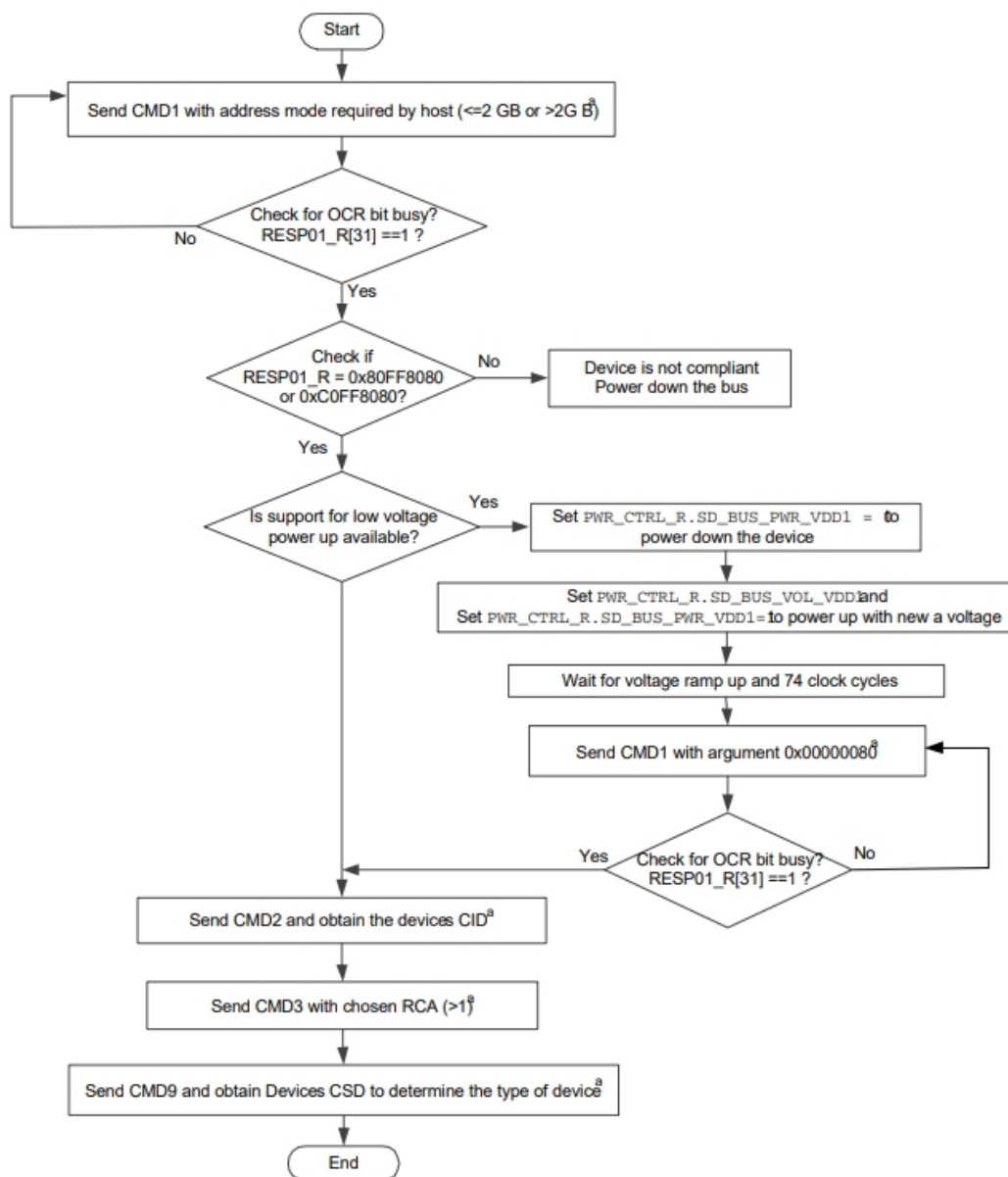


Figure 12-4-35 Card

Initialization and Identification Programming Sequence

12.4.2.11.2 Issue CMD without Data Transfer for an eMMC Device

The programming sequence for issuing a command without data transfer for an eMMC device is similar to the sequence mentioned in “Issuing CMD without Data Transfer” .

12.4.2.11.3 Issue CMD with Data Transfer for an eMMC Device

The programming sequence for issuing a command with data transfer for an eMMC device is similar to the programming sequence mentioned in “Issuing CMD with Data Transfer”. Data transfer can be done with or without using DMA. The supported DMA modes are SDMA, ADMA2, and ADMA3. Note:An application must ensure that the BOOT_ACK_ENABLE bit in BOOT_CTRL_R register is cleared, if set due to previous boot CMD, before initiating any data transfer CMD.

12.4.2.11.4 Switch to Various Speed Modes in an eMMC Device

Figure 12-4-36 shows the programming sequence to switch to various speed modes in an eMMC device. In Figure 4-41, note that CMD6 is effective only during the transfer state.

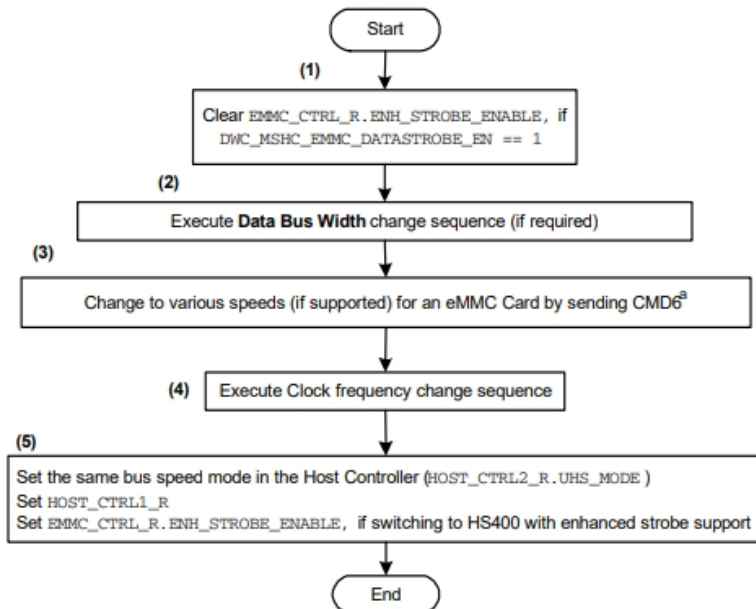


Figure 12-4-36 Programming Sequence to

Switch to Various Speed Modes in an eMMC Device

12.4.2.11.5 Changing the Data Bus Width for an eMMC Device

Figure 12-4-37 shows the programming sequence to change the data bus width for an eMMC device. In Figure 4-42, note that CMD6 is effective only during transfer state.

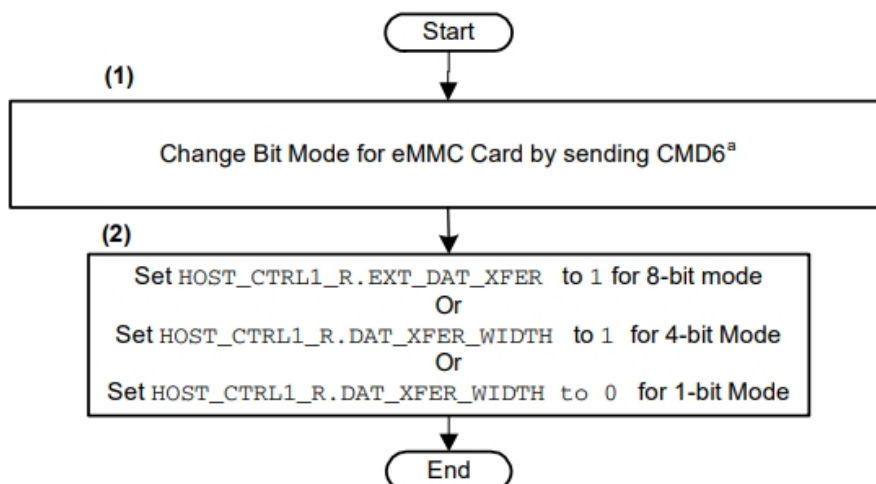


Figure 12-4-37 Programming

Sequence to Change Data Bus Width for an eMMC Device

12.4.2.11.6 Tuning Sequence for an eMMC Device

The tuning sequence for an eMMC device is similar to the programming sequence shown in Figure 4-10 except that for an eMMC device CMD21 is sent for tuning instead of CMD19 as mentioned in Figure 4-10. For eMMC, CMD21 shall be sent for tuning instead of CMD19 as defined in flow diagram

12.4.2.11.7 Command Queueing

12.4.2.11.7.1 Command Queueing Engine Initialization

Figure 12-4-38 shows the programming sequence to initialize the command queuing engine.

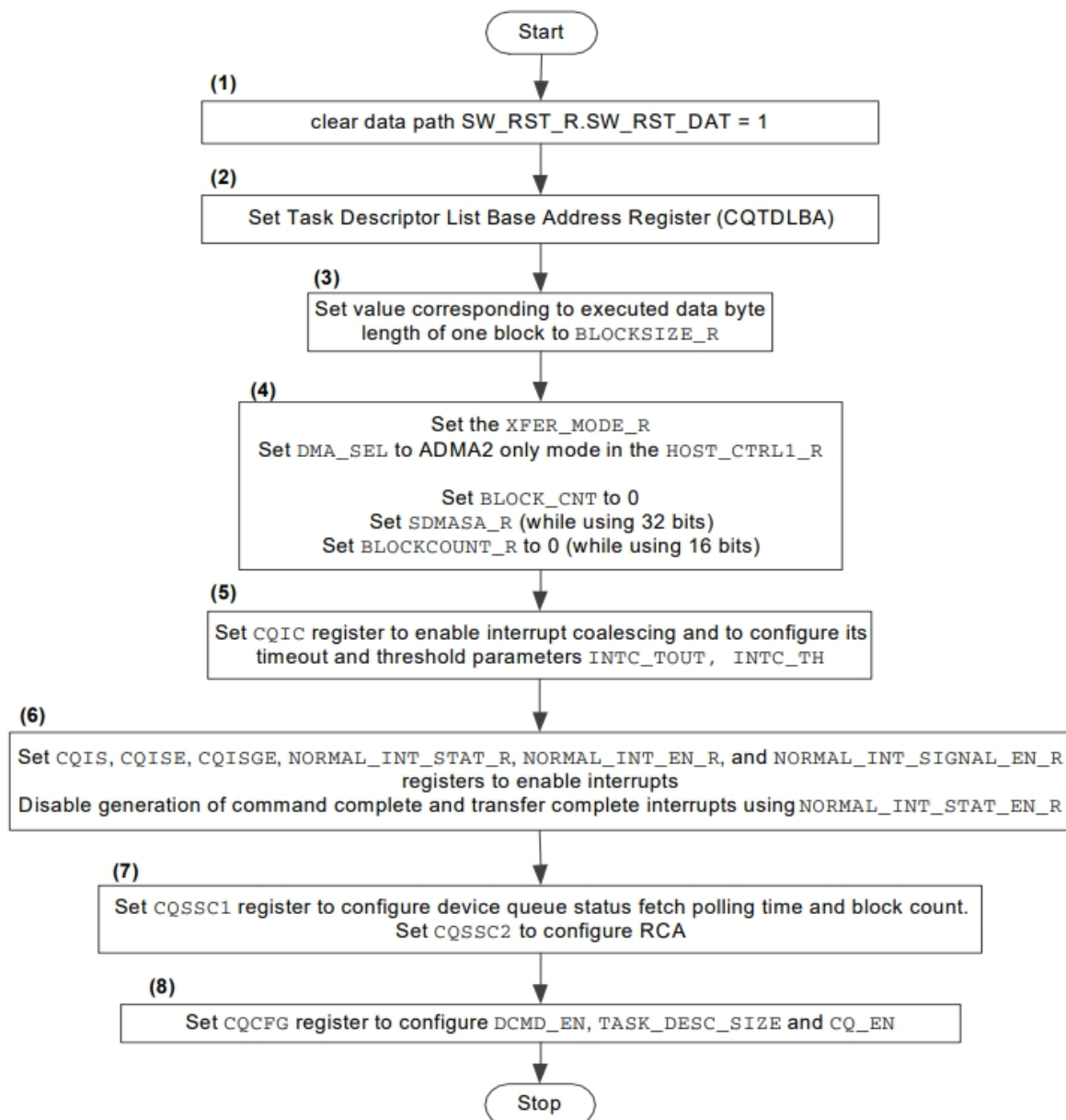


Figure 12-4-38

Initializing the Command Queuing Engine

12.4.2.11.7.2 Task Submission and Completion in a Command Queuing Engine

Figure 12-4-39 shows the sequence to submit and complete a task in a command queuing engine.

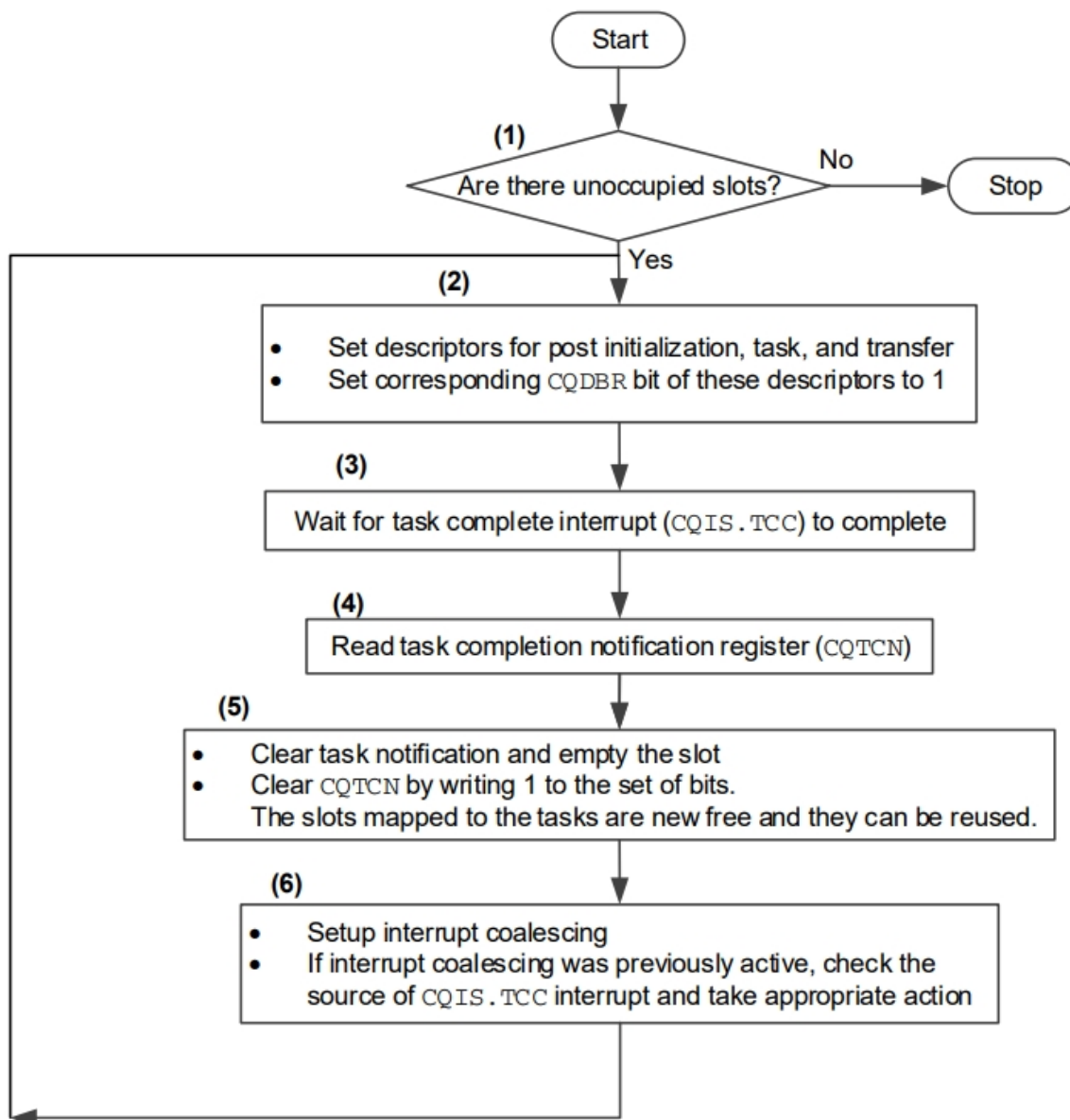


Figure 12-4-39

Submitting and Completing a Task

12.4.3 Register Descriptions

12.4.3.1 Register Summary

Offset Address	Register	Description	Page
0x0	SDMASA_R	SDMA System Address register	
0x4	BLOCKSIZE_R	Block Size register	
0x6	BLOCKCOUNT_R	16-bit Block Count register	
0x8	ARGUMENT_R	Argument register	
0xc	XFER_MODE_R	Transfer Mode register	
0xe	CMD_R	Command register	
0x10	RESP01_R	Response Register 0/1	

Offset Address	Register	Description	Page
0x14	RESP23_R	Response Register 2/3	
0x18	RESP45_R	Response Register 4/5	
0x1c	RESP67_R	Response Register 6/7	
0x20	BUF_DATA_R	Buffer Data Port Register	
0x24	PSTATE_REG	Present State Register	
0x28	HOST_CTRL1_R	Host Control 1 Register	
0x29	PWR_CTRL_R	Power Control Register	
0x2a	BGAP_CTRL_R	Block Gap Control Register	
0x2b	WUP_CTRL_R	Wakeup Control Register	
0x2c	CLK_CTRL_R	Clock Control Register	
0x2e	TOUT_CTRL_R	Timeout Control Register	
0x2f	SW_RST_R	Software Reset Register	
0x30	NORMAL_INT_STAT_R	Normal Interrupt Status Register	
0x32	ERROR_INT_STAT_R	Error Interrupt Status Register	
0x34	NORMAL_INT_STAT_EN_R	Normal Interrupt Status Enable Register	
0x36	ERROR_INT_STAT_EN_R	Error Interrupt Status Enable Register	
0x38	NORMAL_INT_SIGNAL_EN_R	Normal Interrupt Signal Enable Register	
0x3a	ERROR_INT_SIGNAL_EN_R	Error Interrupt Signal Enable Register	
0x3c	AUTO_CMD_STAT_R	Auto CMD Status Register	
0x3e	HOST_CTRL2_R	Host Control 2 Register	

Offset Address	Register	Description	Page
0x40	CAPABILITIES1_R	Capabilities 1 Register - 0 to 31	
0x44	CAPABILITIES2_R	Capabilities Register - 32 to 63	
0x48	CURR_CAPABILITIES1_R	Maximum Current Capabilities Register - 0 to 31	
0x4c	CURR_CAPABILITIES2_R	Maximum Current Capabilities Register - 32 to 63	
0x50	FORCE_AUTO_CMD_STAT_R	Force Event Register for Auto CMD Error Status register	
0x52	FORCE_ERROR_INT_STAT_R	Force Event Register for Error Interrupt Status	
0x54	ADMA_ERR_STAT_R	ADMA Error Status Register	
0x58	ADMA_SA_LOW_R	ADMA System Address Register - Low	
0x5c	ADMA_SA_HIGH_R	ADMA System Address Register - High	
0x60	PRESET_INIT_R	Preset Value for Initialization	
0x62	PRESET_DS_R	Preset Value for Default Speed	
0x64	PRESET_HS_R	Preset Value for High Speed	
0x66	PRESET_SDR12_R	Preset Value for SDR12	
0x68	PRESET_SDR25_R	Preset Value for SDR25	
0x6a	PRESET_SDR50_R	Preset Value for SDR50	
0x6c	PRESET_SDR104_R	Preset Value for SDR104	
0x6e	PRESET_DDR50_R	Preset Value for DDR50	

Offset Address	Register	Description	Page
0x74	PRESET_UHS2_R	Preset Value for UHS-II	
0x78	ADMA_ID_LOW_R	ADMA3 Integrated Descriptor Address Register - Low	
0x7c	ADMA_ID_HIGH_R	ADMA3 Integrated Descriptor Address Register - High	
0x80	UHS_II_BLOCK_SIZE_R	UHS-II Block Size Register	
0x84	UHS_II_BLOCK_COUNT_R	UHS-II Block Count Register	
0x88	UHS_II_COMMAND_PKT_0_3_R	UHS-II Command Packet Register (Byte 0, 1, 2 and 3)	
0x8c	UHS_II_COMMAND_PKT_4_7_R	UHS-II Command Packet Register (Byte 4, 5, 6 and 7)	
0x90	UHS_II_COMMAND_PKT_8_11_R	UHS-II Command Packet Register (Byte 8, 9, 10 and 11)	
0x94	UHS_II_COMMAND_PKT_12_15_R	UHS-II Command Packet Register (Byte 12, 13, 14 and 15)	
0x98	UHS_II_COMMAND_PKT_16_19_R	UHS-II Command Packet Register (Byte 16, 17, 18 and 19)	
0x9c	UHS_II_XFER_MODE_R	UHS-II Transfer Mode Register	
0x9e	UHS_II_CMD_R	UHS-II Command Register	
0xa0	UHS_II_RESP_0_3_R	UHS-II Response Register (Byte 0, 1, 2 and 3)	
0xa4	UHS_II_RESP_4_7_R	UHS-II Response Register (Byte 4, 5, 6 and 7)	
0xa8	UHS_II_RESP_8_11_R	UHS-II Response Register (Byte 8, 9, 10 and 11)	

Offset Address	Register	Description	Page
0xac	UHS_II_RESP_12_15_R	UHS-II Response Register (Byte 12, 13, 14 and 15)	
0xb0	UHS_II_RESP_16_19_R	UHS-II Response Register (Byte 16, 17, 18 and 19)	
0xb4	UHS_II_MSG_SEL_R	UHS-II MSG Select Register	
0xb8	UHS_II_MSG_R	UHS-II MSG Register	
0xbc	UHS_II_DEV_INTR_STATUS_R	UHS-II Device Interrupt Status Register	
0xbe	UHS_II_DEV_SEL_R	UHS-II Device Select Register	
0xbf	UHS_II_DEV_INTR_CODE_R	UHS-II Device Interrupt Code Register	
0xc0	UHS_II_SOFT_RESET_R	UHS-II Software Reset Register	
0xc2	UHS_II_TIMER_CNTRL_R	UHS-II Timer Control Register	
0xc4	UHS_II_ERR_INTR_STATUS_R	UHS-II Error Interrupt Status Register	
0xc8	UHS_II_ERR_INTR_STATUS_EN_R	UHS-II Error Interrupt Status Enable Register	
0xcc	UHS_II_ERR_INTR_SIGNAL_EN_R	UHS-II Error Interrupt Signal Enable Register	
0xe0	P_UHS_II_SETTINGS_R	Pointer for UHS-II Settings	
0xe2	P_UHS_II_HOST_CAPAB	Pointer for UHS-II Host Capabilities	
0xe4	P_UHS_II_TEST	Pointer for UHS-II Test	
0xe6	P_EMBEDDED_CNTRL	Pointer for Embedded Control	
0xe8	P_VENDOR_SPECIFIC_AREA	Pointer for Vendor Specific Area 1	
0xea	P_VENDOR2_SPECIFIC_AREA	Pointer for Vendor Specific Area 2	

Offset Address	Register	Description	Page
0xfc	SLOT_INTR_STATUS_R	Slot Interrupt Status Register	
0xfe	HOST_CNTRL_VERS_R	Host Controller Version	
P_UHS_II_SETTINGS_R[11:0]	UHS2_GEN_SET_R	UHS-II General Setting register	
P_UHS_II_SETTINGS_R[11:0] + 0x4	UHS2_PHY_SET_R	UHS-II Phy Setting register	
P_UHS_II_SETTINGS_R[11:0] + 0x8	UHS2_LNK_TRAN_SET_1_R	UHS-II LINK/TRAN Setting register	
P_UHS_II_SETTINGS_R[11:0] + 0xc	UHS2_LNK_TRAN_SET_2_R	UHS-II LINK/TRAN Setting 2 register	
P_UHS_II_HOST_CAPAB[11:0]	UHS2_GEN_CAP_R	UHS2 General Capabilities register	
P_UHS_II_HOST_CAPAB[11:0] + 0x4	UHS2_PHY_CAP_R	UHS2 PHY Capabilities register	
P_UHS_II_HOST_CAPAB[11:0] + 0x8	UHS2_LNK_TRAN_CAP_1_R	UHS2 Link Tran Capabilities register (0 to 31)	
P_UHS_II_HOST_CAPAB[11:0] + 0xc	UHS2_LNK_TRAN_CAP_2_R	UHS2 Link Tran Capabilities register (32 to 63)	
P_UHS_II_TEST[11:0]	FORCE_UHS_II_ERR_INTR_STATUS_R	Force Event for UHS-II Error Interrupt Status Register	
P_EMBEDDED_CNTRL[11:0]	EMBEDDED_CTRL_R	Embedded Control register	
P_VENDOR2_SPECIFIC_AREA	CQVER	Command Queuing Version register	
P_VENDOR2_SPECIFIC_AREA + 0x4	CQCAP	Command Queuing Capabilities register	
P_VENDOR2_SPECIFIC_AREA + 0x8	CQCFG	Command Queuing Configuration register	
P_VENDOR2_SPECIFIC_AREA + 0xc	CQCTL	Command Queuing Control register	
P_VENDOR2_SPECIFIC_AREA + 0x10	CQIS	Command Queuing Interrupt Status register	

Offset Address	Register	Description	Page
P_VENDOR2_SPECIFIC_AREA + 0x14	CQISE	Command Queuing Interrupt Status Enable register	
P_VENDOR2_SPECIFIC_AREA + 0x18	CQISGE	Command Queuing Interrupt signal enable register	
P_VENDOR2_SPECIFIC_AREA + 0x1c	CQIC	Command Queuing Interrupt Coalescing register	
P_VENDOR2_SPECIFIC_AREA + 0x20	CQTDLBA	Command Queuing Task Descriptor List Base Address register	
P_VENDOR2_SPECIFIC_AREA + 0x24	CQTDLBAU	Command Queuing Task Descriptor List Base Address Upper register	
P_VENDOR2_SPECIFIC_AREA + 0x28	CQTDBR	Command Queuing DoorBell register	
P_VENDOR2_SPECIFIC_AREA + 0x2c	CQTCN	Command Queuing TaskClear Notification register	
P_VENDOR2_SPECIFIC_AREA + 0x30	CQDQS	Device queue status register	
P_VENDOR2_SPECIFIC_AREA + 0x34	CQDPT	Device pending tasks register	
P_VENDOR2_SPECIFIC_AREA + 0x38	CQTCLR	Command Queuing DoorBell register	
P_VENDOR2_SPECIFIC_AREA + 0x40	CQSSC1	CQ Send Status Configuration 1 register	
P_VENDOR2_SPECIFIC_AREA + 0x44	CQSSC2	CQ Send Status Configuration 2 register	
P_VENDOR2_SPECIFIC_AREA + 0x48	CQCRDCT	Command response for direct command register	
P_VENDOR2_SPECIFIC_AREA + 0x50	CQRMEM	Command response mode error mask register	
P_VENDOR2_SPECIFIC_AREA + 0x54	CQTERRI	CQ Task Error Information register	

Offset Address	Register	Description	Page
P_VENDOR2_SPECIFIC_AREA + 0x58	CQCRI	CQ Command response index	
P_VENDOR2_SPECIFIC_AREA + 0x5c	CQCRA	CQ Command response argument register	
DWC_MSHC_PTR_PHY_REGS + 0x0	PHY_CNFG	SD/eMMC PHY General Configuration	
DWC_MSHC_PTR_PHY_REGS + 0x4	CMDPAD_CNFG	SD/eMMC PHY CMD/RESP PAD Setting	
DWC_MSHC_PTR_PHY_REGS + 0x6	DATPAD_CNFG	SD/eMMC PHY Data PAD Setting	
DWC_MSHC_PTR_PHY_REGS + 0x8	CLKPAD_CNFG	SD/eMMC PHY Clock PAD Setting	
DWC_MSHC_PTR_PHY_REGS + 0xA	STBPAD_CNFG	SD/eMMC PHY Strobe PAD Setting	
DWC_MSHC_PTR_PHY_REGS + 0xC	RSTNPAD_CNFG	SD/eMMC PHY RSTN PAD Setting	
DWC_MSHC_PTR_PHY_REGS + 0xE	PADTEST_CNFG	SD/eMMC PHY PAD TEST interface Setting	
DWC_MSHC_PTR_PHY_REGS + 0x10	PADTEST_OUT	SD/eMMC PHY PAD TEST Data out value	
DWC_MSHC_PTR_PHY_REGS + 0x12	PADTEST_IN	SD/eMMC PHY PAD TEST Data in value	
DWC_MSHC_PTR_PHY_REGS + 0x18	PRBS_CNFG	Controller PRBS Config register	
DWC_MSHC_PTR_PHY_REGS + 0x1A	PHYLPBK_CNFG	Loopback Config register	
DWC_MSHC_PTR_PHY_REGS + 0x1C	COMMDL_CNFG	Common DelayLine config settings register	
DWC_MSHC_PTR_PHY_REGS + 0x1D	SDCLKDL_CNFG	SD/eMMC DelayLine settings	
DWC_MSHC_PTR_PHY_REGS + 0x1E	SDCLKDL_DC	SD/eMMC DelayLine Delay code setting	
DWC_MSHC_PTR_PHY_REGS + 0x20	SMPLDL_CNFG	SD/eMMC cclk_rx DelayLine settings	
DWC_MSHC_PTR_PHY_REGS + 0x21	ATDL_CNFG	SD/eMMC drift_cclk_rx DelayLine configuration settings	

Offset Address	Register	Description	Page
DWC_MSHC_PTR_PHY_REGS + 0x24	DLL_CTRL	SD/eMMC PHY DLL control setting	
DWC_MSHC_PTR_PHY_REGS + 0x25	DLL_CNFG1	DLL Config register 1	
DWC_MSHC_PTR_PHY_REGS + 0x26	DLL_CNFG2	DLL Config register 2	
DWC_MSHC_PTR_PHY_REGS + 0x28	DLLDL_CNFG	DLL Config register 2	
DWC_MSHC_PTR_PHY_REGS + 0x29	DLL_OFFST	DLL Offset setting register	
DWC_MSHC_PTR_PHY_REGS + 0x2A	DLLMST_TSTDC	DLL Master test code setting register	
DWC_MSHC_PTR_PHY_REGS + 0x2C	DLLLBT_CNFG	DLL LBT setting register	
DWC_MSHC_PTR_PHY_REGS + 0x2E	DLL_STATUS	DLL Status register	
DWC_MSHC_PTR_PHY_REGS + 0x30	DLLDBG_MLKDC	DLL Master lock code debug register	
DWC_MSHC_PTR_PHY_REGS + 0x32	DLLDBG_SLKDC	DLL Master Slave code debug register	
P_VENDOR_SPECIFIC_AREA[11:0]	MSHC_VER_ID_R	MSHC version	
P_VENDOR_SPECIFIC_AREA[11:0] + 0x4	MSHC_VER_TYPE_R	MSHC version type	
P_VENDOR_SPECIFIC_AREA[11:0] + 0x8	MSHC_CTRL_R	MSHC Control register	
P_VENDOR_SPECIFIC_AREA[11:0] + 0x10	MBIU_CTRL_R	MBIU Control register	
P_VENDOR_SPECIFIC_AREA[11:0] + 0x2c	EMMC_CTRL_R	eMMC Control register	
P_VENDOR_SPECIFIC_AREA[11:0] + 0x2e	BOOT_CTRL_R	eMMC Boot Control register	
P_VENDOR_SPECIFIC_AREA[11:0] + 0x30	GP_IN_R	General Purpose Input register	
P_VENDOR_SPECIFIC_AREA[11:0] + 0x34	GP_OUT_R	General Purpose Output register	
P_VENDOR_SPECIFIC_AREA[11:0] + 0x40	AT_CTRL_R	Tuning and Auto-tuning control register	
P_VENDOR_SPECIFIC_AREA[11:0] + 0x44	AT_STAT_R	Tuning and Auto-tuning status register	

12.4.3.2 Register Descriptions

SDMASA_R

SDMASA_R is a SDMA System Address register

Offset Address : 0x0 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
31:0	R/W	BLOCKCNT_SDMA_SA	32-bit Block Count (SDMA System Address) - SDMA System Address (Host Version 4 Enable = 0): This register contains the system memory address for an SDMA transfer in the 32-bit addressing mode. When the Host Controller stops an SDMA transfer, this register points to the system address of the next contiguous data position. It can be accessed only if no transaction is executing. Reading this register during data transfers may return an invalid value. - 32-bit Block Count (Host Version 4 Enable = 1): From the Host Controller Version 4.10 specification, this register is redefined as 32-bit Block Count. The Host Controller decrements the block count of this register for every block transfer and the data transfer stops when the count reaches zero. This register must be accessed when no transaction is executing. Reading this register during data transfers may return invalid value. Following are the values for BLOCKCNT_SDMA_SA: - 0xFFFF_FFFF: 4G - 1 Block - ... - 0x0000_0002: 2 Blocks - 0x0000_0001: 1 Block - 0x0000_0000: Stop Count Note: - For Host Version 4 Enable = 0, the Host driver does not program the system address in this register while operating in ADMA mode. The system address must be programmed in the ADMA System Address register. - For Host Version 4 Enable = 0, the Host driver programs a non-zero 32-bit block count value in this register when Auto CMD23 is enabled for non-DMA and ADMA modes. Auto CMD23 cannot be used with SDMA. - This register must be programmed with a non-zero value for data transfer if the 32-bit Block count register is used instead of the 16-bit Block count register.	0x0

BLOCKSIZE_R

BLOCKSIZE_R is a Block Size register Offset Address : 0x4 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
15	R	RSVD_BLOCKSIZE15	This bit of the BLOCKSIZE_R register is reserved. It always returns 0.	0x0

Bits	Access	Name	Description	Reset
14:12	R/W	SDMA_BUF_BDARY	SDMA Buffer Boundary These bits specify the size of contiguous buffer in system memory. The SDMA transfer waits at every boundary specified by these fields and the Host Controller generates the DMA interrupt to request the Host Driver to update the SDMA System Address register. Values: ■ 0x0 (BYTES_4K): 4K bytes SDMA Buffer Boundary ■ 0x1 (BYTES_8K): 8K bytes SDMA Buffer Boundary ■ 0x2 (BYTES_16K): 16K bytes SDMA Buffer Boundary ■ 0x3 (BYTES_32K): 32K bytes SDMA Buffer Boundary ■ 0x4 (BYTES_64K): 64K bytes SDMA Buffer Boundary ■ 0x5 (BYTES_128K): 128K bytes SDMA Buffer Boundary ■ 0x6 (BYTES_256K): 256K bytes SDMA Buffer Boundary ■ 0x7 (BYTES_512K): 512K bytes SDMA Buffer Boundary	0x0
11:0	R/W	XFER_BLOCK_SIZE	Transfer Block Size These bits specify the block size of data transfers. In case of memory, it is set to 512 bytes. It can be accessed only if no transaction is executing. Read operations during transfers may return an invalid value, and write operations are ignored. Following are the values for XFER_BLOCK_SIZE: ■ 0x1: 1 byte ■ 0x2: 2 bytes ■ 0x3: 3 bytes ■ ■ 0x1FF: 511 byte ■ 0x200: 512 bytes ■ ■ 0x800: 2048 bytes Note: This register must be programmed with a non-zero value for data transfer.	0x0

BLOCKCOUNT_R

BLOCKCOUNT_R is a 16-bit Block Count register Offset Address : 0x6 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset

Bits	Access	Name	Description	Reset
15:0	R/W	BLOCK_CNT	16-bit Block Count ■ If the Host Version 4 Enable bit is set 0 or the 16-bit Block Count register is set to non-zero, the 16-bit Block Count register is selected. ■ If the Host Version 4 Enable bit is set 1 and the 16-bit Block Count register is set to zero, the 32-bit Block Count register is selected. Following are the values for BLOCK_CNT: ■ 0x0: Stop Count ■ 0x1: 1 Block ■ 0x2: 2 Blocks ■ ... - ... ■ 0xFFFF: 65535 Blocks Note: For Host Version 4 Enable = 0, this register must be set to 0000h before programming the 32-bit block count register when Auto CMD23 is enabled for non-DMA and ADMA modes.	0x0

ARGUMENT_R

ARGUMENT_R is an Argument register Offset Address : 0x8 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
31:0	R/W	ARGUMENT	Command Argument These bits specify the SD/eMMC command argument that is specified in bits 39-8 of the Command format.	0x0

XFER_MODE_R

XFER_MODE_R is a Transfer Mode register Offset Address : 0xc Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
15:9	R	RSVD	These bits of the XFER_MODE_R register are reserved. They always return 0.	0x0

Bits	Access	Name	Description	Reset
8	R/W	RESP_INT_DISABLE	Response Interrupt Disable The Host Controller supports response check function to avoid overhead of response error check by the Host driver. Response types of only R1 and R5 can be checked by the Controller. If Host Driver checks the response error, set this bit to 0 and wait for Command Complete Interrupt and then check the response register. If the Host Controller checks the response error, set this bit to 1 and set the Response Error Check Enable bit to 1. The Command Complete Interrupt is disabled by this bit regardless of the Command Complete Signal Enable. Note: During tuning (when the Execute Tuning bit in the Host Control2 register is set), the Command Complete Interrupt is not generated irrespective of the Response Interrupt Disable setting. Values: ■ 0x0 (ENABLED): Response Interrupt is enabled ■ 0x1 (DISABLED): Response Interrupt is disabled	0x0
7	R/W	RESP_ERR_CHK_ENABLE	Response Error Check Enable The Host Controller supports response check function to avoid overhead of response error check by Host driver. Response types of only R1 and R5 can be checked by the Controller. If the Host Controller checks the response error, set this bit to 1 and set Response Interrupt Disable to 1. If an error is detected, the Response Error interrupt is generated in the Error Interrupt Status register. Note: ■ Response error check must not be enabled for any response type other than R1 and R5. ■ Response check must not be enabled for the tuning command. Values: ■ 0x0 (DISABLED): Response Error Check is disabled ■ 0x1 (ENABLED): Response Error Check is enabled	0x0

Bits	Access	Name	Description	Reset
6	R/W	RESP_TYPE	Response Type R1/R5 This bit selects either R1 or R5 as a response type when the Response Error Check is selected. Error statuses checked in R1: ■ OUT_OF_RANGE ■ ADDRESS_ERROR ■ BLOCK_LEN_ERROR ■ WP_VIOLATION ■ CARD_IS_LOCKED ■ COM_CRC_ERROR ■ CARD_ECC_FAILED ■ CC_ERROR ■ ERROR Response Flags checked in R5: ■ COM_CRC_ERROR ■ ERROR ■ FUNCTION_NUMBER ■ OUT_OF_RANGE Values: ■ 0x0 (RESP_R1): R1 (Memory) ■ 0x1 (RESP_R5): R5 (SDIO)	0x0
5	R/W	MULTI_BLK_SEL	Multi/Single Block Select This bit is set when issuing multiple-block transfer commands using the DAT line. If this bit is set to 0, it is not necessary to set the Block Count register. Values: ■ 0x0 (SINGLE): Single Block ■ 0x1 (MULTI): Multiple Block	0x0
4	R/W	DATA_XFER_DIR	Data Transfer Direction Select This bit defines the direction of DAT line data transfers. This bit is set to 1 by the Host Driver to transfer data from the SD/eMMC card to the Host Controller and it is set to 0 for all other commands. Values: ■ 0x1 (READ): Read (Card to Host) ■ 0x0 (WRITE): Write (Host to Card)	0x0
3:2	R/W	AUTO_CMD_ENABLE	Auto Command Enable This field determines use of Auto Command functions. Note: In SDIO, this field must be set as 00b (Auto Command Disabled). Values: ■ 0x0 (AUTO_CMD_DISABLED): Auto Command Disabled ■ 0x1 (AUTO_CMD12_ENABLED): Auto CMD12 Enable ■ 0x2 (AUTO_CMD23_ENABLED): Auto CMD23 Enable ■ 0x3 (AUTO_CMD_AUTO_SEL): Auto CMD Auto Select	0x0

Bits	Access	Name	Description	Reset
1	R/W	BLOCK_COUNT_ENABLE	Block Count Enable This bit is used to enable the Block Count register, which is relevant for multiple block transfers. If this bit is set to 0, the Block Count register is disabled, which is useful in executing an infinite transfer. Values: ■ 0x1 (ENABLED): Enable ■ 0x0 (DISABLED): Disable	0x0
0	R/W	DMA_ENABLE	DMA Enable This bit enables the DMA functionality. If this bit is set to 1, a DMA operation begins when the Host Driver writes to the Command register. You can select one of the DMA modes by using DMA Select in the Host Control 1 register. Values: ■ 0x1 (ENABLED): DMA Data transfer ■ 0x0 (DISABLED): No data transfer or Non-DMA data transfer	0x0

CMD_R

CMD_R is a Command register Offset Address : 0xe Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
15:14	R	RSVD	These bits of the CMD_R register are reserved. They always return 0.	0x0
13:8	R/W	CMD_INDEX	Command Index These bits are set to the command number that is specified in bits 45-40 of the Command Format.	0x0
7:6	R/W	CMD_TYPE	Command Type These bits indicate the command type. Note: While issuing Abort CMD using CMD12/CMD52 or reset CMD using CMD0/CMD52, CMD_TYPE field shall be set to 0x3. In case CMD0 is used for initialization set as 0x0 Values: ■ 0x3 (ABORT_CMD): Abort ■ 0x2 (RESUME_CMD): Resume ■ 0x1 (SUSPEND_CMD): Suspend ■ 0x0 (NORMAL_CMD): Normal	0x0

Bits	Access	Name	Description	Reset
5	R/W	DATA_PRESENT_SEL	Data Present Select This bit is set to 1 to indicate that data is present and that the data is transferred using the DAT line. This bit is set to 0 in the following instances: ■ Command using the CMD line ■ Command with no data transfer but using busy signal on the DAT[0] line ■ Resume Command Values: ■ 0x0 (NO_DATA): No Data Present ■ 0x1 (DATA): Data Present	0x0
4	R/W	CMD_IDX_CHK_ENABLE	Command Index Check Enable This bit enables the Host Controller to check the index field in the response to verify if it has the same value as the command index. If the value is not the same, it is reported as a Command Index error. Note: ■ Index Check enable must be set to 0 for the command with no response, R2 response, R3 response and R4 response. ■ For the tuning command, this bit must always be set to enable the index check. Values: ■ 0x0 (DISABLED): Disable ■ 0x1 (ENABLED): Enable	0x0
3	R/W	CMD_CRC_CHK_ENABLE	Command CRC Check Enable This bit enables the Host Controller to check the CRC field in the response. If an error is detected, it is reported as a Command CRC error. Note: ■ CRC Check enable must be set to 0 for the command with no response, R3 response, and R4 response. ■ For the tuning command, this bit must always be set to 1 to enable the CRC check. Values: ■ 0x0 (DISABLED): Disable ■ 0x1 (ENABLED): Enable	0x0
2	R/W	SUB_CMD_FLAG	Sub Command Flag This bit distinguishes between a main command and a sub command. Values: ■ 0x0 (MAIN): Main Command ■ 0x1 (SUB): Sub Command	0x0

Bits	Access	Name	Description	Reset
1:0	R/W	RESP_TYPE_SELECT	Response Type Select This bit indicates the type of response expected from the card. Values: ■ 0x0 (NO_RESP): No Response ■ 0x1 (RESP_LEN_136): Response Length 136 ■ 0x2 (RESP_LEN_48): Response Length 48 ■ 0x3 (RESP_LEN_48B): Response Length 48; Check Busy after response	0x0

RESP01_R

RESP01_R is a Response Register 0/1 Offset Address : 0x10 Total Reset Value: 0x0

Bits	Name	Access	Description	Reset
31:0	R	RESP01	Command Response These bits reflect 39-8 bits of SD/eMMC Response Field. In UHS-II mode, it stores the response of the TRANS_ABORT CCMD. Note: For Auto CMD, the 32-bit response (bits 39-8 of the Response Field) is updated in the RESP67_R register.	

RESP23_R

RESP23_R is a Response Register 2/3 Offset Address : 0x14 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
31:0	R	RESP23	Command Response These bits reflect 71-40 bits of the SD/eMMC Response Field. In UHS-II mode, it is reserved.	0x0

RESP45_R

RESP45_R is a Response Register 4/5 Offset Address : 0x18 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
31:0	R	RESP45	Command Response These bits reflect 103-72 bits of the Response Field. In UHSII mode, it stores the lower 4-byte CMD12 response.	0x0

RESP67_R

RESP67_R is a Response Register 6/7 Offset Address : 0x1c Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
31:0	R	RESP67	Command Response These bits reflect bits 135-104 of SD/EMMC Response Field. In UHS-II mode, it stores the upper 4-byte CMD12 response. Note: For Auto CMD, this register also reflects the 32-bit response (bits 39-8 of the Response Field).	0x0

BUF_DATA_R

BUF_DATA_R is a Buffer Data Port Register Offset Address : 0x20 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
31:0	R/W	BUF_DATA	Buffer Data These bits enable access to the Host Controller packet buffer.	0x0

PSTATE_REG

PSTATE_REG is a Present State Register Offset Address : 0x24 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
31	R	UHS2_IF_DETECT	UHS-II Interface Detection This bit indicates whether a card supports the UHS-II interface. For SD/eMMC mode, this bit always returns 0. Values: ■ 0x0 (FALSE): UHS-II interface is not detected ■ 0x1 (TRUE): UHS-II interface is detected	0x0
30	R	LANE_SYNC	Lane Synchronization This bit indicates whether a lane is synchronized in the UHSII mode. For SD/eMMC mode, this bit always returns 0. Values: ■ 0x0 (FALSE): UHS-II PHY is not initialized ■ 0x1 (TRUE): UHS-II PHY is initialized	0x0
29	R	IN_DORMANT_ST	In Dormant Status This bit indicates whether UHS-II lanes enter Dormant state in the UHS-II mode. For SD/eMMC mode, this bit always returns 0. Values: ■ 0x0 (FALSE): Not in DORMANT state ■ 0x1 (TRUE): In DORMANT state	0x0
28	R	SUB_CMD_STAT	Sub Command Status This bit is used to distinguish between a main command and a sub command status. Values: ■ 0x0 (FALSE): Main Command Status ■ 0x1 (TRUE): Sub Command Status	0x0
27	R	CMD_ISSUE_ERR	Command Not Issued by Error This bit is set if a command cannot be issued after setting the command register due to an error except the Auto CMD12 error. Values: ■ 0x0 (FALSE): No error for issuing a command ■ 0x1 (TRUE): Command cannot be issued	0x0

Bits	Access	Name	Description	Reset
26	R	RSVD_26	This bit of the PRESENT_ST_R register is reserved bits. It always returns 0.	0x0
25	R	HOST_REG_VOL	Host Regulator Voltage Stable This bit is used to check whether the host regulator voltage is stable for switching the voltage of UHS-I mode. This bit reflects the synchronized value of the host_reg_vol_stable signal. Values: ■ 0x0 (FALSE): Host Regulator Voltage is not stable ■ 0x1 (TRUE): Host Regulator Voltage is stable	0x0
24	R	CMD_LINE_LVL	Command-Line Signal Level This bit is used to check the CMD line level to recover from errors and for debugging. These bits reflect the value of the sd_cmd_in signal. This bit is irrelevant for an UHS-II mode and always returns 0.	0x0
23:20	R	DAT_3_0	DAT[3:0] Line Signal Level This bit is used to check the DAT line level to recover from errors and for debugging. These bits reflect the value of the sd_dat_in (lower nibble) signal. These bits are irrelevant for the UHS-II mode and always returns 0.	0x0
19	R	WR_PROTECT_SW_LVL	Write Protect Switch Pin Level This bit is supported only for memory and combo cards. This bit reflects the synchronized value of the card_write_protsignal. Values: ■ 0x0 (FALSE): Write protected ■ 0x1 (TRUE): Write enabled	0x0
18	R	CARD_DETECT_PIN_LEVEL	Card Detect Pin Level This bit reflects the inverse synchronized value of the card_detect_n signal. Values: ■ 0x0 (FALSE): No card present ■ 0x1 (TRUE): Card Present	0x0
17	R	CARD_STABLE	Card Stable This bit indicates the stability of the Card Detect Pin Level. A card is not detected if this bit is set to 1 and the value of the CARD_INSERTED bit is 0. Values: ■ 0x0 (FALSE): Reset or Debouncing ■ 0x1 (TRUE): No Card or Inserted	0x0

Bits	Access	Name	Description	Reset
16	R	CARD_INSERTED	Card Inserted This bit indicates whether a card has been inserted. The Host Controller debounces this signal so that Host Driver need not wait for it to stabilize. Values: ■ 0x0 (FALSE): Reset, Debouncing, or No card ■ 0x1 (TRUE): Card Inserted	0x0
15:12	R	RSVD_15_12	These bits of the PRESENT_STAT_R register are reserved. They always return 0.	0x0
11	R	BUF_RD_ENABLE	Buffer Read Enable This bit is used for non-DMA transfers. This bit is set if valid data exists in the Host buffer. Values: ■ 0x0 (DISABLED): Read disable ■ 0x1 (ENABLED): Read enable	0x0
10	R	BUF_WR_ENABLE	Buffer Write Enable This bit is used for non-DMA transfers. This bit is set if space is available for writing data. Values: ■ 0x0 (DISABLED): Write disable ■ 0x1 (ENABLED): Write enable	0x0
9	R	RD_XFER_ACTIVE	Read Transfer Active This bit indicates whether a read transfer is active for SD/eMMC mode. This bit irrelevant for the UHS-II mode and always returns 0. Values: ■ 0x0 (INACTIVE): No valid data ■ 0x1 (ACTIVE): Transferring data	0x0
8	R	WR_XFER_ACTIVE	Write Transfer Active This status indicates whether a write transfer is active for SD/eMMC mode. It is irrelevant for UHS-II mode and always returns 0. Values: ■ 0x0 (INACTIVE): No valid data ■ 0x1 (ACTIVE): Transferring data	0x0
7:4	R	DAT_7_4	DAT[7:4] Line Signal Level This bit is used to check the DAT line level to recover from errors and for debugging. These bits reflect the value of the sd_dat_in (upper nibble) signal. These bits are irrelevant for the UHS-II mode and always return 0.	0x0
3	R	RE_TUNE_REQ	Re-Tuning Request DWC_mshc does not generate retuning request. The software must maintain the Retuning timer.	0x0

Bits	Access	Name	Description	Reset
2	R	DAT_LINE_ACTIVE	DAT Line Active (SD/eMMC Mode only) This bit indicates whether one of the DAT lines on the SD/eMMC bus is in use. This bit is irrelevant for the UHS-II mode. In the case of read transactions, this bit indicates whether a read transfer is executing on the SD/eMMC bus. In the case of write transactions, this bit indicates whether a write transfer is executing on the SD/eMMC bus. For a command with busy, this status indicates whether the command executing busy is executing on an SD or eMMC bus. Values: ■ 0x0 (INACTIVE): DAT Line Inactive ■ 0x1 (ACTIVE): DAT Line Active	0x0
1	R	CMD_INHIBIT_DAT	Command Inhibit (DAT) This bit is applicable for SD/eMMC mode and is generated if either DAT line active or Read transfer active is set to 1. If this bit is set to 0, it indicates that the Host Controller can issue subsequent SD/eMMC commands. For the UHS-II mode, this bit is irrelevant and always returns 0. Values: ■ 0x0 (READY): Can issue command which used DAT line ■ 0x1 (NOT_READY): Cannot issue command which used DAT line	0x0
0	R	CMD_INHIBIT	Command Inhibit (CMD) This bit indicates the following : ■ SD/eMMC mode: If this bit is set to 0, it indicates that the CMD line is not in use and the Host controller can issue an SD/eMMC command using the CMD line. This bit is set when the command register is written. This bit is cleared when the command response is received. This bit is not cleared by the response of auto CMD12/23 but cleared by the response of read/write command. ■ UHS-II mode: If this bit is set to 0, it indicates that a command packet can be issued by the Host Controller. Values: ■ 0x0 (READY): Host Controller is ready to issue a command ■ 0x1 (NOT_READY): Host Controller is not ready to issue a command	0x0

HOST_CTRL1_R

HOST_CTRL1_R is a Host Control 1 Register Offset Address : 0x28 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
7	R/W	CARD_DETECT_SIG_SEL	Card Detect Signal Selection This bit selects a source for card detection. When the source for the card detection is switched, the interrupt must be disabled during the switching period. Values: ■ 0x1 (CARD_DT_TEST_LEVEL): Card Detect Test Level is selected (for test purpose) ■ 0x0 (SDCD_PIN): SDCD# (card_detect_n signal) is selected (for normal use)	0x0
6	R/W	CARD_DETECT_TEST_LVL	Card Detect Test Level This bit is enabled while the Card Detect Signal Selection is set to 1 and it indicates whether a card inserted or not. Values: ■ 0x1 (CARD_INSERTED): Card Inserted ■ 0x0 (No_CARD): No Card	0x0
5	R/W	EXT_DAT_XFER	Extended Data Transfer Width This bit controls 8-bit bus width mode of embedded device. This bit is not applicable for UHS-II mode. Values: ■ 0x1 (EIGHT_BIT): 8-bit Bus Width ■ 0x0 (DEFAULT): Bus Width is selected by the Data Transfer Width	0x0
4:3	R/W	DMA_SEL	DMA Select This field is used to select the DMA type. When Host Version 4 Enable is 1 in Host Control 2 register: ■ 0x0 : SDMA is selected ■ 0x1 : Reserved ■ 0x2 : ADMA2 is selected ■ 0x3 : ADMA2 or ADMA3 is selected When Host Version 4 Enable is 0 in Host Control 2 register: ■ 0x0 : SDMA is selected ■ 0x1 : Reserved ■ 0x2 : 32-bit Address ADMA2 is selected ■ 0x3 : 64-bit Address ADMA2 is selected Values: ■ 0x0 (SDMA): SDMA is selected ■ 0x1 (RSVD_BIT): Reserved ■ 0x2 (ADMA2): ADMA2 is selected ■ 0x3 (ADMA2_3): ADMA2 or ADMA3 is selected	0x0

Bits	Access	Name	Description	Reset
2	R/W	HIGH_SPEED_EN	High Speed Enable (SD/eMMC Mode only) In SD/eMMC mode, this bit is used to determine the selection of preset value for High Speed mode. Before setting this bit, the Host Driver checks the High Speed Support in the Capabilities register. Note: DWC_MSHC always outputs the sd_cmd_out and sd_dat_out lines at the rising edge of cclk_tx clock irrespective of this bit. Please refer the section Connecting the Clock IO interface in the Mobile Storage Host Controller user guide on clocking requirement for an SD/eMMC card. In UHS-II mode, this bit is irrelevant. Values: ■ 0x1 (HIGH_SPEED): High Speed mode ■ 0x0 (NORMAL_SPEED): Normal Speed mode	0x0
1	R/W	DAT_XFER_WIDTH	Data Transfer Width For SD/eMMC mode, this bit selects the data transfer width of the Host Controller. The Host Driver sets it to match the data width of the SD/eMMC card. In UHS-II mode, this bit is irrelevant. Values: ■ 0x1 (FOUR_BIT): 4-bit mode ■ 0x0 (ONE_BIT): 1-bit mode	0x0
0	R/W	LED_CTRL	LED Control This bit is used to caution the user not to remove the card while the SD card is being accessed. The value is reflected on the led_control signal. Values: ■ 0x0 (OFF): LED off ■ 0x1 (ON): LED on	0x0

[PWR_CTRL_R](#)

PWR_CTRL_R is a Power Control Register Offset Address : 0x29 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset

Bits	Access	Name	Description	Reset
7:5	R/W	SD_BUS_VOL_VDD2	SD Bus Voltage Select for VDD2. This bit determines supply voltage range to VDD2 of UHS-II card. This bit be can set to 0x5 if 1.8V VDD2 Support in the Capabilities register is set to 1. This is irrelevant for SD/eMMC card. Values: ■ 0x7 (NOT_USED7): Not used ■ 0x6 (NOT_USED6): Not used ■ 0x5 (V_1_8): 1.8V ■ 0x4 (V_1_2): Reserved for 1.2V ■ 0x3 (RSVD3): Reserved ■ 0x2 (RSVD2): Reserved ■ 0x1 (RSVD1): Reserved ■ 0x0 (NO_VDD2): VDD2 Not Supported	0x0
4	R/W	SD_BUS_PWR_VDD2	SD Bus Power for VDD2. This bit enables VDD2 power of UHS-II card. This setting is available on the sd_vdd2_on output of DWC_mshc so that it can be used to control the VDD2 power supply of the card. This is irrelevant for SD/eMMC card. Values: ■ 0x0 (OFF): Power off ■ 0x1 (ON): Power on	0x0

Bits	Access	Name	Description	Reset
3:1	R/W	SD_BUS_VOL_VDD1	SD Bus Voltage Select for VDD1/eMMC Bus Voltage Select for VDD These bits enable the Host Driver to select the voltage level for an SD/eMMC card. Before setting this register, the Host Driver checks the Voltage Support bits in the Capabilities register. If an unsupported voltage is selected, the Host System does not supply the SD Bus voltage. The value set in this field is available on the DWC_mshc output signal (sd_vdd1_sel), which is used by the voltage switching circuitry. SD Bus Voltage Select options: ■ 0x7 : 3.3V(Typical) ■ 0x6 : 3.0V(Typical) ■ 0x5 : 1.8V(Typical) for Embedded ■ 0x4 : 0x0 - Reserved eMMC Bus Voltage Select options: ■ 0x7 : 3.3V(Typical) ■ 0x6 : 1.8V(Typical) ■ 0x5 : 1.2V(Typical) ■ 0x4 : 0x0 - Reserved Values: ■ 0x7 (V_3_3): 3.3V (Typ.) ■ 0x6 (V_3_0): 3.0V (Typ.) ■ 0x5 (V_1_8): 1.8V (Typ.) for Embedded ■ 0x4 (RSVD4): Reserved ■ 0x3 (RSVD3): Reserved ■ 0x2 (RSVD2): Reserved ■ 0x1 (RSVD1): Reserved ■ 0x0 (RSVD0): Reserved	0x0
0	R/W	SD_BUS_PWR_VDD1	SD Bus Power for VDD1 This bit enables VDD1 power of the card. This setting is available on the sd_vdd1_on output of DWC_mshc so that it can be used to control the VDD1 power supply of the card. Before setting this bit, the SD Host Driver sets the SD Bus Voltage Select bit. If the Host Controller detects a No Card state, this bit is cleared. In SD mode, if this bit is cleared, the Host Controller stops the SD Clock by clearing the SD_CLK_IN bit in the CLK_CTRL_R register. In UHS-II mode, before clearing this bit, the Host Driver clears the SD_CLK_IN bit in the CLK_CTRL_R register. Values: ■ 0x0 (OFF): Power off ■ 0x1 (ON): Power on	0x0

BGAP_CTRL_R

BGAP_CTRL_R is a Block Gap Control Register Offset Address : 0x2a Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
7:4	R	RSVD_7_4	These bits of the Block Gap Control register are reserved. They always return 0.	0x0
3	R/W	INT_AT_BGAP	Interrupt At Block Gap This bit is valid only in the 4-bit mode of an SDIO card and is used to select a sample point in the interrupt cycle. Setting to 1 enables interrupt detection at the block gap for a multiple block transfer. In UHS-II mode, this bit is disabled. Values: ■ 0x0 (DISABLE): Disabled ■ 0x1 (ENABLE): Enabled	0x0
2	R/W	RD_WAIT_CTRL	Read Wait Control This bit is used to enable the read wait protocol to stop read data using DAT[2] line if the card supports read wait. Otherwise, the Host Controller has to stop the card clock to hold the read data. In UHS-II mode, Read Wait is disabled. Values: ■ 0x0 (DISABLE): Disable Read Wait Control ■ 0x1 (ENABLE): Enable Read Wait Control	0x0
1	R/W	CONTINUE_REQ	Continue Request This bit is used to restart the transaction, which was stopped using the Stop At Block Gap Request. The Host Controller automatically clears this bit when the transaction restarts. If stop at block gap request is set to 1, any write to this bit is ignored. Values: ■ 0x0 (NO_AFFECT): No Affect ■ 0x1 (RESTART): Restart	0x0
0	R/W	STOP_BG_REQ	Stop At Block Gap Request This bit is used to stop executing read and write transactions at the next block gap for non-DMA, SDMA, and ADMA transfers. Values: ■ 0x0 (XFER): Transfer ■ 0x1 (STOP): Stop	0x0

WUP_CTRL_R

WUP_CTRL_R is a Wakeup Control Register Offset Address : 0x2b Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
7:3	R	RSVD_7_3	These bits of Wakeup Control register are reserved. They always return 0.	0x0

Bits	Access	Name	Description	Reset
2	R/W	CARD_REMOVAL	Wakeup Event Enable on SD Card Removal This bit enables wakeup event through Card Removal assertion in the Normal Interrupt Status register. For the SDIO card, Wake Up Support (FN_WUS) in the Card Information Structure (CIS) register does not affect this bit. Values: ■ 0x0 (DISABLED): Disable ■ 0x1 (ENABLED): Enable	0x0
1	R/W	CARD_INSERT	Wakeup Event Enable on SD Card Insertion This bit enables wakeup event through Card Insertion assertion in the Normal Interrupt Status register. FN_WUS (Wake Up Support) in CIS does not affect this bit. Values: ■ 0x0 (DISABLED): Disable ■ 0x1 (ENABLED): Enable	0x0
0	R/W	CARD_INT	Wakeup Event Enable on Card Interrupt This bit enables wakeup event through a Card Interrupt assertion in the Normal Interrupt Status register. This bit can be set to 1 if FN_WUS (Wake Up Support) in CIS is set to 1. Values: ■ 0x0 (DISABLED): Disable ■ 0x1 (ENABLED): Enable	0x0

CLK_CTRL_R

CLK_CTRL_R is a Clock Control Register Offset Address : 0x2c Total Reset Value: 0x0

Bits	Access	Name	Description	Reset

Bits	Access	Name	Description	Reset
15:8	R/W	FREQ_SEL	SDCLK/RCLK Frequency Select These bits are used to select the frequency of the SDCLK signal. These bits depend on setting of Preset Value Enable in the Host Control 2 register. If Preset Value Enable = 0, these bits are set by the Host Driver. If Preset Value Enable = 1, these bits are automatically set to a value specified in one of the Preset Value register. The value is reflected on the lower 8-bit of the card_clk_freq_sel signal. 10-bit Divided Clock Mode: ■ 0x3FF : 1/2046 Divided clock ■ ■ N : 1/2N Divided Clock ■ ■ 0x002 : 1/4 Divided Clock ■ 0x001 : 1/2 Divided Clock ■ 0x000 : Base clock (10MHz - 255 MHz) Programmable Clock Mode : Enables the Host System to select a fine grain SD clock frequency: ■ 0x3FF : Base clock $M / 1024$ ■ ■ $N-1$: Base clock M / N ■ ■ 0x002 : Base clock $M / 3$ ■ 0x001 : Base clock $M / 2$ ■ 0x000 : Base clock * M	0x0
7:6	R/W	UPPER_FREQ_SEL	These bits specify the upper 2 bits of 10-bit SDCLK/RCLK Frequency Select control. The value is reflected on the upper 2 bits of the card_clk_freq_sel signal.	0x0
5	R/W	CLK_GEN_SELECT	Clock Generator Select This bit is used to select the clock generator mode in SDCLK/RCLK Frequency Select. If Preset Value Enable = 0, this bit is set by the Host Driver. If Preset Value Enable = 1, this bit is automatically set to a value specified in one of the Preset Value registers. The value is reflected on the card_clk_gen_sel signal. Values: ■ 0x0 (FALSE): Divided Clock Mode ■ 0x1 (TRUE): Programmable Clock Mode	0x0
4	R	RSVD_4	This bit of the CLK_CTRL_R register is reserved. It always returns 0.	0x0

Bits	Access	Name	Description	Reset
3	R/W	PLL_ENABLE	PLL Enable This bit is used to activate the PLL (applicable when Host Version 4 Enable = 1). When Host Version 4 Enable = 0, INTERNAL_CLK_EN bit may be used to activate PLL. The value is reflected on the card_clk_en signal. Note: If this bit is not used to to active the PLL when Host Version 4 Enable = 1, it is recommended to set this bit to '1' . Values: ■ 0x0 (FALSE): PLL is in low power mode ■ 0x1 (TRUE): PLL is enabled	0x0
2	R/W	SD_CLK_EN	SD/eMMC Clock Enable This bit stops the SDCLK or RCLK when set to 0. The SDCLK/RCLK Frequency Select bit can be changed when this bit is set to 0. The value is reflected on the clk2card_on pin. Values: ■ 0x0 (FALSE): Disable providing SDCLK/RCLK ■ 0x1 (TRUE): Enable providing SDCLK/RCLK	0x0
1	R	INTERNAL_CLK_STABLE	Internal Clock Stable This bit enables the Host Driver to check the clock stability twice after the Internal Clock Enable bit is set and after the PLL Enable bit is set. This bit reflects the synchronized value of the intclk_stable signal after the Internal Clock Enable bit is set to 1 and also reflects the synchronized value of the card_clk_stable signal after the PLL Enable bit is set to 1. Values: ■ 0x0 (FALSE): Not Ready ■ 0x1 (TRUE): Ready	0x0
0	R/W	INTERNAL_CLK_EN	Internal Clock Enable This bit is set to 0 when the Host Driver is not using the Host Controller or the Host Controller awaits a wakeup interrupt. The Host Controller must stop its internal clock to enter a very low power state. However, registers can still be read and written to. The value is reflected on the intclk_en signal. Note: If this bit is not used to control the internal clock (base clock and master clock), it is recommended to set this bit to '1' . Values: ■ 0x0 (FALSE): Stop ■ 0x1 (TRUE): Oscillate	0x0

TOUT_CTRL_R

TOUT_CTRL_R is a Timeout Control Register Offset Address : 0x2e Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
7:4	R	RSVD_7_4	These bits of the Timeout Control register are reserved. They always return 0.	0x0
3:0	R/W	TOUT_CNT	Data Timeout Counter Value. This value determines the interval by which DAT line timeouts are detected. The Timeout clock frequency is generated by dividing the base clock TMCLK value by this value. When setting this register, prevent inadvertent timeout events by clearing the Data Timeout Error Status Enable (in the Error Interrupt Status Enable register). The values for these bits are: ■ 0xF : Reserved ■ 0xE : TMCLK x 2²⁷ ■ ■ 0x1 : TMCLK x 2¹⁴ ■ 0x0 : TMCLK x 2¹³ Note: During a boot operating in an eMMC mode, an application must configure the boot data timeout value (approximately 1 sec) in this bit.	0x0

SW_RST_R

SW_RST_R is a Software Reset Register Offset Address : 0x2f Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
7:3	R	RSVD_7_3	These bits of the SW_RST_R register are reserved.They always return 0.	0x0

Bits	Access	Name	Description	Reset
2	R/W	SW_RST_DAT	Software Reset For DAT line This bit is used in SD/eMMC mode and it resets only a part of the data circuit and the DMA circuit is also reset. The following registers and bits are cleared by this bit: ■ Buffer Data Port register ■ Buffer is cleared and initialized. ■ Present state register - Buffer Read Enable - Buffer Write Enable - Read Transfer Active - Write Transfer Active - DAT Line Active - Command Inhibit (DAT) ■ Block Gap Control register - Continue Request - Stop At Block Gap Request ■ Normal Interrupt status register - Buffer Read Ready - Buffer Write Ready - DMA Interrupt - Block Gap Event - Transfer Complete In UHS-II mode, this bit shall be set to 0 Values: ■ 0x0 (FALSE): Work ■ 0x1 (TRUE): Reset	0x0
1	R/W	SW_RST_CMD	Software Reset For CMD line This bit resets only a part of the command circuit to be able to issue a command. It bit is also used to initialize a UHS-II command circuit. This reset is effective only for a command issuing circuit (including response error statuses related to Command Inhibit (CMD) control) and does not affect the data transfer circuit. Host Controller can continue data transfer even after this reset is executed while handling subcommand-response errors. The following registers and bits are cleared by this bit: ■ Present State register : Command Inhibit (CMD) bit ■ Normal Interrupt Status register : Command Complete bit ■ Error Interrupt Status : Response error statuses related to Command Inhibit (CMD) bit Values: ■ 0x0 (FALSE): Work ■ 0x1 (TRUE): Reset	0x0

Bits	Access	Name	Description	Reset
0	R/W	SW_RST_ALL	Software Reset For All This reset affects the entire Host Controller except for the card detection circuit. During its initialization, the Host Driver sets this bit to 1 to reset the Host Controller. All registers are reset except the capabilities register. If this bit is set to 1, the Host Driver must issue reset command and reinitialize the card. Values: ■ 0x0 (FALSE): Work ■ 0x1 (TRUE): Reset	0x0

NORMAL_INT_STAT_R

NORMAL_INT_STAT_R is a Normal Interrupt Status Register Offset Address : 0x30 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
15	R	ERR_INTERRUPT	Error Interrupt If any of the bits in the Error Interrupt Status register are set, then this bit is set. Values: ■ 0x0 (FALSE): No Error ■ 0x1 (TRUE): Error	0x0
14	R/W1C	CQE_EVENT	Command Queuing Event This status is set if Command Queuing/Crypto related event has occurred in eMMC/SD mode. Read CQHCI's CQIS/CRNQIS register for more details. In UHS-II Mode, this bit is irrelevant. Values: ■ 0x0 (FALSE): No Event ■ 0x1 (TRUE): Command Queuing Event is detected	0x0
13	R	FX_EVENT	FX Event This status is set when R[14] of response register is set to 1 and Response Type R1/R5 is set to 0 in Transfer Mode register. This interrupt is used with response check function. Values: ■ 0x0 (FALSE): No Event ■ 0x1 (TRUE): FX Event is detected	0x0
12	R	RE_TUNE_EVENT	Re-tuning Event This bit is set if the Re-Tuning Request changes from 0 to 1. Re-Tuning request is not supported.	0x0
11	R	INT_C	INT_C (Embedded) This bit is set if INT_C is enabled and if INT_C# pin is in low level. The INT_C# pin is not supported.	0x0
10	R	INT_B	INT_B (Embedded) This bit is set if INT_B is enabled and if INT_B# pin is in low level. The INT_B# pin is not supported.	0x0

Bits	Access	Name	Description	Reset
9	R	INT_A	INT_A (Embedded) This bit is set if INT_A is enabled and if INT_A# pin is in low level. The INT_A# pin is not supported.	0x0
8	R	CARD_INTERRUPT	Card Interrupt This bit reflects the synchronized value of: ■ DAT[1] Interrupt Input for SD Mode ■ DAT[2] Interrupt Input for UHS-II Mode Values: ■ 0x0 (FALSE): No Card Interrupt ■ 0x1 (TRUE): Generate Card Interrupt	0x0
7	R/W1C	CARD_REMOVAL	Card Removal This bit is set if the Card Inserted in the Present State register changes from 1 to 0. Values: ■ 0x0 (FALSE): Card state stable or Debouncing ■ 0x1 (TRUE): Card Removed	0x0
6	R/W1C	CARD_INSERTION	Card Insertion This bit is set if the Card Inserted in the Present State register changes from 0 to 1. Values: ■ 0x0 (FALSE): Card state stable or Debouncing ■ 0x1 (TRUE): Card Inserted	0x0
5	R/W1C	BUF_RD_READY	Buffer Read Ready This bit is set if the Buffer Read Enable changes from 0 to 1. Values: ■ 0x0 (FALSE): Not ready to read buffer ■ 0x1 (TRUE): Ready to read buffer	0x0
4	R/W1C	BUF_WR_READY	Buffer Write Ready This bit is set if the Buffer Write Enable changes from 0 to 1. Values: ■ 0x0 (FALSE): Not ready to write buffer ■ 0x1 (TRUE): Ready to write buffer	0x0
3	R/W1C	DMA_INTERRUPT	DMA Interrupt This bit is set if the Host Controller detects the SDMA Buffer Boundary during transfer. In case of ADMA, by setting the Int field in the descriptor table, the Host controller generates this interrupt. This interrupt is not generated after a Transfer Complete. Values: ■ 0x0 (FALSE): No DMA Interrupt ■ 0x1 (TRUE): DMA Interrupt is generated	0x0

Bits	Access	Name	Description	Reset
2	R/W1C	BGAP_EVENT	Block Gap Event This bit is set when both read/write transaction is stopped at block gap due to a Stop at Block Gap Request. Values: ■ 0x0 (FALSE): No Block Gap Event ■ 0x1 (TRUE): Transaction stopped at block gap	0x0
1	R/W1C	XFER_COMPLETE	Transfer Complete This bit is set when a read/write transfer and a command with status busy is completed. Values: ■ 0x0 (FALSE): Not complete ■ 0x1 (TRUE): Command execution is completed	0x0
0	R/W1C	CMD_COMPLETE	Command Complete In an SD/eMMC Mode, this bit is set when the end bit of a response except for Auto CMD12 and Auto CMD23. In UHS II Mode, this bit is set when response packet is received. This interrupt is not generated when the Response Interrupt Disable in Transfer Mode Register is set to 1. Values: ■ 0x0 (FALSE): No command complete ■ 0x1 (TRUE): Command Complete	0x0

ERROR_INT_STAT_R

ERROR_INT_STAT_R is an Error Interrupt Status Register Offset Address : 0x32 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
15	R/W1C	VENDOR_ERR3	This bit (VENDOR_ERR3) of the ERROR_INT_STAT_R register is reserved. It always returns 0.	0x0
14	R/W1C	VENDOR_ERR2	This bit (VENDOR_ERR2) of the ERROR_INT_STAT_R register is reserved. It always returns 0.	0x0
13	R/W1C	VENDOR_ERR1	This bit (VENDOR_ERR1) of the ERROR_INT_STAT_R register is reserved. It always returns 0.	0x0
12	R/W1C	BOOT_ACK_ERR	Boot Acknowledgement Error This bit is set when there is a timeout for boot acknowledgement or when detecting boot ack status having a value other than 010. This is applicable only when boot acknowledgement is expected in eMMC mode. In SD/UHS-II mode, this bit is irrelevant. Values: ■ 0x0 (FALSE): No error ■ 0x1 (TRUE): Error	0x0

Bits	Access	Name	Description	Reset
11	R/W1C	RESP_ERR	Response Error Host Controller Version 4.00 supports response error check function to avoid overhead of response error check by Host Driver during DMA execution. If Response Error Check Enable is set to 1 in the Transfer Mode register, Host Controller Checks R1 or R5 response. If an error is detected in a response, this bit is set to 1. This is applicable in SD/eMMC mode. In UHS-II mode, this bit is irrelevant. Values: ■ 0x0 (FALSE): No error ■ 0x1 (TRUE): Error	0x0
10	R/W1C	TUNING_ERR	Tuning Error This bit is set when an unrecoverable error is detected in a tuning circuit except during the tuning procedure (occurrence of an error during tuning procedure is indicated by Sampling Clock Select in the Host Control 2 register). By detecting Tuning Error, Host Driver needs to abort a command executing and perform tuning. To reset tuning circuit, Sampling Clock Select is set to 0 before executing tuning procedure. The Tuning Error is higher priority than the other error interrupts generated during data transfer. By detecting Tuning Error, the Host Driver must discard data transferred by a current read/write command and retry data transfer after the Host Controller retrieved from the tuning circuit error. This is applicable in SD/eMMC mode. In UHS-II mode, this bit is irrelevant. Values: ■ 0x0 (FALSE): No error ■ 0x1 (TRUE): Error	0x0

Bits	Access	Name	Description	Reset
9	R/W1C	ADMA_ERR	ADMA Error This bit is set when the Host Controller detects error during ADMA-based data transfer. The error could be due to following reasons: ■ Error response received from System bus (Master I/F) ■ ADMA3,ADMA2 Descriptors invalid ■ CQE Task or Transfer descriptors invalid When the error occurs, the state of the ADMA is saved in the ADMA Error Status register. In eMMC CQE mode: The Host Controller generates this Interrupt when it detects an invalid descriptor data (Valid=0) at the ST_FDS state. ADMA Error State in the ADMA Error Status indicates that an error has occurred in ST_FDS state. The Host Driver may find that Valid bit is not set at the error descriptor. Values: ■ 0x0 (FALSE): No error ■ 0x1 (TRUE): Error	0x0
8	R/W1C	AUTO_CMD_ERR	Auto CMD Error This error status is used by Auto CMD12 and Auto CMD23 in SD/eMMC mode. This bit is set when detecting that any of the bits D00 to D05 in Auto CMD Error Status register has changed from 0 to 1. D07 is effective in case of Auto CMD12. Auto CMD Error Status register is valid while this bit is set to 1 and may be cleared by clearing of this bit. In UHS-II mode, this bit is irrelevant. Values: ■ 0x0 (FALSE): No error ■ 0x1 (TRUE): Error	0x0
7	R/W1C	CUR_LMT_ERR	Current Limit Error By setting the SD Bus Power bit in the Power Control register, the Host Controller is requested to supply power for the SD Bus. If the Host Controller supports the Current Limit function, it can be protected from an illegal card by stopping power supply to the card in which case this bit indicates a failure status. A reading of 1 for this bit means that the Host Controller is not supplying power to the SD card due to some failure. A reading of 0 for this bit means that the Host Controller is supplying power and no error has occurred. The Host Controller may require some sampling time to detect the current limit. DWC_mshc Host Controller does not support this function, this bit is always set to 0. Values: ■ 0x0 (FALSE): No error ■ 0x1 (TRUE): Power Fail	0x0

Bits	Access	Name	Description	Reset
6	R/W1C	DATA_END_BIT_ERR	Data End Bit Error This error occurs in SD/eMMC mode either when detecting 0 at the end bit position of read data that uses the DAT line or at the end bit position of the CRC status. In UHS-II mode, this bit is irrelevant. Values: ■ 0x0 (FALSE): No error ■ 0x1 (TRUE): Error	0x0
5	R/W1C	DATA_CRC_ERR	Data CRC Error This error occurs in SD/eMMC mode when detecting CRC error when transferring read data which uses the DAT line, when detecting the Write CRC status having a value of other than 010 or when write CRC status timeout. In UHS-II mode, this bit is irrelevant. Values: ■ 0x0 (FALSE): No error ■ 0x1 (TRUE): Error	0x0
4	R/W1C	DATA_TOUT_ERR	Data Timeout Error This bit is set in SD/eMMC mode when detecting one of the following timeout conditions: ■ Busy timeout for R1b, R5b type ■ Busy timeout after Write CRC status ■ Write CRC Status timeout ■ Read Data timeout In UHS-II mode, this bit is irrelevant. Values: ■ 0x0 (FALSE): No error ■ 0x1 (TRUE): Time out	0x0
3	R/W1C	CMD_IDX_ERR	Command Index Error This bit is set if a Command Index error occurs in the command respons in SD/eMMC mode. In UHS-II mode, this bit is irrelevant. Values: ■ 0x0 (FALSE): No error ■ 0x1 (TRUE): Error	0x0
2	R/W1C	CMD_END_BIT_ERR	Command End Bit Error This bit is set when detecting that the end bit of a command response is 0 in SD/eMMC mode. In UHS-II mode, this bit is irrelevant. Values: ■ 0x0 (FALSE): No error ■ 0x1 (TRUE): End Bit error generated	0x0

Bits	Access	Name	Description	Reset
1	R/W1C	CMD_CRC_ERR	Command CRC Error Command CRC Error is generated in SD/eMMC mode for following two cases. ■ If a response is returned and the Command Timeout Error is set to 0 (indicating no timeout), this bit is set to 1 when detecting a CRC error in the command response. ■ The Host Controller detects a CMD line conflict by monitoring the CMD line when a command is issued. If the Host Controller drives the CMD line to 1 level, but detects 0 level on the CMD line at the next SD clock edge, then the Host Controller aborts the command (stop driving CMD line) and set this bit to 1. The Command Timeout Error is also set to 1 to distinguish a CMD line conflict. In UHS-II mode, this bit is irrelevant. Values: ■ 0x0 (FALSE): No error ■ 0x1 (TRUE): CRC error generated	0x0
0	R/W1C	CMD_TOUT_ERR	Command Timeout Error In SD/eMMC Mode, this bit is set only if no response is returned within 64 SD clock cycles from the end bit of the command. If the Host Controller detects a CMD line conflict, along with Command CRC Error bit, this bit is set to 1, without waiting for 64 SD/eMMC card clock cycles. In UHS-II mode, this bit is irrelevant. Values: ■ 0x0 (FALSE): No error ■ 0x1 (TRUE): Time out	0x0

NORMAL_INT_STAT_EN_R

NORMAL_INT_STAT_EN_R is a Normal Interrupt Status Enable Register Offset Address : 0x34 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
15	R	RSVD_15	This bit of the NORMAL_INT_STAT_EN_R register is reserved. It always returns 0.	0x0
14	R/W	CQE_EVENT_STAT_EN	CQE Event Status Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
13	R/W	FX_EVENT_STAT_EN	FX Event Status Enable This bit is added from Version 4.10. Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0

Bits	Access	Name	Description	Reset
12	R/W	RE_TUNE_EVENT_STAT_EN	Re-Tuning Event (UHS-I only) Status Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
11	R/W	INT_C_STAT_EN	INT_C (Embedded) Status Enable If this bit is set to 0, the Host Controller clears the interrupt request to the System. The Host Driver may clear this bit before servicing the INT_C and may set this bit again after all interrupt requests to INT_C pin are cleared to prevent inadvertent interrupts. Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
10	R/W	INT_B_STAT_EN	INT_B (Embedded) Status Enable If this bit is set to 0, the Host Controller clears the interrupt request to the System. The Host Driver may clear this bit before servicing the INT_B and may set this bit again after all interrupt requests to INT_B pin are cleared to prevent inadvertent interrupts. Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
9	R/W	INT_A_STAT_EN	INT_A (Embedded) Status Enable If this bit is set to 0, the Host Controller clears the interrupt request to the System. The Host Driver may clear this bit before servicing the INT_A and may set this bit again after all interrupt requests to INT_A pin are cleared to prevent inadvertent interrupts. Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
8	R/W	CARD_INTERRUPT_STAT_EN	Card Interrupt Status Enable If this bit is set to 0, the Host Controller clears the interrupt request to the System. The Card Interrupt detection is stopped when this bit is cleared and restarted when this bit is set to 1. The Host Driver may clear the Card Interrupt Status Enable before servicing the Card Interrupt and may set this bit again after all interrupt requests from the card are cleared to prevent inadvertent interrupts. By setting this bit to 0, interrupt input must be masked by implementation so that the interrupt input is not affected by external signal in any state (for example, floating). Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0

Bits	Access	Name	Description	Reset
7	R/W	CARD_REMOVAL_STAT_EN	Card Removal Status Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
6	R/W	CARD_INSERTION_STAT_EN	Card Insertion Status Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
5	R/W	BUF_RD_READY_STAT_EN	Buffer Read Ready Status Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
4	R/W	BUF_WR_READY_STAT_EN	Buffer Write Ready Status Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
3	R/W	DMA_INTERRUPT_STAT_EN	DMA Interrupt Status Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
2	R/W	BGAP_EVENT_STAT_EN	Block Gap Event Status Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
1	R/W	XFER_COMPLETE_STAT_EN	Transfer Complete Status Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
0	R/W	CMD_COMPLETE_STAT_EN	Command Complete Status Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0

ERROR_INT_STAT_EN_R

ERROR_INT_STAT_EN_R is an Error Interrupt Status Enable Register Offset Address : 0x36 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
15	R/W	VENDOR_ERR_STAT_EN3	The 15th bit of Error Interrupt Status Enable register is reserved. Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0

Bits	Access	Name	Description	Reset
14	R/W	VENDOR_ERR_STAT_EN2	The 14th bit of Error Interrupt Status Enable register is reserved. Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
13	R/W	VENDOR_ERR_STAT_EN1	The 13th bit of Error Interrupt Status Enable register is reserved. Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
12	R/W	BOOT_ACK_ERR_STAT_EN	Boot Acknowledgment Error (eMMC Mode only) Setting this bit to 1 enables setting of Boot Acknowledgment Error in Error Interrupt Status register (ERROR_INT_STAT_R). Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
11	R/W	RESP_ERR_STAT_EN	Response Error Status Enable (SD Mode only) Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
10	R/W	TUNING_ERR_STAT_EN	Tuning Error Status Enable (UHS-I Mode only) Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
9	R/W	ADMA_ERR_STAT_EN	ADMA Error Status Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
8	R/W	AUTO_CMD_ERR_STAT_EN	Auto CMD Error Status Enable (SD/eMMC Mode only). Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
7	R/W	CUR_LMT_ERR_STAT_EN	Current Limit Error Status Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
6	R/W	DATA_END_BIT_ERR_STAT_EN	Data End Bit Error Status Enable (SD/eMMC Mode only). Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0

Bits	Access	Name	Description	Reset
5	R/W	DATA_CRC_ERR_STAT_EN	Data CRC Error Status Enable (SD/eMMC Mode only) Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
4	R/W	DATA_TOUT_ERR_STAT_EN	Data Timeout Error Status Enable (SD/eMMC Mode only) Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
3	R/W	CMD_IDX_ERR_STAT_EN	Command Index Error Status Enable (SD/eMMC Mode only) Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
2	R/W	CMD_END_BIT_ERR_STAT_EN	Command End Bit Error Status Enable (SD/eMMC Mode only) Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
1	R/W	CMD_CRC_ERR_STAT_EN	Command CRC Error Status Enable (SD/eMMC Mode only) Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
0	R/W	CMD_TOUT_ERR_STAT_EN	Command Timeout Error Status Enable (SD/eMMC Mode only). Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0

NORMAL_INT_SIGNAL_EN_R

NORMAL_INT_SIGNAL_EN_R is a Normal Interrupt Signal Enable Register Offset Address : 0x38 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
15	R	RSVD_15	This bit of the NORMAL_INT_STAT_R register is reserved. It always returns 0.	0x0
14	R/W	CQE_EVENT_SIGNAL_EN	Command Queuing Engine Event Signal Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0

Bits	Access	Name	Description	Reset
13	R/W	FX_EVENT_SIGNAL_EN	FX Event Signal Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
12	R/W	RE_TUNE_EVENT_SIGNAL_EN	Re-Tuning Event (UHS-I only) Signal Enable. Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
11	R/W	INT_C_SIGNAL_EN	INT_C (Embedded) Signal Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
10	R/W	INT_B_SIGNAL_EN	INT_B (Embedded) Signal Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
9	R/W	INT_A_SIGNAL_EN	INT_A (Embedded) Signal Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
8	R/W	CARD_INTERRUPT_SIGNAL_EN	Card Interrupt Signal Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
7	R/W	CARD_REMOVAL_SIGNAL_EN	Card Removal Signal Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
6	R/W	CARD_INSERTION_SIGNAL_EN	Card Insertion Signal Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
5	R/W	BUF_RD_READY_SIGNAL_EN	Buffer Read Ready Signal Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
4	R/W	BUF_WR_READY_SIGNAL_EN	Buffer Write Ready Signal Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
3	R/W	DMA_INTERRUPT_SIGNAL_EN	DMA Interrupt Signal Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0

Bits	Access	Name	Description	Reset
2	R/W	BGAP_EVENT_SIGNAL_EN	Block Gap Event Signal Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
1	R/W	XFER_COMPLETE_SIGNAL_EN	Transfer Complete Signal Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
0	R/W	CMD_COMPLETE_SIGNAL_EN	Command Complete Signal Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0

ERROR_INT_SIGNAL_EN_R

ERROR_INT_SIGNAL_EN_R is an Error Interrupt Signal Enable Register Offset Address : 0x3a Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
15	R/W	VENDOR_ERR_SIGNAL_EN3	The 16th bit of Error Interrupt Signal Enable is reserved. Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
14	R/W	VENDOR_ERR_SIGNAL_EN2	The 15th bit of Error Interrupt Signal Enable is reserved. Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
13	R/W	VENDOR_ERR_SIGNAL_EN1	The 14th bit of Error Interrupt Signal Enable is reserved. Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
12	R/W	BOOT_ACK_ERR_SIGNAL_EN	Boot Acknowledgment Error (eMMC Mode only). Setting this bit to 1 enables generating interrupt signal when Boot Acknowledgement Error in Error Interrupt Status register is set. Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
11	R/W	RESP_ERR_SIGNAL_EN	Response Error Signal Enable (SD Mode only) Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0

Bits	Access	Name	Description	Reset
10	R/W	TUNING_ERR_SIGNAL_EN	Tuning Error Signal Enable (UHS-I Mode only) Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
9	R/W	ADMA_ERR_SIGNAL_EN	ADMA Error Signal Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
8	R/W	AUTO_CMD_ERR_SIGNAL_EN	Auto CMD Error Signal Enable (SD/eMMC Mode only) Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
7	R/W	CUR_LMT_ERR_SIGNAL_EN	Current Limit Error Signal Enable Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
6	R/W	DATA_END_BIT_ERR_SIGNAL_EN	Data End Bit Error Signal Enable (SD/eMMC Mode only) Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
5	R/W	DATA_CRC_ERR_SIGNAL_EN	Data CRC Error Signal Enable (SD/eMMC Mode only) Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
4	R/W	DATA_TOUT_ERR_SIGNAL_EN	Data Timeout Error Signal Enable (SD/eMMC Mode only) Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
3	R/W	CMD_IDX_ERR_SIGNAL_EN	Command Index Error Signal Enable (SD/eMMC Mode only) Values: ■ 0x0 (FALSE): No error ■ 0x1 (TRUE): Error	0x0
2	R/W	CMD_END_BIT_ERR_SIGNAL_EN	Command End Bit Error Signal Enable (SD/eMMC Mode only) Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0

Bits	Access	Name	Description	Reset
1	R/W	CMD_CRC_ERR_SIGNAL_EN	Command CRC Error Signal Enable (SD/eMMC Mode only) Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0
0	R/W	CMD_TOUT_ERR_SIGNAL_EN	Command Timeout Error Signal Enable (SD/eMMC Mode only) Values: ■ 0x0 (FALSE): Masked ■ 0x1 (TRUE): Enabled	0x0

AUTO_CMD_STAT_R

AUTO_CMD_STAT_R is an Auto CMD Status Register Offset Address : 0x3c Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
15:8	R	RSVD_15_8	These bits of the AUTO_CMD_STAT_R register are reserved bits. They always return 0.	0x0
7	R	CMD_NOT_ISSUED_AUTO_CMD12	Command Not Issued By Auto CMD12 Error If this bit is set to 1, CMD_wo_DAT is not executed due to an Auto CMD12 Error (D04-D01) in this register. This bit is set to 0 when Auto CMD Error is generated by Auto CMD23. Values: ■ 0x1 (TRUE): Not Issued ■ 0x0 (FALSE): No Error	0x0
6	R	RSVD_6	This bit of the AUTO_CMD_STAT_R register is reserved. It always returns 0.	0x0
5	R	AUTO_CMD_RESP_ERR	Auto CMD Response Error This bit is set when Response Error Check Enable in the Transfer Mode register is set to 1 and an error is detected in R1 response of either Auto CMD12 or CMD13. This status is ignored if any bit between D00 to D04 is set to 1. Values: ■ 0x1 (TRUE): Error ■ 0x0 (FALSE): No Error	0x0
4	R	AUTO_CMD_IDX_ERR	Auto CMD Index Error This bit is set if the command index error occurs in response to a command. Values: ■ 0x1 (TRUE): Error ■ 0x0 (FALSE): No Error	0x0

Bits	Access	Name	Description	Reset
3	R	AUTO_CMD_EBIT_ERR	Auto CMD End Bit Error This bit is set when detecting that the end bit of command response is 0. Values: ■ 0x1 (TRUE): End Bit Error Generated ■ 0x0 (FALSE): No Error	0x0
2	R	AUTO_CMD_CRC_ERR	Auto CMD CRC Error This bit is set when detecting a CRC error in the command response. Values: ■ 0x1 (TRUE): CRC Error Generated ■ 0x0 (FALSE): No Error	0x0
1	R	AUTO_CMD_TOUT_ERR	Auto CMD Timeout Error This bit is set if no response is returned with 64 SDCLK cycles from the end bit of the command.If this bit is set to 1, error status bits (D04-D01) are meaningless. Values: ■ 0x1 (TRUE): Time out ■ 0x0 (FALSE): No Error	0x0
0	R	AUTO_CMD12_NOT_EXEC	Auto CMD12 Not Executed If multiple memory block data transfer is not started due to a command error, this bit is not set because it is not necessary to issue an Auto CMD12.Setting this bit to 1 means that the Host Controller cannot issue Auto CMD12 to stop multiple memory block data transfer, due to some error. If this bit is set to 1, error status bits (D04-D01) is meaningless.This bit is set to 0 when Auto CMD Error is generated by Auto CMD23. Values: ■ 0x1 (TRUE): Not Executed ■ 0x0 (FALSE): Executed	0x0

HOST_CTRL2_R

HOST_CTRL2_R is a Host Control 2 Register Offset Address : 0x3e Total Reset Value: 0x0

Bits	Access	Name	Description	Reset

Bits	Access	Name	Description	Reset
15	R/W	PRESET_VAL_ENABLE	Preset Value Enable This bit enables automatic selection of SDCLK frequency and Driver strength Preset Value registers. When Preset Value Enable is set, SDCLK frequency generation (Frequency Select and Clock Generator Select) and the driver strength selection are performed by the controller. These values are selected from set of Preset Value registers based on selected speed mode. Note: For more information, see the FAQ on Preset Register in the DWC_mshc Databook. Values: ■ 0x0 (FALSE): SDCLK and Driver Strength are controlled by Host Driver ■ 0x1 (TRUE): Automatic Selection by Preset Value are Enabled	0x0
14	R/W	ASYNC_INT_ENABLE	Asynchronous Interrupt Enable This bit can be set if a card supports asynchronous interrupts and Asynchronous Interrupt Support is set to 1 in the Capabilities register. Values: ■ 0x0 (FALSE): Disabled ■ 0x1 (TRUE): Enabled	0x0
13	R/W	ADDRESSING	64-bit Addressing This bit is effective when Host Version 4 Enable is set to 1. Values: ■ 0x0 (FALSE): 32 bits addressing ■ 0x1 (TRUE): 64 bits addressing	0x0

Bits	Access	Name	Description	Reset
12	R/W	HOST_VER4_ENABLE	Host Version 4 Enable This bit selects either Version 3.00 compatible mode or Version 4 mode. Functions of following fields are modified for Host Version 4 mode: ■ SDMA Address: SDMA uses ADMA System Address (05Fh-058h) instead of SDMA System Address register (003h-000h) ■ ADMA2/ADMA3 selection: ADMA3 is selected by DMA select in Host Control 1 register ■ 64-bit ADMA Descriptor Size: 128-bit descriptor is used instead of 96-bit descriptor when 64-bit Addressing is set to 1 ■ Selection of 32-bit/64-bit System Addressing: Either 32-bit or 64-bit system addressing is selected by 64-bit Addressing bit in this register ■ 32-bit Block Count: SDMA System Address register (003h-000h) is modified to 32-bit Block Count register Note: It is recommended not to program ADMA3 Integrated Descriptor Address registers, UHS-II registers and Command Queuing registers (if applicable) while operating in Host version less than 4 mode (Host Version 4 Enable = 0). Values: ■ 0x0 (FALSE): Version 3.00 compatible mode ■ 0x1 (TRUE): Version 4 mode	0x0
11	R/W	CMD23_ENABLE	CMD23 Enable If the card supports CMD23, this bit is set to 1. This bit is used to select Auto CMD23 or Auto CMD12 for ADMA3 data transfer. Values: ■ 0x0 (FALSE): Auto CMD23 is disabled ■ 0x1 (TRUE): Auto CMD23 is enabled	0x0
10	R/W	ADMA2_LEN_MODE	ADMA2 Length Mode This bit selects ADMA2 Length mode to be either 16-bit or 26-bit. Values: ■ 0x0 (FALSE): 16-bit Data Length Mode ■ 0x1 (TRUE): 26-bit Data Length Mode	0x0
9	R	RSVD_9	This bit of the HOST_CTRL2_R register is reserved. It always returns 0.	0x0
8	R/W	UHS2_IF_ENABLE	UHS-II Interface Enable This bit is used to enable the UHS-II Interface. The value is reflected on the uhs2_if_en pin. Values: ■ 0x0 (FALSE): SD/eMMC Interface Enabled ■ 0x1 (TRUE): UHS-II Interface Enabled	0x0

Bits	Access	Name	Description	Reset
7	R/W	SAMPLE_CLK_SEL	Sampling Clock Select This bit is used by the Host Controller to select the sampling clock in SD/eMMC mode to receive CMD and DAT. This bit is set by the tuning procedure and is valid after the completion of tuning (when Execute Tuning is cleared). Setting this bit to 1 means that tuning is completed successfully and setting this bit to 0 means that tuning has failed. The value is reflected on the sample_cclk_sel pin. This bit is irrelevant in UHS-II mode Values: ■ 0x0 (FALSE): Fixed clock is used to sample data ■ 0x1 (TRUE): Tuned clock is used to sample data	0x0
6	R/W	EXEC_TUNING	Execute Tuning This bit is set to 1 to start the tuning procedure in UHS I/eMMC speed modes and this bit is automatically cleared when tuning procedure is completed. This bit is irrelevant in UHS-II mode Values: ■ 0x0 (FALSE): Not Tuned or Tuning completed ■ 0x1 (TRUE): Execute Tuning	0x0
5:4	R/W	DRV_STRENGTH_SEL	Driver Strength Select This bit is used to select the Host Controller output driver in 1.8V signaling UHS-I/eMMC speed modes. The bit depends on setting of Preset Value Enable. The value is reflected on the uhs1_drv_sth pin. This bit is irrelevant in UHS-II mode Values: ■ 0x0 (TYPEB): Driver TYPEB is selected ■ 0x1 (TYPEA): Driver TYPEA is selected ■ 0x2 (TYPEC): Driver TYPEC is selected ■ 0x3 (TYPED): Driver TYPED is selected	0x0
3	R/W	SIGNALING_EN	1.8V Signaling Enable This bit controls voltage regulator for I/O cell in UHS-I/eMMC speed modes. Setting this bit from 0 to 1 starts changing the signal voltage from 3.3V to 1.8V. Host Controller clears this bit if switching to 1.8 signaling fails. The value is reflected on the uhs1_svwolt_en pin. This bit shall be set to 0 in UHS-II mode Note: This bit must be set for all UHS-I speed modes (SDR12/SDR25/SDR50/SDR104/DDR50). Values: ■ 0x0 (V_3_3): 3.3V Signalling ■ 0x1 (V_1_8): 1.8V Signalling	0x0

Bits	Access	Name	Description	Reset
2:0	R/W	UHS_MODE_SEL	UHS Mode/eMMC Speed Mode Select These bits are used to select UHS mode in the SD mode of operation. In eMMC mode, these bits are used to select eMMC Speed mode. UHS Mode (SD/UHS-II mode only): ■ 0x0: SDR12 ■ 0x1: SDR25 ■ 0x2: SDR50 ■ 0x3: SDR104 ■ 0x4: DDR50 ■ 0x5: Reserved ■ 0x6: Reserved ■ 0x7: UHS-II eMMC Speed Mode (eMMC mode only): ■ 0x0: Legacy ■ 0x1: High Speed SDR ■ 0x2: Reserved ■ 0x3: HS200 ■ 0x4: High Speed DDR ■ 0x5: Reserved ■ 0x6: Reserved ■ 0x7: HS400 Values: ■ 0x0 (SDR12): SDR12/Legacy ■ 0x1 (SDR25): SDR25/High Speed SDR ■ 0x2 (SDR50): SDR50 ■ 0x3 (SDR104): SDR104/HS200 ■ 0x4 (DDR50): DDR50/High Speed DDR ■ 0x5 (RSVD5): Reserved ■ 0x6 (RSVD6): Reserved ■ 0x7 (UHS2): UHS-II/HS400	0x0

CAPABILITIES1_R

CAPABILITIES1_R is a Capabilities 1 Register - 0 to 31 Offset Address : 0x40 Total Reset Value: 0x256ec881

Bits	Access	Name	Description	Reset
31:30	R	SLOT_TYPE_R	Slot Type These bits indicate usage of a slot by a specific Host System. Values: ■ 0x0 (REMOVABLE_SLOT): Removable Card Slot ■ 0x1 (EMBEDDED_SLOT): Embedded Slot for one Device ■ 0x2 (SHARED_SLOT): Shared Bus Slot (SD mode) ■ 0x3 (UHS2_EMBEDDED_SLOT): UHS-II Multiple Embedded Devices	0x0

Bits	Access	Name	Description	Reset
29	R	ASYNC_INT_SUPPORT	Asynchronous Interrupt Support (SD Mode only) Values: ■ 0x0 (FALSE): Asynchronous Interrupt Not Supported ■ 0x1 (TRUE): Asynchronous Interrupt Supported	0x1
28	R	SYS_ADDR_64_V3	64-bit System Address Support for V3 This bit sets the Host controller to support 64-bit System Addressing of V3 mode. SDMA cannot be used in 64-bit Addressing in Version 3 Mode. If this bit is set to 1, 64-bit ADMA2 with using 96-bit Descriptor can be enabled by setting Host Version 4 Enable (HOST_VER4_ENABLE = 0) and DMA select (DMA_SEL = 11b). Values: ■ 0x0 (FALSE): 64-bit System Address for V3 is Not Supported ■ 0x1 (TRUE): 64-bit System Address for V3 is Supported	0x0
27	R	SYS_ADDR_64_V4	64-bit System Address Support for V4 This bit sets the Host Controller to support 64-bit System Addressing of V4 mode. When this bit is set to 1, full or part of 64-bit address must be used to decode the Host Controller Registers so that Host Controller Registers can be placed above system memory area. 64-bit address decode of Host Controller registers is effective regardless of setting to 64-bit Addressing in Host Control 2. If this bit is set to 1, 64-bit DMA Addressing for version 4 is enabled by setting Host Version 4 Enable (HOST_VER4_ENABLE = 1) and by setting 64-bit Addressing (ADDRESSING = 1) in the Host Control 2 register. SDMA can be used and ADMA2 uses 128-bit Descriptor. Values: ■ 0x0 (FALSE): 64-bit System Address for V4 is Not Supported ■ 0x1 (TRUE): 64-bit System Address for V4 is Supported	0x0
26	R	VOLT_18	Voltage Support for 1.8V Values: ■ 0x0 (FALSE): 1.8V Not Supported ■ 0x1 (TRUE): 1.8V Supported	0x1
25	R	VOLT_30	Voltage Support for SD 3.0V or Embedded 1.2V Values: ■ 0x0 (FALSE): SD 3.0V or Embedded 1.2V Not Supported ■ 0x1 (TRUE): SD 3.0V or Embedded Supported	0x0

Bits	Access	Name	Description	Reset
24	R	VOLT_33	Voltage Support for 3.3V Values: ■ 0x0 (FALSE): 3.3V Not Supported ■ 0x1 (TRUE): 3.3V Supported	0x1
23	R	SUS_RES_SUPPORT	Suspend/Resume Support This bit indicates whether the Host Controller supports Suspend/Resume functionality. If this bit is 0, the Host Driver does not issue either Suspend or Resume commands because the Suspend and Resume mechanism is not supported. Values: ■ 0x0 (FALSE): Not Supported ■ 0x1 (TRUE):Supported	0x0
22	R	SDMA_SUPPORT	SDMA Support This bit indicates whether the Host Controller is capable of using SDMA to transfer data between the system memory and the Host Controller directly. Values: ■ 0x0 (FALSE): SDMA not Supported ■ 0x1 (TRUE): SDMA Supported	0x1
21	R	HIGH_SPEED_SUPPORT	High Speed Support This bit indicates whether the Host Controller and the Host System supports High Speed mode and they can supply the SD Clock frequency from 25 MHz to 50 MHz. Values: ■ 0x0 (FALSE): High Speed not Supported ■ 0x1 (TRUE): High Speed Supported	0x1
20	R	RSVD_20	This bit of the CAPABILITIES1_R is a reserved. It always returns 0.	0x0
19	R	ADMA2_SUPPORT	ADMA2 Support This bit indicates whether the Host Controller is capable of using ADMA2. Values: ■ 0x0 (FALSE): ADMA2 not Supported ■ 0x1 (TRUE): ADMA2 Supported	0x1
18	R	Embedded_8_BIT	8-bit Support for Embedded Device This bit indicates whether the Host Controller is capable of using an 8-bit bus width mode. This bit is not effective when the Slot Type is set to 10b. Values: ■ 0x0 (FALSE): 8-bit Bus Width not Supported ■ 0x1 (TRUE): 8-bit Bus Width Supported	0x1

Bits	Access	Name	Description	Reset
17:16	R	MAX_BLK_LEN	Maximum Block Length This bit indicates the maximum block size that the Host driver can read and write to the buffer in the Host Controller. The buffer transfers this block size without wait cycles. The transfer block length is always 512 bytes for the SD Memory irrespective of this bit Values: ■ 0x0 (ZERO): 512 Byte ■ 0x1 (ONE): 1024 Byte ■ 0x2 (TWO): 2048 Byte ■ 0x3 (THREE): Reserved	0x2
15:8	R	BASE_CLK_FREQ	Base Clock Frequency for SD clock These bits indicate the base (maximum) clock frequency for the SD Clock. The definition of these bits depend on the Host Controller Version. ■ 6-Bit Base Clock Frequency: This mode is supported by the Host Controller version 1.00 and 2.00. The upper 2 bits are not effective and are always 0. The unit values are 1 MHz. The supported clock range is 10 MHz to 63 MHz. - 0x00 : Get information through another method - 0x01 : 1 MHz - 0x02 : 2 MHz - - 0x3F : 63 MHz - 0x40-0xFF : Not Supported ■ 8-Bit Base Clock Frequency: This mode is supported by the Host Controller version 3.00. The unit values are 1 MHz. The supported clock range is 10 MHz to 255 MHz. - 0x00 : Get information through another method - 0x01 : 1 MHz - 0x02 : 2 MHz - - 0xFF : 255 MHz If the frequency is 16.5 MHz, the larger value is set to 0001001b (17 MHz) because the Host Driver uses this value to calculate the clock divider value and it does not exceed the upper limit of the SD Clock frequency. If these bits are all 0, the Host system has to get information using a different method.	200
7	R	TOUT_CLK_UNIT	Timeout Clock Unit This bit shows the unit of base clock frequency used to detect Data Timeout Error. Values: ■ 0x0 (KHZ): KHz ■ 0x1 (MHZ): MHz	0x1
6	R	RSVD_6	This bit of the CAPABILITIES1_R register is reserved.	0x0

Bits	Access	Name	Description	Reset
5:0	R	TOUT_CLK_FREQ	Timeout Clock Frequency This bit shows the base clock frequency used to detect Data Timeout Error. The Timeout Clock unit defines the unit of timeout clock frequency. It can be KHz or MHz. ■ 0x00 : Get information through another method ■ 0x01 : 1KHz / 1MHz ■ 0x02 : 2KHz / 2MHz ■ 0x03 : 3KHz / 3MHz ■ ■ 0x3F : 63KHz / 63MHz	0x1

CAPABILITIES2_R

CAPABILITIES2_R is a Capabilities Register - 32 to 63 Offset Address : 0x44 Total Reset Value: 0x08005177

Bits	Access	Name	Description	Reset
31:30	R	RSVD_62_63	These bits (RSVD_62_63) of the CAPABILITIES2_R register are reserved bits. They always return 0.	0x0
29	R	RSVD_61	This bit (RSVD_61) of the CAPABILITIES2_R register is reserved. It always returns 0.	0x0
28	R	VDD2_18V_SUPPORT	1.8V VDD2 Support This bit indicates support of VDD2 for the Host System. Values: ■ 0x0 (FALSE): 1.8V VDD2 is not Supported ■ 0x1 (TRUE): 1.8V VDD2 is Supported	0x0
27	R	ADMA3_SUPPORT	ADMA3 Support This bit indicates whether the Host Controller is capable of using ADMA3. Values: ■ 0x0 (FALSE): ADMA3 not Supported ■ 0x1 (TRUE): ADMA3 Supported	0x1
26:24	R	RSVD_56_58	These bits (RSVD_56_58) of the CAPABILITIES2_R register are reserved. They always return 0.	0x0
23:16	R	CLK_MUL	Clock Multiplier These bits indicate the clock multiplier of the programmable clock generator. Setting these bits to 0 means that the Host Controller does not support a programmable clock generator. ■ 0x0: Clock Multiplier is not Supported ■ 0x1: Clock Multiplier M = 2 ■ 0x2: Clock Multiplier M = 3 ■ ■ 0xFF: Clock Multiplier M = 256	0x0

Bits	Access	Name	Description	Reset
15:14	R	RE_TUNING_MODES	Re-Tuning Modes (UHS-I only) These bits select the re-tuning method and limit the maximum data length. Values: ■ 0x0 (MODE1): Timer ■ 0x1 (MODE2): Timer and Re-Tuning Request (Not supported) ■ 0x2 (MODE3): Auto Re-Tuning (for transfer) ■ 0x3 (RSVD_MODE): Reserved	0x2
13	R	USE_TUNING_SDR50	Use Tuning for SDR50 (UHS-I only) Values: ■ 0x0 (ZERO): SDR50 does not require tuning ■ 0x1 (ONE): SDR50 requires tuning	0x1
12	R	RSVD_44	This bit (RSVD_44) of the CAPABILITIES2_R register is reserved. It always returns 0.	0x0
11:8	R	RETUNE_CNT	Timer Count for Re-Tuning (UHS-I only) ■ 0x0: Re-Tuning Timer disabled ■ 0x1: 1 seconds ■ 0x2: 2 seconds ■ 0x3: 4 seconds ■ ■ 0xB: 1024 seconds ■ 0xC: Reserved ■ 0xD: Reserved ■ 0xE: Reserved ■ 0xF: Get information from other source	0x1
7	R	RSVD_39	This bit (RSVD_39) of the CAPABILITIES2_R register is reserved. It always returns 0.	0x0
6	R	DRV_TYPED	Driver Type D Support (UHS-I only) This bit indicates support of Driver Type D for 1.8 Signaling. Values: ■ 0x0 (FALSE): Driver Type D is not supported ■ 0x1 (TRUE): Driver Type D is supported	0x1
5	R	DRV_TYPEC	Driver Type C Support (UHS-I only) This bit indicates support of Driver Type C for 1.8 Signaling. Values: ■ 0x0 (FALSE): Driver Type C is not supported ■ 0x1 (TRUE): Driver Type C is supported	0x1
4	R	DRV_TYPEA	Driver Type A Support (UHS-I only) This bit indicates support of Driver Type A for 1.8 Signaling. Values: ■ 0x0 (FALSE): Driver Type A is not supported ■ 0x1 (TRUE): Driver Type A is supported	0x1

Bits	Access	Name	Description	Reset
3	R	UHS2_SUPPORT	UHS-II Support (UHS-II only) This bit indicates whether Host Controller supports UHS-II. Values: ■ 0x0 (FALSE): UHS-II is not supported ■ 0x1 (TRUE): UHS-II is supported	0x0
2	R	DDR50_SUPPORT	DDR50 Support (UHS-I only) Values: ■ 0x0 (FALSE): DDR50 is not supported ■ 0x1 (TRUE): DDR50 is supported	0x1
1	R	SDR104_SUPPORT	SDR104 Support (UHS-I only) This bit mentions that SDR104 requires tuning. Values: ■ 0x0 (FALSE): SDR104 is not supported ■ 0x1 (TRUE): SDR104 is supported	0x1
0	R	SDR50_SUPPORT	SDR50 Support (UHS-I only) This bit indicates that SDR50 is supported. The bit 13 (USE_TUNING_SDR50) indicates whether SDR50 requires tuning or not. Values: ■ 0x0 (FALSE): SDR50 is not supported ■ 0x1 (TRUE): SDR50 is supported	0x1

CURR_CAPABILITIES1_R

CURR_CAPABILITIES1_R is a Maximum Current Capabilities Register - 0 to 31 Offset Address : 0x48 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
31:24	R	RSVD_31_24	These bits of the CURR_CAPABILITIES1_R register are reserved. They always return 0.	0x0
23:16	R	MAX_CUR_18V	Maximum Current for 1.8V This bit specifies the Maximum Current for 1.8V VDD1 power supply for the card. ■ 0: Get information through another method ■ 1: 4mA ■ 2: 8mA ■ 3: 13mA ■ ■ 255: 1020mA	0x0

Bits	Access	Name	Description	Reset
15:8	R	MAX_CUR_30V	Maximum Current for 3.0V This bit specifies the Maximum Current for 3.0V VDD1 power supply for the card. ■ 0: Get information through another method ■ 1: 4mA ■ 2: 8mA ■ 3: ■ 255: 1020mA	0x0
7:0	R	MAX_CUR_33V	Maximum Current for 3.3V This bit specifies the Maximum Current for 3.3V VDD1 power supply for the card. ■ 0: Get information through another method ■ 1: 4mA ■ 2: 8mA ■ 3: 13mA ■ ■ 255: 1020mA	0x0

CURR_CAPABILITIES2_R

CURR_CAPABILITIES2_R is a Maximum Current Capabilities Register - 32 to 63 Offset Address : 0x4c Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
31:8	R	RSVD_63_40	These bits of the CURR_CAPABILITIES2_R register are reserved. They always return 0. Reserved Field: Yes	0x0
7:0	R	MAX_CUR_VDD2_18V	Maximum Current for 1.8V VDD2 This bit specifies the Maximum Current for 1.8V VDD2 power supply for the UHS-II card. ■ 0: Get information through another method ■ 1: 4mA ■ 2: 8mA ■ 3: 13mA ■ ■ 255: 1020mA	0x0

FORCE_AUTO_CMD_STAT_R

FORCE_AUTO_CMD_STAT_R is a Force Event Register for Auto CMD Error Status register Offset Address : 0x50
Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
15:8	R	RSVD_15_8	These bits of the FORCE_AUTO_CMD_STAT_R register are reserved. They always return 0.	0x0

Bits	Access	Name	Description	Reset
7	W	FORCE_CMD_NOT_ISSUED_AUTO_CMD12	Force Event for Command Not Issued By Auto CMD12 Error Values: ■ 0x1 (TRUE): Command Not Issued By Auto CMD12 Error Status is set ■ 0x0 (FALSE): Not Affected	0x0
6	R	RSVD_6	This bit of the FORCE_AUTO_CMD_STAT_R register are reserved. They always return 0.	0x0
5	W	FORCE_AUTO_CMD_RESP_ERR	Force Event for Auto CMD Response Error Values: ■ 0x1 (TRUE): Auto CMD Response Error Status is set ■ 0x0 (FALSE): Not Affected	0x0
4	W	FORCE_AUTO_CMD_IDX_ERR	Force Event for Auto CMD Index Error Values: ■ 0x1 (TRUE): Auto CMD Index Error Status is set ■ 0x0 (FALSE): Not Affected	0x0
3	W	FORCE_AUTO_CMD_EBIT_ERR	Force Event for Auto CMD End Bit Error Values: ■ 0x1 (TRUE): Auto CMD End Bit Error Status is set ■ 0x0 (FALSE): Not Affected	0x0
2	W	FORCE_AUTO_CMD_CRC_ERR	Force Event for Auto CMD CRC Error Values: ■ 0x1 (TRUE): Auto CMD CRC Error Status is set ■ 0x0 (FALSE): Not Affected	0x0
1	W	FORCE_AUTO_CMD_TOUT_ERR	Force Event for Auto CMD Timeout Error Values: ■ 0x1 (TRUE): Auto CMD Timeout Error Status is set ■ 0x0 (FALSE): Not Affected	0x0
0	W	FORCE_AUTO_CMD12_NOT_EXEC	Force Event for Auto CMD12 Not Executed Values: ■ 0x1 (TRUE): Auto CMD12 Not Executed Status is set ■ 0x0 (FALSE): Not Affected	0x0

FORCE_ERROR_INT_STAT_R

FORCE_ERROR_INT_STAT_R is a Force Event Register for Error Interrupt Status Offset Address : 0x52 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
15	W	FORCE_VENDOR_ERR3	This bit (FORCE_VENDOR_ERR3) of the FORCE_ERROR_INT_STAT_R register is reserved. It always returns 0.	0x0
14	W	FORCE_VENDOR_ERR2	This bit (FORCE_VENDOR_ERR2) of the FORCE_ERROR_INT_STAT_R register is reserved. It always returns 0.	0x0
13	W	FORCE_VENDOR_ERR1	This bit (FORCE_VENDOR_ERR1) of the FORCE_ERROR_INT_STAT_R register is reserved. It always returns 0.	0x0
12	W	FORCE_BOOT_ACK_ERR	Force Event for Boot Ack error Values: ■ 0x0 (FALSE): Not Affected ■ 0x1 (TRUE): Boot ack Error Status is set	0x0
11	W	FORCE_RESP_ERR	Force Event for Response Error (SD Mode only) Values: ■ 0x0 (FALSE): Not Affected ■ 0x1 (TRUE): Response Error Status is set	0x0
10	W	FORCE_TUNING_ERR	Force Event for Tuning Error (UHS-I Mode only) Values: ■ 0x0 (FALSE): Not Affected ■ 0x1 (TRUE): Tuning Error Status is set	0x0
9	W	FORCE_ADMA_ERR	Force Event for ADMA Error Values: ■ 0x0 (FALSE): Not Affected ■ 0x1 (TRUE): ADMA Error Status is set	0x0
8	W	FORCE_AUTO_CMD_ERR	Force Event for Auto CMD Error (SD/eMMC Mode only) Values: ■ 0x0 (FALSE): Not Affected ■ 0x1 (TRUE): Auto CMD Error Status is set	0x0
7	W	FORCE_CUR_LMT_ERR	Force Event for Current Limit Error Values: ■ 0x0 (FALSE): Not Affected ■ 0x1 (TRUE): Current Limit Error Status is set	0x0
6	W	FORCE_DATA_END_BIT_ERR	Force Event for Data End Bit Error (SD/eMMC Mode only) Values: ■ 0x0 (FALSE): Not Affected ■ 0x1 (TRUE): Data End Bit Error Status is set	0x0

Bits	Access	Name	Description	Reset
5	W	FORCE_DATA_CRC_ERR	Force Event for Data CRC Error (SD/eMMC Mode only) Values: ■ 0x0 (FALSE): Not Affected ■ 0x1 (TRUE): Data CRC Error Status is set	0x0
4	W	FORCE_DATA_TOUT_ERR	Force Event for Data Timeout Error (SD/eMMC Mode only) Values: ■ 0x0 (FALSE): Not Affected ■ 0x1 (TRUE): Data Timeout Error Status is set	0x0
3	W	FORCE_CMD_IDX_ERR	Force Event for Command Index Error (SD/eMMC Mode only) Values: ■ 0x0 (FALSE): Not Affected ■ 0x1 (TRUE): Command Index Error Status is set	0x0
2	W	FORCE_CMD_END_BIT_ERR	Force Event for Command End Bit Error (SD/eMMC Mode only) Values: ■ 0x0 (FALSE): Not Affected ■ 0x1 (TRUE): Command End Bit Error Status is set	0x0
1	W	FORCE_CMD_CRC_ERR	Force Event for Command CRC Error (SD/eMMC Mode only) Values: ■ 0x0 (FALSE): Not Affected ■ 0x1 (TRUE): Command CRC Error Status is set	0x0
0	W	FORCE_CMD_TOUT_ERR	Force Event for Command Timeout Error (SD/eMMC Mode only) Values: ■ 0x0 (FALSE): Not Affected ■ 0x1 (TRUE): Command Timeout Error Status is set	0x0

ADMA_ERR_STAT_R

ADMA_ERR_STAT_R is an ADMA Error Status Register Offset Address : 0x54 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
7:3	R	RSVD_7_3	These bits of the ADMA_ERR_STAT_R register are reserved. They always return 0.	0x0

Bits	Access	Name	Description	Reset
2	R	ADMA_LEN_ERR	ADMA Length Mismatch Error States This error occurs in the following instances: ■ While the Block Count Enable is being set, the total data length specified by the Descriptor table is different from that specified by the Block Count and Block Length ■ When the total data length cannot be divided by the block length Values: ■ 0x0 (NO_ERR): No Error ■ 0x1 (ERROR): Error	0x0
1:0	R	ADMA_ERR_STATES	ADMA Error States These bits indicate the state of ADMA when an error occurs during ADMA data transfer. Values: ■ 0x0 (ST_STOP): Stop DMA - SYS_ADR register points to a location next to the error descriptor ■ 0x1 (ST_FDS): Fetch Descriptor - SYS_ADR register points to the error descriptor ■ 0x2 (UNUSED): Never set this state ■ 0x3 (ST_TFR): Transfer Data - SYS_ADR register points to a location next to the error descriptor	0x0

ADMA_SA_LOW_R

ADMA_SA_LOW_R is an ADMA System Address Register - Low Offset Address : 0x58 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
31:0	R/W	ADMA_SA_LOW	ADMA System Address These bits indicate the lower 32 bits of the ADMA system address. ■ SDMA: If Host Version 4 Enable is set to 1, this register stores the system address of the data location ■ ADMA2: This register stores the byte address of the executing command of the descriptor table ■ ADMA3: This register is set by ADMA3. ADMA2 increments the address of this register that points to the next line, every time a Descriptor line is fetched.	0x0

[ADMA_SA_HIGH_R](#) ADMA_SA_HIGH_R is an ADMA System Address Register - High Offset Address : 0x5c Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
31:0	R/W	ADMA_SA_HIGH	ADMA System Address These bits indicate the higher 32-bit of the ADMA system address.	0x0

ARGUMENT_RPRESET_INIT_R

PRESET_INIT_R is a Preset Value for Initialization Offset Address : 0x60 Total Reset Value: 0x00fa

Bits	Access	Name	Description	Reset
15:14	R	DRV_SEL_VAL	Driver Strength Select Value These bits indicate that the Driver strength is supported by 1.8V signaling bus speed modes. These bits are meaningless for 3.3V signaling. Values: ■ 0x0 (TYPEB): Driver Type B is selected ■ 0x1 (TYPEA): Driver Type A is selected ■ 0x2 (TYPEC): Driver Type C is selected ■ 0x3 (TYPED): Driver Type D is selected	0x0
13:11	R	RSVD_13_11	These bits of the PRESET_INIT_R register are reserved. They always return 0.	0x0
10	R	CLK_GEN_SEL_VAL	Clock Generator Select Value This bit is effective when the Host Controller supports a programmable clock generator. Values: ■ 0x0 (FALSE): Host Controller Ver2.0 Compatible Clock Generator ■ 0x1 (PROG): Programmable Clock Generator	0x0
9:0	R	FREQ_SEL_VAL	SDCLK/RCLK Frequency Select Value 10-bit preset value to be set in SDCLK/RCLK Frequency Select field of the Clock Control register described by a Host System.	250

PRESET_DS_R

PRESET_DS_R is a Preset Value for Default Speed Offset Address : 0x62 Total Reset Value: 0x0004

Bits	Access	Name	Description	Reset
15:14	R	DRV_SEL_VAL	Driver Strength Select Value These bits indicate the Driver strength value supported by 1.8V signaling bus speed modes. This field is meaningless for the Default speed mode as it uses 3.3V signaling. Values: ■ 0x0 (TYPEB): Driver Type B is selected ■ 0x1 (TYPEA): Driver Type A is selected ■ 0x2 (TYPEC): Driver Type C is selected ■ 0x3 (TYPED): Driver Type D is selected	0x0
13:11	R	RSVD_13_11	These bits of the PRESET_DS_R register are reserved. They always return 0.	0x0
10	R	CLK_GEN_SEL_VAL	Clock Generator Select Value This bit is effective when Host Controller supports programmable clock generator. Values: ■ 0x0 (FALSE): Host Controller Ver2.0 Compatible Clock Generator ■ 0x1 (PROG): Programmable Clock Generator	0x0

Bits	Access	Name	Description	Reset
9:0	R	FREQ_SEL_VAL	SDCLK/RCLK Frequency Select Value 10-bit preset value to be set in SDCLK/RCLK Frequency Select field of the Clock Control register described by a Host System.	0x4

PRESET_HS_R

PRESET_HS_R is a Preset Value for High Speed Offset Address : 0x64 Total Reset Value: 0x0002

Bits	Access	Name	Description	Reset
15:14	R	DRV_SEL_VAL	Driver Strength Select Value These bits indicate the Driver strength value supported by 1.8V signaling bus speed modes. This field is meaningless for High speed mode as it uses 3.3V signaling. Values: ■ 0x0 (TYPEB): Driver Type B is selected ■ 0x1 (TYPEA): Driver Type A is selected ■ 0x2 (TYPEC): Driver Type C is selected ■ 0x3 (TYPED): Driver Type D is selected	0x0
13:11	R	RSVD_13_11	These bits of the PRESET_HS_R register are reserved. They always return 0.	0x0
10	R	CLK_GEN_SEL_VAL	Clock Generator Select Value This bit is effective when Host Controller supports programmable clock generator. Values: ■ 0x0 (FALSE): Host Controller Ver2.0 Compatible Clock Generator ■ 0x1 (PROG): Programmable Clock Generator	0x0
9:0	R	FREQ_SEL_VAL	SDCLK/RCLK Frequency Select Value 10-bit preset value to be set in SDCLK/RCLK Frequency Select field of the Clock Control register described by a Host System.	0x2

PRESET_SDR12_R

PRESET_SDR12_R is a Preset Value for SDR12 Offset Address : 0x66 Total Reset Value: 0x0004

Bits	Access	Name	Description	Reset
15:14	R	DRV_SEL_VAL	Driver Strength Select Value These bits indicate the Driver strength value supported for the SDR12 bus speed mode. These bits are meaningless for 3.3V signaling. Values: ■ 0x0 (TYPEB): Driver Type B is selected ■ 0x1 (TYPEA): Driver Type A is selected ■ 0x2 (TYPEC): Driver Type C is selected ■ 0x3 (TYPED): Driver Type D is selected	0x0

Bits	Access	Name	Description	Reset
13:11	R	RSVD_13_11	These bits of the PRESET_SDR12_R register are reserved. They always return 0.	0x0
10	R	CLK_GEN_SEL_VAL	Clock Generator Select Value This bit is effective when Host Controller supports programmable clock generator Values: ■ 0x0 (FALSE): Host Controller Ver2.0 Compatible Clock Generator ■ 0x1 (PROG): Programmable Clock Generator	0x0
9:0	R	FREQ_SEL_VAL	SDCLK/RCLK Frequency Select Value 10-bit preset value to be set in SDCLK/RCLK Frequency Select field of the Clock Control register described by a Host System.	0x4

PRESET_SDR25_R

PRESET_SDR25_R is a Preset Value for SDR25 Offset Address : 0x68 Total Reset Value: 0x0002

Bits	Access	Name	Description	Reset
15:14	R	DRV_SEL_VAL	Driver Strength Select Value These bits indicate the Driver strength value supported for the SDR25 bus speed mode. These bits are meaningless for 3.3V signaling. Values: ■ 0x0 (TYPEB): Driver Type B is selected ■ 0x1 (TYPEA): Driver Type A is selected ■ 0x2 (TYPEC): Driver Type C is selected ■ 0x3 (TYPED): Driver Type D is selected	0x0
13:11	R	RSVD_13_11	These bits of the PRESET_SDR25_R register are reserved. They always return 0.	0x0
10	R	CLK_GEN_SEL_VAL	Clock Generator Select Value This bit is effective when Host Controller supports programmable clock generator. Values: ■ 0x0 (FALSE): Host Controller Ver2.0 Compatible Clock Generator ■ 0x1 (PROG): Programmable Clock Generator	0x0
9:0	R	FREQ_SEL_VAL	SDCLK/RCLK Frequency Select Value 10-bit preset value to be set in SDCLK/RCLK Frequency Select field of the Clock Control register described by a Host System.	0x2

PRESET_SDR50_R

PRESET_SDR50_R is a Preset Value for SDR50 Offset Address : 0x6a Total Reset Value: 0x0001

Bits	Access	Name	Description	Reset

Bits	Access	Name	Description	Reset
15:14	R	DRV_SEL_VAL	Driver Strength Select Value These bits indicate Driver strength value supported for SDR50 bus speed mode. These bits are meaningless for 3.3V signaling. Values: ■ 0x0 (TYPEB): Driver Type B is selected ■ 0x1 (TYPEA): Driver Type A is selected ■ 0x2 (TYPEC): Driver Type C is selected ■ 0x3 (TYPED): Driver Type D is selected	0x0
13:11	R	RSVD_13_11	These bits of the PRESET_SDR50_R register are reserved. They always return 0.	0x0
10	R	CLK_GEN_SEL_VAL	Clock Generator Select Value This bit is effective when Host Controller supports programmable clock generator. Values: ■ 0x0 (FALSE): Host Controller Ver2.0 Compatible Clock Generator ■ 0x1 (PROG): Programmable Clock Generator	0x0
9:0	R	FREQ_SEL_VAL	SDCLK/RCLK Frequency Select Value 10-bit preset value to be set in SDCLK/RCLK Frequency Select field of the Clock Control register described by a Host System.	0x1

PRESET_SDR104_R

PRESET_SDR104_R is a Preset Value for SDR104 Offset Address : 0x6c Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
15:14	R	DRV_SEL_VAL	Driver Strength Select Value These bits indicate Driver strength value supported for SDR104 bus speed mode. These bits are meaningless for 3.3V signaling. Values: ■ 0x0 (TYPEB): Driver Type B is selected ■ 0x1 (TYPEA): Driver Type A is selected ■ 0x2 (TYPEC): Driver Type C is selected ■ 0x3 (TYPED): Driver Type D is selected	0x0
13:11	R	RSVD_13_11	These bits of the PRESET_SDR104_R register are reserved. They always return 0.	0x0
10	R	CLK_GEN_SEL_VAL	Clock Generator Select Value This bit is effective when Host Controller supports programmable clock generator. Values: ■ 0x0 (FALSE): Host Controller Ver2.0 Compatible Clock Generator ■ 0x1 (PROG): Programmable Clock Generator	0x0

Bits	Access	Name	Description	Reset
9:0	R	FREQ_SEL_VAL	SDCLK/RCLK Frequency Select Value These bits specify a 10-bit preset value that must be set in the SDCLK/RCLK Frequency Select field of the Clock Control register described by a Host System.	0x0

PRESET_DDR50_R

PRESET_DDR50_R is a Preset Value for DDR50 Offset Address : 0x6e Total Reset Value: 0x0002

Bits	Access	Name	Description	Reset
15:14	R	DRV_SEL_VAL	Driver Strength Select Value These bits indicate Driver strength value supported for DDR50 bus speed mode. These bits are meaningless for 3.3V signaling. Values: ■ 0x0 (TYPEB): Driver Type B is selected ■ 0x1 (TYPEA): Driver Type A is selected ■ 0x2 (TYPEC): Driver Type C is selected ■ 0x3 (TYPED): Driver Type D is selected	0x0
13:11	R	RSVD_13_11	These bits of the PRESET_DDR50_R register are reserved. They always return 0.	0x0
10	R	CLK_GEN_SEL_VAL	Clock Generator Select Value This bit is effective when Host Controller supports programmable clock generator. Values: ■ 0x0 (FALSE): Host Controller Ver2.0 Compatible Clock Generator ■ 0x1 (PROG): Programmable Clock Generator	0x0
9:0	R	FREQ_SEL_VAL	SDCLK/RCLK Frequency Select Value These bits specify a 10-bit preset value that must be set in the SDCLK/RCLK Frequency Select field of the Clock Control register, as described by a Host System.	0x2

PRESET_UHS2_R

PRESET_UHS2_R is a Preset Value for UHS-II Offset Address : 0x74 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset

Bits	Access	Name	Description	Reset
15:14	R	DRV_SEL_VAL	Driver Strength Select Value These bits indicate the Driver strength value supported by 1.8V signaling bus speed modes in the SD mode. This field is meaningless for UHS-II mode. In eMMC mode, these bits can be used for selecting the Drive strength value for HS400 mode. Values: ■ 0x0 (TYPEB): Driver Type B is selected ■ 0x1 (TYPEA): Driver Type A is selected ■ 0x2 (TYPEC): Driver Type C is selected ■ 0x3 (TYPED): Driver Type D is selected	0x0
13:11	R	RSVD_13_11	These bits of UHS-II Preset register are reserved. They always return 0.	0x0
10	R	CLK_GEN_SEL_VAL	Clock Generator Select Value This bit is effective when the Host Controller supports a programmable clock generator. Values: ■ 0x0 (FALSE): Host Controller Ver2.0 Compatible Clock Generator ■ 0x1 (PROG): Programmable Clock Generator	0x0
9:0	R	FREQ_SEL_VAL	SDCLK/RCLK Frequency Select Value These bits specify the 10-bit preset value that must be set in the SDCLK/RCLK Frequency Select field of the Clock Control register, as described by a Host System.	0x0

ADMA_ID_LOW_R

ADMA_ID_LOW_R is an ADMA3 Integrated Descriptor Address Register - Low Offset Address : 0x78 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
31:0	R/W	ADMA_ID_LOW	ADMA Integrated Descriptor Address These bits indicate the lower 32-bit of the ADMA Integrated Descriptor address. The start address of Integrated Descriptor is set to these register bits. The ADMA3 fetches one Descriptor Address and increments these bits to indicate the next Descriptor address.	0x0

ADMA_ID_HIGH_R

ADMA_ID_HIGH_R is an ADMA3 Integrated Descriptor Address Register - High Offset Address : 0x7c Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
31:0	R/W	ADMA_ID_HIGH	ADMA Integrated Descriptor Address These bits indicate the higher 32 bit of the ADMA Integrated Descriptor address.	0x0

UHS_II_BLOCK_SIZE_R

UHS_II_BLOCK_SIZE_R is an UHS-II Block Size Register Offset Address : 0x80 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
15	R	RSVD_15	This bit of the UHS_II_BLOCK_SIZE_R register is reserved. It always returns 0.	0x0
14:12	R/W	UHS_II_SDMA_BOUND	UHS-II SDMA Buffer Boundary (SDMA only) These bits set the page size of the system memory management. When system memory is managed by paging, SDMA data transfer is performed in units of paging. The Host controller generates DMA interrupt at the page boundary and requests the Host driver to update the ADMA system address register. The SDMA waits until the ADMA system address register is written. This is active when DMA Enable in the UHS-II Transfer Mode register is set to 1. Values: ■ 0x0 (BYTES_4K): 4K bytes UHS-II SDMA Boundary ■ 0x1 (BYTES_8K): 8K bytes UHS-II SDMA Boundary ■ 0x2 (BYTES_16K): 16K bytes UHS-II SDMA Boundary ■ 0x3 (BYTES_32K): 32K bytes UHS-II SDMA Boundary ■ 0x4 (BYTES_64K): 64K bytes UHS-II SDMA Boundary ■ 0x5 (BYTES_128K): 128K bytes UHS-II SDMA Boundary ■ 0x6 (BYTES_256K): 256K bytes UHS-II SDMA Boundary ■ 0x7 (BYTES_512K): 512K bytes UHS-II SDMA Boundary	0x0
11:0	R/W	UHS_II_BLK_SIZE	UHS-II Block Size This register specifies the block size of the data packet. SD memory card uses fixed block size of 512 bytes. Variable block size can be used for SDIO. This register is effective when Data Present is set to 1 in the UHS-II Command register. ■ 0x0001 : 1 Byte ■ 0x0002 : 2 Bytes ■ 0x0003 : 3 Bytes ■ ■ ■ ■ 0x01FF : 511 Bytes ■ 0x0200 : 512 Bytes ■ ■ ■ ■ 0x0800 : 2048 Bytes Note: UHS-II Block size must be programmed with non-zero value for a data transfer.	0x0

UHS_II_BLOCK_COUNT_R

UHS_II_BLOCK_COUNT_R is an UHS-II Block Count Register Offset Address : 0x84 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset

Bits	Access	Name	Description	Reset
31:0	R/W	BLK_COUNT	UHS-II Blocks Count These bits specify the number of blocks to be transferred. This register is effective when Data Present is set to 1 in the UHS-II Command register and is enabled when Block Count Enable is set to 1 and Block/Byte Mode is set to 0 in the UHS-II Transfer Mode register. ■ 0x00000000 : Stop Count ■ 0x00000001 : 1 block ■ 0x00000002 : 2 blocks ■ ■ ■ ■ 0xFFFFFFFF : 4G-1 blocks Note: ■ When this register is read during a data transfer, it may return an invalid value. This register must be read only when there is no data transfer in progress or after a data transfer had completed. ■ UHS-II Block count must be programmed with a non-zero value for data transfer when Block Count Enable is set to 1 in the UHS-II Transfer Mode Register.	0x0

UHS_II_COMMAND_PKT_0_3_R

UHS_II_COMMAND_PKT_0_3_R is an UHS-II Command Packet Register (Byte 0, 1, 2 and 3) Offset Address : 0x88
Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
31:0	R/W	COMMAND_PKT_0_3	UHS-II command packet This register represents byte number 0 to byte number 3 of UHS-II command packet. These bytes must be programmed in Little-endian format. The UHS-II command packet can be a maximum of 20 bytes. The command length is specified by the UHS-II Command register. For more information on the UHS-II command packet, refer the UHS-II Addendum.	0x0

UHS_II_COMMAND_PKT_4_7_R

UHS_II_COMMAND_PKT_4_7_R is an UHS-II Command Packet Register (Byte 4, 5, 6 and 7) Offset Address : 0x8c
Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
31:0	R/W	COMMAND_PKT_4_7	UHS-II command packet This register represents byte number 4 to byte number 7 of UHS-II command packet. These bytes must be programmed in Little-endian format. The UHS-II command packet can be a maximum of 20 bytes. The command length is specified by the UHS-II Command register. For more information on the UHS-II command packet, refer the UHS-II Addendum.	0x0

UHS_II_COMMAND_PKT_8_11_R

UHS_II_COMMAND_PKT_8_11_R is an UHS-II Command Packet Register (Byte 8, 9, 10 and 11) Offset Address : 0x90 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
31:0	R/W	COMMAND_PKT_8_11	UHS-II command packet These bits represent byte number 8 to byte number 11 of UHS-II command packet. These bytes must be programmed in Little-endian format. The UHS-II command packet can be a maximum of 20 bytes. The command length is specified by the UHS-II Command register. For more information on the UHS-II command packet, refer the UHS-II Addendum.	0x0

[UHS_II_COMMAND_PKT_12_15_R](#)

UHS_II_COMMAND_PKT_12_15_R is an UHS-II Command Packet Register (Byte 12, 13, 14 and 15) Offset Address : 0x94 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
31:0	R/W	COMMAND_PKT_12_15	UHS-II command packet These bits represent byte number 12 to byte number 15 of UHS-II command packet. These bytes must be programmed in Little-endian format. The UHS-II command packet can be a maximum of 20 bytes. The command length is specified by the UHS-II Command register. For more information on the UHS-II command packet, refer the UHS-II Addendum.	0x0

[UHS_II_COMMAND_PKT_16_19_R](#)

UHS_II_COMMAND_PKT_16_19_R is an UHS-II Command Packet Register (Byte 16, 17, 18 and 19) Offset Address : 0x98 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
31:0	R/W	COMMAND_PKT_16_19	UHS-II command packet These bits represent byte number 16 to byte number 19 of the UHS-II command packet. These bytes must be programmed in Little-endian format. The UHS-II command packet can be a maximum of 20 bytes. The command length is specified by the UHS-II Command register. For more information on the UHS-II command packet, refer the UHS-II Addendum.	0x0

[UHS_II_XFER_MODE_R](#)

UHS_II_XFER_MODE_R is an UHS-II Transfer Mode Register Offset Address : 0x9c Total Reset Value: 0x0

Bits	Access	Name	Description	Reset

Bits	Access	Name	Description	Reset
15	R/W	HALF_FULL_SEL	Half/Full Select This bit is used to select the two-lane half-duplex mode. This bit must be set to 1 only when both the host and the device support two-lane half-duplex mode. The Host driver must read the UHS-II General Capabilities register to identify the capability of the Host to support two-lane half-duplex mode. Values: ■ 0x0 (FULL_DUPLEX): Full Duplex mode ■ 0x1 (HALF_DUPLEX): Half Duplex mode	0x0
14	R/W	EBSY_WAIT	EBSY Wait This bit is set while issuing a command that is accompanied by the EBSY packet to indicate end of command execution. If this bit is set to 1, the Host controller waits to receive EBSY packet and on receiving an EBSY packet, the Transfer Complete interrupt in the Normal Interrupt Status register is set to 1 to indicate end of busy. If error is indicated in the EBSY packet, the EBSY Error in the UHS-II Error Interrupt Status register is set to 1 and the Error Interrupt in the Normal Interrupt Status register is also set. Values: ■ 0x0 (CMD_WITHOUT_EBSY_WAIT): Issue a command without busy ■ 0x1 (CMD_WITH_EBSY_WAIT): Wait EBSY while issuing command with busy	0x0
13:9	R	RESERVED_13_9	These bits (RESERVED_13_9) of the UHS_II_XFER_MODE_R register are reserved. They always return 0.	0x0
8	R/W	RESP_INTR_DISABLE	Response Interrupt Disable This bit along with the Response Error Check Enable bit enable the Host controller to check the response error. The Host controller version 4.00 supports response error check function to avoid overhead of response error check by Host driver. The Command Complete interrupt is disabled when this bit is set to 1 regardless of Command Complete Signal Enable. Note that only response of type R1 and R5 can be checked by the Host controller. If this bit is set to 0, the Host driver must check the response register after generation of the Command Complete interrupt. Values: ■ 0x0 (ENABLE_RESP_INTR): Response Interrupt is enabled ■ 0x1 (DISABLE_RESP_INTR): Response Interrupt is disabled	0x0

Bits	Access	Name	Description	Reset
7	R/W	RESP_ERR_CHK_EN	Response Error Check Enable This bit along with the Response Interrupt Disable enables the Host controller to check the response error. If Host driver checks the response error, this bit is set to 0 and Response Interrupt Disable is set to 0. Host controller version 4.00 supports response error check function to avoid overhead of response error check by Host driver. If an error is detected, the RES Packet error interrupt is generated in the UHS-II Error Interrupt Status register. Values: ■ 0x0 (DISABLE_RESP_ERR_CHK): Response Error Check is disabled ■ 0x1 (ENABLE_RESP_ERR_CHK): Response Error Check is enabled	0x0
6	R/W	RESP_TYP	Response Type R1/R5 This bit selects a response type when the response error check is enabled. Note: Following errors in the response are checked by Host controller: Error statuses checked in R1 ■ Bit31 OUT_OF_RANGE ■ Bit30 ADDRESS_ERROR ■ Bit29 BLOCK_LEN_ERROR ■ Bit26 WP_VIOLATION ■ Bit25 CARD_IS_LOCKED ■ Bit23 COM_CRC_ERROR ■ Bit21 CARD_ECC_FAILED ■ Bit20 CC_ERROR ■ Bit19 ERROR Error statuses checked in R5 ■ Bit07 COM_CRC_ERROR ■ Bit03 ERROR ■ Bit01 FUNCTION_NUMBER ■ Bit00 OUT_OF_RANGE Values: ■ 0x0 (RESP_R1): R1 response (Memory) ■ 0x1 (RESP_R5): R5 response (SDIO)	0x0

Bits	Access	Name	Description	Reset
5	R/W	BLK_BYTE_MODE	Block/Byte Mode This bit specifies whether the data transfer is in block mode or byte mode when the Data Present bit in the UHS-II Command register is set to 1. Note: ■ Block mode: UHS-II Block size register represents the size of one DATA packet and UHS-II Block Count register represents the number of data blocks to be transmitted/received when block count is enabled. If block count is not enabled then it is treated as infinite transfer by the Host controller. An Abort command must be issued to end the infinite transfer. In the special case when ADMA2 is used and block count is disabled, transfer is ended after completing the last descriptor line. ■ Byte mode: Only one DATA packet is transmitted/received. Size of this DATA packet is dictated by the UHS-II Block Size register. Block count and block count enable is not considered for byte mode. Values: ■ 0x0 (BLOCK_MODE): Block mode ■ 0x1 (BYTE_MODE): Byte mode	0x0
4	R/W	D DATA_XFER_DIR	ata Transfer Direction This bit specifies direction of data transfer when Data Present in the UHS-II Command register is set to 1. Values: ■ 0x0 (READ): Read (Card to Host) ■ 0x1 (WRITE): Write (Host to Card)	0x0
3:2	R	RESERVED_3_2	These bits (RESERVED_3_2) of the UHS_II_XFER_MODE_R register are reserved. They always return 0.	0x0
1	R/W	BLK_CNT_EN	Block Count Enable This bit terminates data transfer by the block count that is specified using the UHS-II Block Count register. If this bit is set to 0, data transfer can either be terminated by end of descriptor line for ADMA2 operation, by CMD12, or by the TRANS_ABORT command. Values: ■ 0x0 (DISABLE_BLK_CNT): Block Count is disabled ■ 0x1 (ENABLE_BLK_CNT): Block Count is enabled	0x0
0	R/W	DMA_EN	DMA Enable This bit specifies whether DMA is used or not. DMA type is selected using DMA Select in Host Control1 register. Values: ■ 0x0 (DISABLE_DMA): DMA is disabled ■ 0x1 (ENABLE_DMA): DMA is enabled	0x0

UHS_II_CMD_R is an UHS-II Command Register Offset Address : 0x9e Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
15:13	R	RESERVED_15_13	These bits (RESERVED_15_13) of the UHS_II_CMD_R register are reserved. They always return 0.	0x0
12:8	R/W	UHS_II_CMD_PKT_LEN	UHS-II Command Packet Length These bits set the command packet length, which is set in the UHS-II Command Packet register. ■ 0x03-0x00 : 3-0 Bytes (Not used) ■ 0x04 : 4 Bytes ■ ■ 0x0C : 12 Bytes ■ ■ 0x14 : 20 Bytes ■ 0x1F-0x15 : Not used Note: The UHS-II protocol defines all command packet lengths in multiples of four. Setting these bits to a non multiple of four is not recommended.	0x0
7:6	R/W	UHS_II_CMD_TYP	UHS-II Command Type These bits are used to distinguish different command types. Note: While issuing UHS-II GO_DORMANT_STATE CCMD or FULL_RESET CCMD, UHS_II_CMD_TYPE shall be set to 0x3 For different command types, the response of the command (UHS-II RES packet) is stored in different register as shown below: ■ 0x0 : UHS-II Response register (0B3h-0A0h) ■ 0x1 : Response register (013h-010h) 0x2 : Response register (01Fh-018h) ■ 0x3 : UHS-II Response register (0B3h-0A0h) Values: ■ 0x0 (NORMAL_CCMD): Normal Command ■ 0x1 (TRANS_ABORT_CCMD): TRANS_ABORT command ■ 0x2 (CMD12_OR_SDIO_ABORT): CMD12 or SDIO Abort command ■ 0x3 (GO_DORMANT_CCMD): Go Dormant Command	0x0
5	R/W	DATA_PRESENT	Data Present This bit specifies whether the command is accompanied by data packet. Values: ■ 0x0 (NO_DATA_PRESENT): No data present ■ 0x1 (DATA_PRESENT): Data present	0x0
4:3	R	RESERVED_4_3	These bits (RESERVED_4_3) of the UHS_II_CMD_R register are reserved. They always return 0.	0x0

Bits	Access	Name	Description	Reset
2	R/W	SUB_CMD_FLAG	Sub Command Flag This bit is used to distinguish between the main command and the sub command. Setting of this bit is checked by the Sub Command Status in the Present state register. Note: The Host controller does not refer to this bit to issue a command. This bit is by the Host driver to manage the main command and the sub command. For more information, refer the Sub Command Status field in the Present State register. Values: ■ 0x0 (SUB_CMD): Sub Command ■ 0x1 (MAIN_CMD): Main Command	0x0
1:0	R	RESERVED_1_0	These bits (RESERVED_1_0) of the UHS_II_CMD_R register are reserved. They always return 0.	0x0

UHS_II_RESP_0_3_R

UHS_II_RESP_0_3_R is an UHS-II Response Register (Byte 0, 1, 2 and 3)

Offset Address : 0xa0 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
31:0	R	RESP_PKT_0_3	UHS-II response packet This register represents byte number 0 to byte number 3 of UHS-II response packet. These bytes are stored in Little endian format. The UHS-II response packet can be a maximum of 20 bytesFor more information on the format of a UHS-II response packet, refer the UHS-II Addendum.	0x0

UHS_II_RESP_4_7_R

UHS_II_RESP_4_7_R is an UHS-II Response Register (Byte 4, 5, 6 and 7) Offset Address : 0xa4 Total Reset Value:

0x0

Bits	Access	Name	Description	Reset
31:0	R	RESP_PKT_4_7	UHS-II response packet This register represents byte number 4 to byte number 7 of UHS-II response packet. These bytes are stored in Little-endian format. The UHS-II response packet can be a maximum of 20 bytesFor more information on the format of a UHS-II response packet, refer the UHS-II Addendum.	0x0

UHS_II_RESP_8_11_R

UHS_II_RESP_8_11_R is an UHS-II Response Register (Byte 8, 9, 10 and 11) Offset Address : 0xa8 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset

Bits	Access	Name	Description	Reset
31:0	R	RESP_PKT_8_11	UHS-II response packet These bits represent byte number 8 to byte number 11 of UHS-II response packet. These bytes are stored in Little-endian format. The UHS-II response packet can be a maximum of 20 bytesFor more information on the format of a UHS-II response packet, refer the UHS-II Addendum.	0x0

UHS_II_RESP_12_15_R

UHS_II_RESP_12_15_R is an UHS-II Response Register (Byte 12, 13, 14 and 15) Offset Address : 0xac Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
31:0	R	RESP_PKT_12_15	UHS-II response packet These bits represent byte number 12 to byte number 15 of the UHS-II response packet. These bytes are stored in Little-endian format. The UHS-II response packet can be a maximum of 20 bytes.For more information on the format of a UHS-II response packet, refer the UHS-II Addendum.	0x0

UHS_II_RESP_16_19_R

UHS_II_RESP_16_19_R is an UHS-II Response Register (Byte 16, 17, 18 and 19) Offset Address : 0xb0 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
31:0	R	RESP_PKT_16_19	UHS-II response packets This bit represents the byte number 16 to byte number 19 of the UHS-II response packet. These bytes are stored in Little-endian format. The UHS-II response packets can be a maximum of 20 bytes.For more information on the format of a UHS-II response packet, refer the UHS-II Addendum.	0x0

UHS_II_MSG_SEL_R

UHS_II_MSG_SEL_R is an UHS-II MSG Select Register Offset Address : 0xb4 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
7:2	R	RESERVED_7_2	These bits of the UHS_II_MSG_SEL_R register are reserved. They always return 0.	0x0

Bits	Access	Name	Description	Reset
1:0	R/W	UHS_II_MSG_SEL	UHS-II MSG Select These bits denote one of the four messages that can be read from the UHS-II MSG register. The host controller holds the last four message packets received from the UHS-II card. These bits select one of these four messages that can be read from the UHS-II MSG register. Values: ■ 0x0 (LATEST_MSG): The latest MSG ■ 0x1 (PREV_FIRST_MSG): One MSG before ■ 0x2 (PREV_SECOND_MSG): Two MSGs before ■ 0x3 (PREV_THIRD_MSG): Three MSGs before	0x0

UHS_II_MSG_R

UHS_II_MSG_R is an UHS-II MSG Register

Offset Address : 0xb8 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
31:0	R	UHS_II_MSG	UHS-II MSG These bits contain the MSG packet selected by the UHS-II MSG Packet Select register. The Host controller holds the last four MSG packets received from the UHS-II card. Usually two duplicate MSG packets are sent from the UHS-II card. One of these two MSG packets that are recognized as valid is stored in the MSG packet buffer.	0x0

UHS_II_DEV_INTR_STATUS_R

UHS_II_DEV_INTR_STATUS_R is an UHS-II Device Interrupt Status Register Offset Address : 0xbc Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
15	R/W1C	DEV_INTR_STATUS15	This bit indicates that the INT MSG is received from Device ID 15. Values: ■ 0x0 (NO_INT): No INT MSG received ■ 0x1 (INT): INT MSG received	0x0
14	R/W1C	DEV_INTR_STATUS14	This bit indicates that the INT MSG is received from Device ID 14. Values: ■ 0x0 (NO_INT): No INT MSG received ■ 0x1 (INT): INT MSG received	0x0
13	R/W1C	DEV_INTR_STATUS13	This bit indicates that the INT MSG is received from Device ID 13. Values: ■ 0x0 (NO_INT): No INT MSG received ■ 0x1 (INT): INT MSG received	0x0

Bits	Access	Name	Description	Reset
12	R/W1C	DEV_INTR_STATUS12	This bit indicates that the INT MSG is received from Device ID 12. Values: ■ 0x0 (NO_INT): No INT MSG received ■ 0x1 (INT): INT MSG received	0x0
11	R/W1C	DEV_INTR_STATUS11	This bit indicates that the INT MSG is received from Device ID 11. Values: ■ 0x0 (NO_INT): No INT MSG received ■ 0x1 (INT): INT MSG received	0x0
10	R/W1C	DEV_INTR_STATUS10	This bit indicates that the INT MSG is received from Device ID 10. Values: ■ 0x0 (NO_INT): No INT MSG received ■ 0x1 (INT): INT MSG received	0x0
9	R/W1C	DEV_INTR_STATUS9	This bit indicates that the INT MSG is received from Device ID 9. Values: ■ 0x0 (NO_INT): No INT MSG received ■ 0x1 (INT): INT MSG received	0x0
8	R/W1C	DEV_INTR_STATUS8	This bit indicates that the INT MSG is received from Device ID 8. Values: ■ 0x0 (NO_INT): No INT MSG received ■ 0x1 (INT): INT MSG received	0x0
7	R/W1C	DEV_INTR_STATUS7	This bit indicates that the INT MSG is received from Device ID 7. Values: ■ 0x0 (NO_INT): No INT MSG received ■ 0x1 (INT): INT MSG received	0x0
6	R/W1C	DEV_INTR_STATUS6	This bit indicates that the INT MSG is received from Device ID 6. Values: ■ 0x0 (NO_INT): No INT MSG received ■ 0x1 (INT): INT MSG received	0x0
5	R/W1C	DEV_INTR_STATUS5	This bit indicates that the INT MSG is received from Device ID 5. Values: ■ 0x0 (NO_INT): No INT MSG received ■ 0x1 (INT): INT MSG received	0x0
4	R/W1C	DEV_INTR_STATUS4	This bit indicates that the INT MSG is received from Device ID 4. Values: ■ 0x0 (NO_INT): No INT MSG received ■ 0x1 (INT): INT MSG received	0x0

Bits	Access	Name	Description	Reset
3	R/W1C	DEV_INTR_STATUS3	This bit indicates that the INT MSG is received from Device ID 3. Values: ■ 0x0 (NO_INT): No INT MSG received ■ 0x1 (INT): INT MSG received	0x0
2	R/W1C	DEV_INTR_STATUS2	This bit indicates that the INT MSG is received from Device ID 2. Values: ■ 0x0 (NO_INT): No INT MSG received ■ 0x1 (INT): INT MSG received	0x0
1	R/W1C	DEV_INTR_STATUS1	This bit indicates that the INT MSG is received from Device ID 1. Values: ■ 0x0 (NO_INT): No INT MSG received ■ 0x1 (INT): INT MSG received	0x0
0	R/W1C	DEV_INTR_STATUS0	This bit of the UHS_II_DEV_INTR_STATUS_R register is reserved. It always returns 0.	0x0

UHS_II_DEV_SEL_R

UHS_II_DEV_SEL_R is an UHS-II Device Select Register This register is applicable for UHS-II mode only. Offset Address : 0xbe Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
7	R/W	INT_MSG_EN	INT MSG Enable This bit enables receipt of the INT MSG. If this bit is set to 1, receipt of INT MSG is informed by the Card Interrupt in the Normal Interrupt Status register. If this bit is set to 0, the Host Controller ignores receipt of INT MSG and may not set the UHS-II Device Interrupt Code register. Support of INT MSG Interrupt is optional. If an attempt is made to set this bit to 1 while this bit is read 0, the INT MSG Interrupt is not supported by the Host Controller. In this case, the UHS-II Device Interrupt Status register always reads 0 and the UHS-II Device Interrupt Code register may not be implemented. Values: ■ 0x0 (DISABLE_INT_MSG): Reception of INT MSG is disabled ■ 0x1 (ENABLE_INT_MSG): Reception of INT MSG is enabled	0x0
6:4	R	RESERVED_6_4	These bits of the UHS_II_DEV_SEL_R register are reserved. They always return 0.	0x0

Bits	Access	Name	Description	Reset
3:0	R/W	DEV_SEL	UHS-II Device Select Host controller holds an INT MSG packet per device. One of the INT MSGs (up to 15) can be selected by this field and read from UHS-II Device Interrupt Code register(0BFh). This field is effective when INT MSG Enable is set to 1. The number of devices implemented in the Host Controller is indicated by the Number of Devices Supported in the UHS-II General Capabilities register. ■ 0x0 : Unselected (Default) ■ 0x1 : INT MSG of Device ID 1 is selected ■ 0x2 : INT MSG of Device ID 2 is selected ■ ■ ■ ■ 0xF : INT MSG of Device ID 15 is selected	0x0

UHS_II_DEV_INTR_CODE_R

UHS_II_DEV_INTR_CODE_R is an UHS-II Device Interrupt Code Register Offset Address : 0xbf Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
7:0	R	DEV_INTR_CODE	Device Interrupt Code These bits specify the INT MSG Packet of the selected device.	0x0

UHS_II_SOFT_RESET_R

UHS_II_SOFT_RESET_R is an UHS-II Software Reset Register

Offset Address : 0xc0 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
15:2	R	RESERVED_15_2	These bits of UHS_II_SOFT_RESET_R register are reserved. They always return 0.	0x0

Bits	Access	Name	Description	Reset
1	R/W	HOST_SD_TRAN_RST	Host SD-TRAN Reset This bit is set to 1 by the Host Driver to reset the SD-TRAN layer when an Abort command or CMD0 is issued to the device or when a data transfer error occurs. This bit is cleared automatically at completion. If CMD0 is issued, SD-TRAN Initialization sequence from CMD8 is required to use UHS-II mode assuming that the bus power is maintained. Note that CM-TRAN Initialization is not required. The Host Controller does the following: ■ SD Clock Enable is maintained (Continue to provide RCLK). ■ All Interrupt Status, Status Enable and Signal Enable are cleared. ■ Data transfer is terminated and data in buffer is discarded. Values: ■ 0x0 (FALSE): Continue normal operation ■ 0x1 (TRUE): Reset SD-TRAN	0x0
0	R/W	HOST_FULL_RST	Host Full Reset This bit is set by the Host Driver to reset Host controller, on issuing the FULL_RESET CCMD (to reset the device). This bit is automatically cleared on completion of the Host controller reset. The initialization sequence from PHY initialization is required to use the UHS-II mode considering that the bus power is maintained. The Host Controller does the following: ■ SD Clock Enable is cleared (Internal Clock is still synchronized) ■ All setting registers are cleared ■ Internal sequencers are reset to same the values as those just after power on ■ Interrupt Status, Status Enable, and Signal Enable are cleared Values: ■ 0x0 (FALSE): Continue normal operation ■ 0x1 (TRUE): Reset Host controller	0x0

UHS_II_TIMER_CNTRL_R

UHS_II_TIMER_CNTRL_R is an UHS-II Timer Control Register Offset Address : 0xc2 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
15:8	R	RESERVED_15_8	These bits of the UHS_II_TIMER_CNTRL_R register are reserved. They always return 0.	0x0

Bits	Access	Name	Description	Reset
7:4	R/W	TIMER_CNT_DEADLOCK	Timeout Counter Value for Deadlock These bits determine the deadlock period while the host is expecting to receive any packet other than the response packet (1 second). While setting these bits, prevent inadvertent timeout events by clearing the Timeout for Deadlock (in the UHS-II Error Interrupt Status Enable register). TMCLK represents the timer clock. DWC_mshc has an extra clock input specifically for the timer. The timeout counter value must be programmed considering the timer clock frequency. ■ 0xF : Reserved ■ 0xE : TMCLK x 2²⁷ ■ ■ ■ ■ 0x1 : TMCLK x 2¹⁴ ■ 0x0 : TMCLK x 2¹³ Note: The Host driver can get information about the timer clock frequency from the Capabilities register (Capabilities_1_R). The Timer clock frequency for DWC_mshc must be configured using the DWC_MSHC_TIMER_CLK_FREQ_UNIT and the DWC_MSHC_TIMER_CLK_FREQ parameters.	0x0
3:0	R/W	TIMEOUT_CNT_CMD_RES	Timeout Counter Value for CMD_RES These bits determine the interval between the command packet and the response packet(5ms). While setting this register, prevent inadvertent timeout events by clearing the Time out for CMD_RES (in the UHS-II Error Interrupt Status Enable register). TMCLK represents the timer clock. DWC_mshc has an extra clock input specifically for the timer (tmclk). The timeout counter value must be programmed considering timer clock frequency. ■ 0xF : Reserved ■ 0xE : TMCLK x 2²⁷ ■ ■ ■ ■ 0x1 : TMCLK x 2¹⁴ ■ 0x0 : TMCLK x 2¹³ Note: Host driver can get information about timer clock frequency from Capabilities register (Capabilities_1_R). Timer clock frequency for DWC_mshc must be configured using parameters DWC_MSHC_TIMER_CLK_FREQ_UNIT and DWC_MSHC_TIMER_CLK_FREQ.	0x0

UHS_II_ERR_INTR_STATUS_R is an UHS-II Error Interrupt Status Register Offset Address : 0xc4 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
31	R/W1C	VENDOR_SPECIFIC_ERR4	This bit (VENDOR_SPECIFIC_ERR4) of the UHS_II_ERR_INTR_STATUS_R register is reserved. It always returns 0. Values: ■ 0x0 (FALSE): Interrupt is not generated ■ 0x1 (TRUE): Vendor Specific Error	0x0
30	R/W1C	VENDOR_SPECIFIC_ERR3	This bit (VENDOR_SPECIFIC_ERR3) of the UHS_II_ERR_INTR_STATUS_R register is reserved. It always returns 0. Values: ■ 0x0 (FALSE): Interrupt is not generated ■ 0x1 (TRUE): Vendor Specific Error	0x0
29	R/W1C	VENDOR_SPECIFIC_ERR2	This bit (VENDOR_SPECIFIC_ERR2) of the UHS_II_ERR_INTR_STATUS_R register is reserved. It always returns 0. Values: ■ 0x0 (FALSE): Interrupt is not generated ■ 0x1 (TRUE): Vendor Specific Error	0x0
28	R/W1C	VENDOR_SPECIFIC_ERR1	This bit (VENDOR_SPECIFIC_ERR1) of the UHS_II_ERR_INTR_STATUS_R register is reserved. It always returns 0. Values: ■ 0x0 (FALSE): Interrupt is not generated ■ 0x1 (TRUE): Vendor Specific Error	0x0
27	R/W1C	VENDOR_SPECIFIC_ERR0	This bit (VENDOR_SPECIFIC_ERR0) of the UHS_II_ERR_INTR_STATUS_R register is reserved. It always returns 0. Values: ■ 0x0 (FALSE): Interrupt is not generated ■ 0x1 (TRUE): Vendor Specific Error	0x0
26:18	R	RESERVED_26_18	These bits (RESERVED_26_18) of the UHS_II_ERR_INTR_STATUS_R register are reserved bits. They always return 0.	0x0
17	R/W1C	TIMEOUT_DEADLOCK	Timeout for Deadlock Setting this bit indicates that a deadlock timeout has occurred. When the Host expects to receive a packet and if does not receive it within a specified time (1 second), then this bit is set to 1. The Timeout value is determined by the setting of Timeout Counter Value for Deadlock in UHS-II Timer Control register. Values: ■ 0x0 (FALSE): Interrupt is not generated ■ 0x1 (TRUE): Timeout for Deadlock Error	0x0

Bits	Access	Name	Description	Reset
16	R/W1C	TIMEOUT_CMD_RES	Timeout for CMD_RES Setting of this bit indicates that an RES packet timeout has occurred. When the Host expects to receive a RES packet and if does not receive it within a specified time (5ms), then this bit is set to 1. The Timeout value is determined by the setting of Timeout Counter Value for CMD_RES in UHS-II Timer Control register. Values: ■ 0x0 (FALSE): Interrupt is not generated ■ 0x1 (TRUE): Timeout for CMD_RES Error	0x0
15	R/W1C	ADMA_ERR	ADMA Error This bit denotes that an ADMA error has occurred in the UHS-II mode. The ADMA error status is indicated in the ADMA Error Status register. Note: There are two ADMA Error bits. One is in Error Interrupt Status register and other one is in the UHS-II Error Interrupt Status register. The former must be enabled and used for SD mode, and latter must be enabled and used for UHS-II mode. Values: ■ 0x0 (FALSE): Interrupt is not generated ■ 0x1 (TRUE): ADMA Error	0x0
14:9	R	RESERVED_14_9	These bits (RESERVED_14_9) of the UHS_II_ERR_INTR_STATUS_R register are reserved. They always return 0.	0x0
8	R/W1C	EBSY_ERR	EBSY Error This bit is set if a packet indicates an error after receiving an EBSY packet. If the EBSY wait bit in UHS-II Transfer mode register is set, the Transfer complete interrupt in the Normal Interrupt Status register is set along with the Error Interrupt if EBSY packet indicated an error. Values: ■ 0x0 (FALSE): Interrupt is not generated ■ 0x1 (TRUE): EBSY Error	0x0
7	R/W1C	UNRECOVERABLE_ERR	Unrecoverable Error Setting of this bit means that Unrecoverable error is set in a packet from a device. Values: ■ 0x0 (FALSE): Interrupt is not generated ■ 0x1 (TRUE): Unrecoverable Error	0x0
6	R	RESERVED_6	This bit (RESERVED_6) of the UHS_II_ERR_INTR_STATUS_R register is reserved. It always returns 0.	0x0

Bits	Access	Name	Description	Reset
5	R/W1C	TID_ERR	TID Error This bit indicates that a TID error has occurred. The Host controller checks the Transaction ID (TID) of the command with that of received packets in a transfer. If there is any mismatch, a TID error is generated. Values: ■ 0x0 (FALSE): Interrupt is not generated ■ 0x1 (TRUE): TID Error	0x0
4	R/W1C	FRAMING_ERR	Framing Error This bit indicates that a Framing error has occurred while receiving a packet. Note: This bit is not set in following scenarios even if Framing error occurs while receiving a packet: ■ Framing error occurs in receiving one of the two duplicate MSG packets and the other is received correctly. ■ Framing error occurs in receiving data packets and a retry counter has not yet expired. Values: ■ 0x0 (FALSE): Interrupt is not generated ■ 0x1 (TRUE): Framing Error	0x0
3	R/W1C	CRC_ERR	CRC Error This bit indicates that a CRC error has occurred while receiving a packet. Note: This bit is not set in the following two scenarios even if CRC error occurs while receiving a packet. ■ CRC error occurs in receiving one of two duplicate MSG packets and the other is received correctly. ■ CRC error occurs in receiving data packets and the retry counter has not yet expired. Values: ■ 0x0 (FALSE): Interrupt is not generated ■ 0x1 (TRUE): CRC Error	0x0
2	R/W1C	RETRY_EXPIRED	Retry Expired Error This bit indicates that the Retry Counter Expired Error has occurred during data transfer. If this bit is set, either a Framing Error or a CRC Error in this register is set. Retry count must be programmed in the UHS-II LINK/TRAN Settings register. Values: ■ 0x0 (FALSE): Interrupt is not generated ■ 0x1 (TRUE): Retry expired Error	0x0

Bits	Access	Name	Description	Reset
1	R/W1C	RES_PKT_ERR	RES Packet Error Host controller version 4.00 supports response error check function to avoid overhead of response error check by Host Driver during DMA execution. If Response Error Check Enable is set to 1 in the UHS-II Transfer mode register, Host Controller Checks R1 or R5 response. If an error is detected in a response, this bit is set to 1. Note: Refer the Response Type field of the UHS-II Transfer Mode register to know the errors that are checked by the Host controller. Values: ■ 0x0 (FALSE): Interrupt is not generated ■ 0x1 (TRUE): RES Packet Error	0x0
0	R/W1C	HEADER_ERR	Header Error This bit indicates that a Header error has occurred in a received packet. The Host controller checks the Destination ID and the Transaction ID of the command with the received packets (RES, MSG and DATA) in a transfer. If there is any mismatch, this error is generated. Values: ■ 0x0 (FALSE): Interrupt is not generated ■ 0x1 (TRUE): Header Error	0x0

UHS_II_ERR_INTR_STATUS_EN_R

UHS_II_ERR_INTR_STATUS_EN_R is an UHS-II Error Interrupt Status Enable Register

Offset Address : 0xc8 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
31	R/W	VENDOR_SPECIFIC_ERR4	This bit (VENDOR_SPECIFIC_ERR4) of the UHS_II_ERR_INTR_STATUS_EN_R register is reserved. It always returns 0. Values: ■ 0x0 (DISABLE): Status is disabled ■ 0x1 (ENABLE): Status is enabled	0x0
30	R/W	VENDOR_SPECIFIC_ERR3	This bit (VENDOR_SPECIFIC_ERR3) of the UHS_II_ERR_INTR_STATUS_EN_R register is reserved. It always returns 0. Values: ■ 0x0 (DISABLE): Status is disabled ■ 0x1 (ENABLE): Status is enabled	0x0

Bits	Access	Name	Description	Reset
29	R/W	VENDOR_SPECIFIC_ERR2	This bit (VENDOR_SPECIFIC_ERR2) of the UHS_II_ERR_INTR_STATUS_EN_R register is reserved. It always returns 0. Values: ■ 0x0 (DISABLE): Status is disabled ■ 0x1 (ENABLE): Status is enabled	0x0
28	R/W	VENDOR_SPECIFIC_ERR1	This bit (VENDOR_SPECIFIC_ERR1) of the UHS_II_ERR_INTR_STATUS_EN_R register is reserved. It always returns 0. Values: ■ 0x0 (DISABLE): Status is disabled ■ 0x1 (ENABLE): Status is enabled	0x0
27	R/W	VENDOR_SPECIFIC_ERR0	This bit (VENDOR_SPECIFIC_ERR0) of the UHS_II_ERR_INTR_STATUS_EN_R register is reserved. It always returns 0. Values: ■ 0x0 (DISABLE): Status is disabled ■ 0x1 (ENABLE): Status is enabled	0x0
26:18	R	RESERVED_26_18	These bits (RESERVED_26_18) of the UHS_II_ERR_INTR_STATUS_EN_R register are reserved. They always return 0.	0x0
17	R/W	TIMEOUT_DEADLOCK	Timeout for Deadlock This bit sets the Timeout for Deadlock bit in the UHS-II Error Interrupt Status register. Values: ■ 0x0 (DISABLE): Status is disabled ■ 0x1 (ENABLE): Status is enabled	0x0
16	R/W	TIMEOUT_CMD_RES	Timeout for CMD_RES This bit sets the Timeout for CMD_RES bit in the UHS-II Error Interrupt Status register. Values: ■ 0x0 (DISABLE): Status is disabled ■ 0x1 (ENABLE): Status is enabled	0x0
15	R/W	ADMA_ERR	ADMA Error This bit sets the ADMA Error bit in the UHS-II Error Interrupt Status register. Values: ■ 0x0 (DISABLE): Status is disabled ■ 0x1 (ENABLE): Status is enabled	0x0
14:9	R	RESERVED_14_9	These bits (RESERVED_14_9) of UHS_II_ERR_INTR_STATUS_EN_R register are reserved. They always return 0.	0x0

Bits	Access	Name	Description	Reset
8	R/W	EBSY_ERR	EBSY Error This bit sets the EBSY Error bit in the UHS-II Error Interrupt Status register. Values: ■ 0x0 (DISABLE): Status is disabled ■ 0x1 (ENABLE): Status is enabled	0x0
7	R/W	UNRECOVERABLE_ERR	Unrecoverable Error This bit sets the Unrecoverable Error bit in the UHS-II Error Interrupt Status register. Values: ■ 0x0 (DISABLE): Status is disabled ■ 0x1 (ENABLE): Status is enabled	0x0
6	R	RESERVED_6	This bit (RESERVED_6) of the UHS_II_ERR_INTR_STATUS_EN_R register is reserved. It always returns 0.	0x0
5	R/W	TID_ERR	TID Error This bit sets the TID Error bit in the UHS-II Error Interrupt Status register. Values: ■ 0x0 (DISABLE): Status is disabled ■ 0x1 (ENABLE): Status is enabled	0x0
4	R/W	FRAMING_ERR	Framing Error This bit sets the Framing Error bit in the UHS-II Error Interrupt Status register. Values: ■ 0x0 (DISABLE): Status is disabled ■ 0x1 (ENABLE): Status is enabled	0x0
3	R/W	CRC_ERR	CRC Error This bit sets the CRC Error bit in the UHS-II Error Interrupt Status register. Values: ■ 0x0 (DISABLE): Status is disabled ■ 0x1 (ENABLE): Status is enabled	0x0
2	R/W	RETRY_EXPIRED	Retry Expired Error This bit sets the Retry Expired Error bit in the UHS-II Error Interrupt Status register. Values: ■ 0x0 (DISABLE): Status is disabled ■ 0x1 (ENABLE): Status is enabled	0x0
1	R/W	RES_PKT_ERR	RES Packet Error This bit sets the RES Packet Error bit in the UHS-II Error Interrupt Status register. Values: ■ 0x0 (DISABLE): Status is disabled ■ 0x1 (ENABLE): Status is enabled	0x0

Bits	Access	Name	Description	Reset
0	R/W	HEADER_ERR	Header Error This bit sets the Header Error bit in the UHS-II Error Interrupt Status register. Values: ■ 0x0 (DISABLE): Status is disabled ■ 0x1 (ENABLE): Status is enabled	0x0

UHS_II_ERR_INTR_SIGNAL_EN_R

UHS_II_ERR_INTR_SIGNAL_EN_R is an UHS-II Error Interrupt Signal Enable Register Offset Address : 0xcc Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
31	R/W	VENDOR_SPECIFIC_ERR4	This bit (VENDOR_SPECIFIC_ERR4) of the UHS_II_ERR_INTR_SIGNAL_EN_R register is reserved. It always returns 0. Values: ■ 0x0 (DISABLE): Status is disabled ■ 0x1 (ENABLE): Status is enabled	0x0
30	R/W	VENDOR_SPECIFIC_ERR3	This bit (VENDOR_SPECIFIC_ERR3) of the UHS_II_ERR_INTR_SIGNAL_EN_R register is reserved. It always returns 0. Values: ■ 0x0 (DISABLE): Status is disabled ■ 0x1 (ENABLE): Status is enabled	0x0
29	R/W	VENDOR_SPECIFIC_ERR2	This bit (VENDOR_SPECIFIC_ERR2) of the UHS_II_ERR_INTR_SIGNAL_EN_R register is reserved. It always returns 0. Values: ■ 0x0 (DISABLE): Status is disabled ■ 0x1 (ENABLE): Status is enabled	0x0
28	R/W	VENDOR_SPECIFIC_ERR1	This bit (VENDOR_SPECIFIC_ERR1) of the UHS_II_ERR_INTR_SIGNAL_EN_R register is reserved. It always returns 0. Values: ■ 0x0 (DISABLE): Status is disabled ■ 0x1 (ENABLE): Status is enabled	0x0
27	R/W	VENDOR_SPECIFIC_ERR0	This bit (VENDOR_SPECIFIC_ERR0) of the UHS_II_ERR_INTR_SIGNAL_EN_R register is reserved. It always returns 0. Values: ■ 0x0 (DISABLE): Status is disabled ■ 0x1 (ENABLE): Status is enabled	0x0
26:18	R	RESERVED_26_18	These bits (RESERVED_26_18) of the UHS_II_ERR_INTR_SIGNAL_EN_R register are reserved bits. They always return 0.	0x0

Bits	Access	Name	Description	Reset
17	R/W	TIMEOUT_DEADLOCK	Timeout for Deadlock This bit generates the interrupt signal when Timeout for Deadlock bit is set in the UHS-II Error Interrupt Status register. Values: ■ 0x0 (DISABLE): Interrupt signal is disabled ■ 0x1 (ENABLE): Interrupt signal is enabled	0x0
16	R/W	TIMEOUT_CMD_RES	Timeout for CMD_RES This bit generates the interrupt signal when Timeout for CMD_RES bit is set in the UHS-II Error Interrupt Status register. Values: ■ 0x0 (DISABLE): Interrupt signal is disabled ■ 0x1 (ENABLE): Interrupt signal is enabled	0x0
15	R/W	ADMA_ERR	ADMA Error This bit generates the interrupt signal when ADMA Error bit is set in the UHS-II Error Interrupt Status register. Values: ■ 0x0 (DISABLE): Interrupt signal is disabled ■ 0x1 (ENABLE): Interrupt signal is enabled	0x0
14:9	R	RESERVED_14_9	These bits (RESERVED_14_9) of the UHS_II_ERR_INTR_SIGNAL_EN_R register are reserved. They always return 0.	0x0
8	R/W	EBSY_ERR	EBSY Error This bit generates the interrupt signal when EBSY Error bit is set in the UHS-II Error Interrupt Status register. Values: ■ 0x0 (DISABLE): Interrupt signal is disabled ■ 0x1 (ENABLE): Interrupt signal is enabled	0x0
7	R/W	UNRECOVERABLE_ERR	Unrecoverable Error This bit generates the interrupt signal when the Unrecoverable Error bit is set in the UHS-II Error Interrupt Status register. Values: ■ 0x0 (DISABLE): Interrupt signal is disabled ■ 0x1 (ENABLE): Interrupt signal is enabled	0x0
6	R	RESERVED_6	This bit (RESERVED_6) of the UHS_II_ERR_INTR_SIGNAL_EN_R register is reserved. It always returns 0.	0x0

Bits	Access	Name	Description	Reset
5	R/W	TID_ERR	TID Error This bit generates the interrupt signal when TID Error bit is set in the UHS-II Error Interrupt Status register. Values: ■ 0x0 (DISABLE): Interrupt signal is disabled ■ 0x1 (ENABLE): Interrupt signal is enabled	0x0
4	R/W	FRAMING_ERR	Framing Error This bit generates the interrupt signal when Framing Error bit is set in the UHS-II Error Interrupt Status register. Values: ■ 0x0 (DISABLE): Interrupt signal is disabled ■ 0x1 (ENABLE): Interrupt signal is enabled	0x0
3	R/W	CRC_ERR	CRC Error This bit generates the interrupt signal when the CRC Error bit is set in the UHS-II Error Interrupt Status register. Values: ■ 0x0 (DISABLE): Interrupt signal is disabled ■ 0x1 (ENABLE): Interrupt signal is enabled	0x0
2	R/W	RETRY_EXPIRED	Retry Expired Error This bit generates the interrupt signal when the Retry Expired Error bit is set in the UHS-II Error Interrupt Status register. Values: ■ 0x0 (DISABLE): Interrupt signal is disabled ■ 0x1 (ENABLE): Interrupt signal is enabled	0x0
1	R/W	RES_PKT_ERR	RES Packet Error This bit generates the interrupt signal when the RES Packet Error bit is set in the UHS-II Error Interrupt Status register. Values: ■ 0x0 (DISABLE): Interrupt signal is disabled ■ 0x1 (ENABLE): Interrupt signal is enabled	0x0
0	R/W	HEADER_ERR	Header Error This bit generates the interrupt signal when the Header Error bit is set in the UHS-II Error Interrupt Status register. Values: ■ 0x0 (DISABLE): Interrupt signal is disabled ■ 0x1 (ENABLE): Interrupt signal is enabled	0x0

P_UHS_II_SETTINGS_R

P_UHS_II_SETTINGS_R is a Pointer for UHS-II Settings Offset Address : 0xe0 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset

Bits	Access	Name	Description	Reset
15:12	R	RESERVED_15_12	These bits of the P_UHS_II_SETTINGS_R register are reserved. They always return 0.	0x0
11:0	R	REG_OFFSET_ADDR	Offset Address of UHS-II Host Settings register.	0x0

[P_UHS_II_HOST_CAPAB](#)

P_UHS_II_HOST_CAPAB is a Pointer for UHS-II Host Capabilities Offset Address : 0xe2 Total Reset Value: 0x0f58

Bits	Access	Name	Description	Reset
15:12	R	RESERVED_15_12	These bits of the P_UHS_II_HOST_CAPAB register are reserved. They always return 0.	0x0
11:0	R	REG_OFFSET_ADDR	Offset Address for UHS-II Host Capabilities register.	0xf58

[P_UHS_II_TEST](#)

P_UHS_II_TEST is a Pointer for UHS-II Test Offset Address : 0xe4 Total Reset Value: 0x0f68

Bits	Access	Name	Description	Reset
15:12	R	RESERVED_15_12	These bits of the P_UHS_II_TEST register are reserved. They always return 0.	0x0
11:0	R	REG_OFFSET_ADDR	Offset Address of UHS-II Test register.	0xf68

[P_EMBEDDED_CNTRL](#)

P_EMBEDDED_CNTRL is a Pointer for Embedded Control Offset Address : 0xe6 Total Reset Value: 0x0f6c

Bits	Access	Name	Description	Reset
15:12	R	RESERVED_15_12	These bits of the P_EMBEDDED_CNTRL register are reserved. They always return 0.	0x0
11:0	R	REG_OFFSET_ADDR	Offset Address of Embedded Control register.	0xf6c

[P_VENDOR_SPECIFIC_AREA](#)

P_VENDOR_SPECIFIC_AREA is a Pointer for Vendor Specific Area 1 Offset Address : 0xe8 Total Reset Value: 0x0500

Bits	Access	Name	Description	Reset
15:12	R	RESERVED_15_12	These bits of the P_VENDOR_SPECIFIC_AREA register are reserved. They always return 0.	0x0
11:0	R	REG_OFFSET_ADDR	Base offset Address for Vendor-Specific registers.	0x500

[P_VENDOR2_SPECIFIC_AREA](#)

P_VENDOR2_SPECIFIC_AREA is a Pointer for Vendor Specific Area 2 Offset Address : 0xea Total Reset Value: 0x0384

Bits	Access	Name	Description	Reset

Bits	Access	Name	Description	Reset
15:0	R	REG_OFFSET_ADDR	Base offset Address for Command Queuing registers.	0x384

SLOT_INTR_STATUS_R

SLOT_INTR_STATUS_R is a Slot Interrupt Status Register Offset Address : 0xfc Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
15:8	R	RESERVED_15_8	These bits of the SLOT_INTR_STATUS_R register are reserved. They always return 0.	0x0
7:0	R	INTR_SLOT	Interrupt signal for each Slot These status bits indicate the logical OR of Interrupt signal and Wakeup signal for each slot. A maximum of 8 slots can be defined. If one interrupt signal is associated with multiple slots, the Host Driver can identify the interrupt that is generated by reading these bits. By a power on reset or by setting Software Reset For All bit, the interrupt signals are de-asserted and this status reads 00h. ■ Bit 00: Slot 1 ■ Bit 01: Slot 2 ■ Bit 02: Slot 3 ■ ■ ■ Bit 07: Slot 8 Note: MSHC Host Controller support single card slot. This register shall always return 0.	0x0

HOST_CNTRL_VERS_R

HOST_CNTRL_VERS_R is a Host Controller Version Offset Address : 0xfe Total Reset Value: 0x0005

Bits	Access	Name	Description	Reset
15:8	R	VENDOR_VERSION_NUM	Vendor Version Number This field is reserved for the vendor version number. Host Driver must not use this status.	0x0

Bits	Access	Name	Description	Reset
7:0	R	SPEC_VERSION_NUM	Specification Version Number These bits indicate the Host controller specification version. The upper and lower 4-bits indicate the version. Values 0x06-0xFF are reserved. Values: ■ 0x0 (VER_1_00): SD Host Controller Specification Version 1.00 ■ 0x1 (VER_2_00): SD Host Controller Specification Version 2.00 ■ 0x2 (VER_3_00): SD Host Controller Specification Version 3.00 ■ 0x3 (VER_4_00): SD Host Controller Specification Version 4.00 ■ 0x4 (VER_4_10): SD Host Controller Specification Version 4.10 ■ 0x5 (VER_4_20): SD Host Controller Specification Version 4.20	0x5

UHS2_GEN_SET_R

UHS2_GEN_SET_R is an UHS-II General Setting register Offset Address : P_UHS_II_SETTINGS_R[11:0] Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
31:12	R	RSVD_31_12	These bits of the UHS2_GEN_SET_R register are reserved. They always return 0.	0x0
11:8	R/W	NUM_LANE	Number of Lanes and Functionalities These bits set the lane configuration of a Host System and the value depends on the capability between the Host Controller and the connected devices. A 2-lane FD mode is mandatory and other modes are optional. Values 0x5-0xF are reserved. Values: ■ 0x0 (II_LANES): 2 Lanes FD or 2L-HD ■ 0x1 (NOT_USED): Not used ■ 0x2 (LANES_2D1U): 3 Lanes 2D1U-FD (Embedded) ■ 0x3 (LANES_1D2U): 3 Lanes 1D2U-FD (Embedded) ■ 0x4 (LANES_2D2U): 4 Lanes 2D2U-FD (Embedded)	0x0
7:1	R	RSVD_7_1	These bits of the UHS2_GEN_SET_R register are reserved bits. They always return 0.	0x0
0	R/W	PWR_MODE	Power mode This bit determines either Fast Mode or Low Power Mode. The Host and all the devices connected to the host are set to the same mode. Values: ■ 0x0 (FAST_MODE): Fast Mode ■ 0x1 (LOW_PWR_MODE): Low Power Mode	0x0

UHS2_PHY_SET_R

UHS2_PHY_SET_R is an UHS-II Phy Setting register Offset Address : P_UHS_II_SETTINGS_R[11:0] + 0x4 Total
Reset Value: 0x0

Bits	Access	Name	Description	Reset
31:24	R	RSVD_31_24	These bits (RSVD_31_24) of the UHS2_PHY_SET_R register are reserved. They always return 0.	0x0
23:20	R/W	N_LSS_DIR	Host N_LSS_DIR This bit is set to the largest value of N_LSS_DIR capabilities among the Host Controller and the connected devices. ■ 0x0 : 8x16 LSS ■ 0x1 : 8x1 LSS ■ 0x2 : 8x2 LSS ■ ■ ■ ■ 0xF : 8x15 LSS	0x0
19:16	R/W	N_LSS_SYN	Host N_LSS_SYN These bits are set to the largest value of N_LSS_SYN capabilities among the Host Controller and the connected devices. ■ 0x0 : 4x16 LSS ■ 0x1 : 4x1 LSS ■ 0x2 : 4x2 LSS ■ ■ ■ ■ 0xF : 4x15 LSS	0x0
15	R/W	HBNATE_EN	Hibernate Enable This bit is set if all devices support Hibernate mode after checking the card capabilities of the Hibernate mode. This bit determines whether Host remains in Dormant state or goes to Hibernate state. In Hibernate mode, VDD1 Power of the card can be cut off. Note: Hibernate functionality will be supported in a future release of DWC_mshc. Values: ■ 0x0 (DISABLED): Hibernate Disabled ■ 0x1 (ENABLED): Hibernate Enabled	0x0
14:8	R	RSVD_14_8	These bits (RSVD_14_8) of the UHS2_PHY_SET_R register are reserved bits. They always return 0.	0x0
7:6	R/W	SPD_RANGE	Speed Range These bits select the PLL multiplier. Change of PLL Multiplier is not effective immediately and is applied after exiting from Dormant State. Values: ■ 0x0 (RANGE_A): Range A (default) ■ 0x1 (RANGE_B): Range B ■ 0x2 (RSVD1): Reserved ■ 0x3 (RSVD2): Reserved	0x0

Bits	Access	Name	Description	Reset
5:0	R	RSVD_5_0	These bits (RSVD_5_0) of the UHS2_PHY_SET_R register are reserved. They always return 0.	0x0

UHS2_LNK_TRAN_SET_1_R

UHS2_LNK_TRAN_SET_1_R is an UHS-II LINK/TRAN Setting register Offset Address :

P_UHS_II_SETTINGS_R[11:0] + 0x80 Total Reset Value: 0x0

Bits	Name	Access	Description	Reset
31:18	R	RSVD_31_18	These bits (RSVD_31_18) of the UHS2_LNK_TRAN_SET_1_R register are reserved. They always return 0.	0x0
17:16	RTRY_CNT	R/W	Retry Count DATA Burst retry count is set to this field. Values: ■ 0x0 (DISABLED): Retry Disabled ■ 0x1 (ONE): 1 time ■ 0x2 (TWO): 2 times ■ 0x3 (THREE): 3 times	0x0
15:8	R/W	N_FCU	Host N_FCU The Host Driver sets the number of blocks in the Data Burst (Flow Control) to this field. The value is smaller than or equal to N_FCU capabilities among the Host Controller and connected devices. Setting 1 to 4 blocks is recommended, considering the buffer size. ■ 0x00: 256 Blocks ■ 0x01: 1 Block ■ 0x02: 2 Blocks ■ 0x03: 3 Blocks ■ ■ ■ ■ 0xFF: 255 Blocks Note: The N_FCU capability of the Host Controller is determined based on the packet buffer depth and maximum block size supported. If smaller block size is used, the Host driver can still program higher value of N_FCU setting (compared to N_FCU capability of host controller) as long as the device supports this higher N_FCU settings and following relation is satisfied in Host controller. $N\_FCU \leq \frac{DWC\_MSHC\_PKT\_BUFFER\_DEPTH}{2 * ((Desired\ block\ size) / (DWC\_MSHC\_MBIU\_DW/8))}$ where, DWC_MSHC_PKT_BUFFER_DEPTH - Packet buffer depth DWC_MSHC_MBIU_DW - AXI data width Desired block size - Size of data packet for data transfer	0x0
7:0	R	RSVD_7_0	These bits (RSVD_7_0) of the UHS2_LNK_TRAN_SET_1_R register are reserved. They always return 0.	0x0

UHS2_LNK_TRAN_SET_2_R

UHS2_LNK_TRAN_SET_2_R is an UHS-II LINK/TRAN Setting 2 register Offset Address :

P_UHS_II_SETTINGS_R[11:0] + 0xc Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
31:8	R	RSVD_31_8	These bits of the UHS2_LNK_TRAN_SET_2_R register are reserved. They always return 0.	0x0
7:0	R/W	N_DATA_GAP	Host N_DATA_GAP These bits set the largest value of N_DATA_GAP capabilities among the Host Controller and connected devices. ■ 0x00: No Gap ■ 0x01: 1 LSS ■ 0x02: 2 LSS ■ 0x03: 3 LSS ■ ■ ■ ■ 0xFF: 255 LSS	0x0

UHS2_GEN_CAP_R

UHS2_GEN_CAP_R is an UHS2 General Capabilities register

Offset Address : P_UHS_II_HOST_CAPAB[11:0] Total Reset Value: 0x00020000

Bits	Access	Name	Description	Reset
31:24	R	RSVD_31_24	These bits of the UHS2_GEN_CAP_R register are reserved. They always return 0.	0x0
23:22	R	BUS_TOPO	Bus Topology This field indicates one of the bus topologies configured by a Host System. This configuration is done by using parameter DWC_MSHC_UHS2_BUS_TOPOLOGY. Values: ■ 0x0 (P2P): P2P Connection ■ 0x1 (RING): Ring Connection ■ 0x2 (HUB): HUB Connection ■ 0x3 (HUB_IN_RING): HUB is connected in Ring	0x0
21:18	R	NUM_DEV	Number of devices supported This field indicates the maximum number of devices supported by the Host Controller. ■ 0x0: Not used ■ 0x1: 1 Device ■ 0x2: 2 Devices ■ ■ ■ ■ 0xF: 15 Devices	0x1

Bits	Access	Name	Description	Reset
17:16	R	RMV_EMBEDDED	Removable/ Embedded This field indicates device type configured by a Host System. Values: ■ 0x0 (RMV_CARD): Removable card ■ 0x1 (EMB_CARD): Embedded Devices ■ 0x2 (RMV_EMB_CARD): Embedded Devices and Removable card ■ 0x3 (RSVD): Reserved	0x0
15	R	BOOT_LOAD	Boot Code Loading This field indicates whether the Host Controller tries to boot the system in the UHS-II mode. If this bit is set to 1, the Synchronization for Boot Code Loading (BSYN) Link Symbol Set (LSS) is sent during the PHY Initialization. Values: ■ 0x1 (TRUE): Execute Boot Code Loading ■ 0x0 (FALSE): No Boot Code Loading	0x0
14	R	ADDR64	64-bit Addressing This field indicates support of 64-bit addressing by the Host Controller. Values: ■ 0x1 (TRUE): 32-bit and 64-bit Addressing is supported ■ 0x0 (FALSE): 32-bit Addressing is supported	0x0
13:8	R	NUM_LANE	Number of Lanes and Functionalities This field indicates support for lanes by the Host Controller. The value 1 indicates the support while 0 means, it is not supported. ■ D08: 2L-HD ■ D09: 2D1U-FD ■ D10: 1D2U-FD ■ D11: 2D2U-FD ■ D12: Reserved ■ D13: Reserved	0x0
7:4	R	GAP	GAP (Group Allocation Power) This field indicates the maximum capability of host power supply for a group configured by a Host System. This field is used to set the argument of the DEVICE_INIT CCMD. This configuration is done by using the DWC_MSHC_UHS2_GAP parameter. ■ 0x0: Not used ■ 0x1: 360 mW ■ 0x2: 720 mW ■ ■ ■ ■ 0xF: 360x15 mW	0x0

Bits	Access	Name	Description	Reset
3:0	R	DAP	DAP (Device Allocation Power) This field indicates the maximum capability of host power supply for a device configured by a Host System. This field is used to set the argument of the DEVICE_INIT CCMD. This configuration is done by using the DWC_MSHC_UHS2_DAP parameter. ■ 0x0: 360 mW (Default) ■ 0x1: 360 mW ■ 0x2: 720 mW ■ ■ ■ ■ 0xF: 360x15 mW	0x0

UHS2_PHY_CAP_R

UHS2_PHY_CAP_R is an UHS2 PHY Capabilities register Offset Address : P_UHS_II_HOST_CAPAB[11:0] + 0x4
Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
31:24	R	RSVD_31_24	These bits of the UHS2_PHY_CAP_R register are reserved. They always return 0.	0x0
23:20	R	N_LSS_DIR	Host N_LSS_DIR This field indicates the minimum N_LSS_DIR required by the Host Controller. This is configured by using the DWC_MSHC_PHY_N_LSS_DIR parameter. ■ 0x0: 8x16 LSS ■ 0x1: 8x1 LSS ■ 0x2: 8x2 LSS ■ ■ ■ ■ 0xF: 8x15 LSS	0x0
19:16	R	N_LSS_SYN	Host N_LSS_SYN This field indicates the minimum N_LSS_SYN required by the Host Controller. This is configured by using the DWC_MSHC_PHY_N_LSS_SYN parameter. ■ 0x0: 4x16 LSS ■ 0x1: 4x1 LSS ■ 0x2: 4x2 LSS ■ ■ ■ ■ 0xF: 4x15 LSS	0x0
15:8	R	RSVD_15_8	These bits of the UHS2_PHY_CAP_R register are reserved bits. They always return 0.	0x0

Bits	Access	Name	Description	Reset
7:6	R	SPD_RANGE	Speed Range This field indicates supported Speed Range by the Host Controller. This is configured using the DWC_MSHC_PHY_SPD_RANGE parameter. Values: ■ 0x0 (RANGE_A): Range A (default) ■ 0x1 (RANGE_B): Range A and Range B ■ 0x2 (RSVD1): Reserved ■ 0x3 (RSVD2): Reserved	0x0
5:0	R	PHY_REV	PHY Revision These bits indicate the supported PHY Revision number. ■ 0x0 : First Revision ■ others : Reserved	0x0

UHS2_LNK_TRAN_CAP_1_R

UHS2_LNK_TRAN_CAP_1_R is an UHS2 Link Tran Capabilities register (0 to 31) Offset Address :

P_UHS_II_HOST_CAPAB[11:0] + 0x8 Total Reset Value: 0x80000200

Bits	Access	Name	Description	Reset
31:20	R	MAX_BLK_LEN	Host Maximum Block Length These bits indicate the maximum block length supported by the Host Controller for UHS-II mode. ■ 0x000: Not used ■ 0x001: 1 byte ■ 0x002: 2 bytes ■ ■ ■ ■ 0x200: 512 bytes ■ ■ ■ ■ 0x800: 2048 bytes ■ 0x801-0xFFFF: Not used	0x800
19	R	RSVD_19	This bit of the UHS2_LINK_TRAN_CAP_1_R register is reserved. It always returns 0.	0x0
18:16	R	DEV_TYPE	Host Device Type This field is fixed to 0h. 0x0: Host Controller Others: Not Used	0x0

Bits	Access	Name	Description	Reset
15:8	R	N_FCU	Host N_FCU These bits indicate the maximum number of blocks supported in a flow control unit by the Host Controller. This value is determined by the supported packet buffer depth (DWC_MSHC_PKT_BUFFER_DEPTH), the AXI data width (DWC_MSHC_MBIU_DW) and the maximum block size (DWC_MSHC_LINK_MAX_BLK_SIZE). ■ 0x00: 256 Blocks ■ 0x01: 1 Block ■ 0x02: 2 Blocks ■ ■ ■ ■ 0xFF: 255 Blocks Note: The value of this field is calculated by following formula: $DWC\_MSHC\_PKT\_BUFFER\_DEPTH / (2 * (DWC\_MSHC\_LINK\_MAX\_BLK\_SIZE / (DWC\_MSHC\_MBIU\_DW/8)))$	0x2
7:6	R	RSVD_7_6	These bits of the UHS2_LINK_TRAN_CAP_1_R register are reserved. They always return 0.	0x0
5:0	R	LNK_REV	Link Revision This field indicates the LINK Revision number. ■ 0x0: First revision ■ others: Reserved	0x0

UHS2_LNK_TRAN_CAP_2_R

UHS2_LNK_TRAN_CAP_2_R is an UHS2 Link Tran Capabilities register (32 to 63) Offset Address :

P_UHS_II_HOST_CAPAB[11:0] + 0xc Total Reset Value: 0x00000004

Bits	Access	Name	Description	Reset
31:8	R	RSVD_31_8	These bits of the UHS2_LNK_TRAN_CAP_2_R register are reserved. They always return 0.	0x0
7:0	R	N_DATA_GAP	Host N_DATA_GAP These bits indicate the minimum number of data gap (DIDL) supported by the Host Controller. This is configured by using the DWC_MSHC_LINK_N_DATA_GAP parameter. ■ 0x00 : No Gap ■ 0x01 : 1 LSS ■ 0x02 : 2 LSS ■ ■ ■ ■ 0xFF : 255 LSS	0x4

FORCE_UHS_II_ERR_INTR_STATUS_R

FORCE_UHS_II_ERR_INTR_STATUS_R is a Force Event for UHS-II Error Interrupt Status Register Offset Address :
P_UHS_II_TEST[11:0] Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
31	W	VENDOR_SPECIFIC_ERR4	This bit (VENDOR_SPECIFIC_ERR4) of the FORCE_UHS_II_ERR_INTR_STATUS_R register is reserved. It always returns 0. Values: ■ 0x0 (FALSE): Not Affected ■ 0x1 (TRUE): Vendor Specific Error status is set	0x0
30	W	VENDOR_SPECIFIC_ERR3	This bit (VENDOR_SPECIFIC_ERR3) of the FORCE_UHS_II_ERR_INTR_STATUS_R register is reserved. It always returns 0. Values: ■ 0x0 (FALSE): Not Affected ■ 0x1 (TRUE): Vendor Specific Error status is set	0x0
29	W	VENDOR_SPECIFIC_ERR2	This bit (VENDOR_SPECIFIC_ERR2) of the FORCE_UHS_II_ERR_INTR_STATUS_R register is reserved. It always returns 0. Values: ■ 0x0 (FALSE): Not Affected ■ 0x1 (TRUE): Vendor Specific Error status is set	0x0
28	W	VENDOR_SPECIFIC_ERR1	This bit (VENDOR_SPECIFIC_ERR1) of the FORCE_UHS_II_ERR_INTR_STATUS_R register is reserved. It always returns 0. Values: ■ 0x0 (FALSE): Not Affected ■ 0x1 (TRUE): Vendor Specific Error status is set	0x0
27	W	VENDOR_SPECIFIC_ERR0	This bit (VENDOR_SPECIFIC_ERR0) of the FORCE_UHS_II_ERR_INTR_STATUS_R register is reserved. It always returns 0. Values: ■ 0x0 (FALSE): Not Affected ■ 0x1 (TRUE): Vendor Specific Error status is set	0x0
26:18	R	RESERVED_26_18	These bits (RESERVED_26_18) of the FORCE_UHS_II_ERR_INTR_STATUS_R register are reserved. They always return 0.	0x0
17	W	TIMEOUT_DEADLOCK	Force Event for Timeout for Deadlock This bit forces the Host Controller to set Timeout for Deadlock in the UHS-II Error Interrupt Status register. Values: ■ 0x0 (FALSE): Not Affected ■ 0x1 (TRUE): Timeout for Deadlock Error status is set	0x0

Bits	Access	Name	Description	Reset
16	W	TIMEOUT_CMD_RES	Force Event for Timeout for CMD_RES This bit forces the Host Controller to set Timeout for CMD_RES in the UHS-II Error Interrupt Status register. Values: ■ 0x0 (FALSE): Not Affected ■ 0x1 (TRUE): Timeout for CMD_RES Error status is set	0x0
15	W	AMDA_ERR	Force Event for ADMA Error This bit forces the Host Controller to set ADMA Error in the UHS-II Error Interrupt Status register. Values: ■ 0x0 (FALSE): Not Affected ■ 0x1 (TRUE): ADMA Error status is set	0x0
14:9	R	RESERVED_14_9	These bits (RESERVED_14_9) of the FORCE_UHS_II_ERR_INTR_STATUS_R register are reserved. They always return 0.	0x0
8	W	EBSY_ERR	Force Event for EBSY Error This bit forces the Host Controller to set EBSY Error in the UHS-II Error Interrupt Status register. Values: ■ 0x0 (FALSE): Not Affected ■ 0x1 (TRUE): EBSY Error status is set	0x0
7	W	UNRECOVERABLE_ERR	Force Event for Unrecoverable Error This bit forces the Host Controller to set Unrecoverable Error in the UHS-II Error Interrupt Status register. Values: ■ 0x0 (FALSE): Not Affected ■ 0x1 (TRUE): Unrecoverable Error status is set	0x0
6	R	RESERVED_6	This bit (RESERVED_6) of the FORCE_UHS_II_ERR_INTR_STATUS_R register is reserved. It always returns 0.	0x0
5	W	TID_ERR	Force Event for TID Error This bit forces the Host Controller to set TID Error in the UHS-II Error Interrupt Status register. Values: ■ 0x0 (FALSE): Not Affected ■ 0x1 (TRUE): TID Error status is set	0x0
4	W	FRAMING_ERR	Force Event for Framing Error This bit forces the Host Controller to set Framing Error in the UHS-II Error Interrupt Status register. Values: ■ 0x0 (FALSE): Not Affected ■ 0x1 (TRUE): Framing Error status is set	0x0

Bits	Access	Name	Description	Reset
3	W	CRC_ERR	Force Event for CRC Error This bit forces the Host Controller to set CRC Error in the UHS-II Error Interrupt Status register. Values: ■ 0x0 (FALSE): Not Affected ■ 0x1 (TRUE): CRC Error status is set	0x0
2	W	RETRY_EXPIRED	Force Event for Retry Expired This bit forces the Host Controller to set Retry Expired in the UHS-II Error Interrupt Status register. Values: ■ 0x0 (FALSE): Not Affected ■ 0x1 (TRUE): Retry expired Error status is set	0x0
1	W	RES_PKT_ERR	Force Event for RES Packet Error This bit forces the Host Controller to set RES Packet Error in the UHS-II Error Interrupt Status register. Values: ■ 0x0 (FALSE): Not Affected ■ 0x1 (TRUE): RES Packet Error status is set	0x0
0	W	HEADER_ERR	Force Event for Header Error This bit forces the Host Controller to set Header Error in the UHS-II Error Interrupt Status register. Values: ■ 0x0 (FALSE): Not Affected ■ 0x1 (TRUE): Header Error status is set	0x0

EMBEDDED_CTRL_R

EMBEDDED_CTRL_R is an Embedded Control register Offset Address : P_EMBEDDED_CNTRL[11:0] Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
31	R	RSVD_31	This bit (RSVD_31) of the EMBEDDED_CTRL_R register is reserved. It always returns 0.	0x0
30:24	R/W	BACK_END_PWR_CTRL	Back-End Power Control (SD Mode) Each bit of this field controls back-end power supply for an embedded device. ■ 0 : Back-End Power is off ■ 1 : Back-End Power is supplied D24 : Back-End Power for Device 1 D25 : Back-End Power for Device 2 D26 : Back-End Power for Device 3 D27 : Back-End Power for Device 4 D28 : Back-End Power for Device 5 D29 : Back-End Power for Device 6 D30 : Back-End Power for Device 7	0x0

Bits	Access	Name	Description	Reset
23	R	RSVD_23	This bit (RSVD_23) of the EMBEDDED_CTRL_R register is reserved. It always returns 0.	0x0
22:20	R/W	INT_PIN_SEL	Interrupt Pin Select These bits enable the interrupt pin inputs. ■ 000 : Interrupts (INT_A,INT_B,INT_C) are disabled ■ xx1 : INT_A is enabled ■ x1x : INT_B is enabled ■ 1xx : INT_C is enabled	0x0
19	R	RSVD_19	This bit (RSVD_19) of the EMBEDDED_CTRL_R register is reserved. It always returns 0.	0x0
18:16	R/W	CLK_PIN_SEL	Clock Pin Select (SD Mode) This bit is selected by one of clock pin outputs. ■ 0x0 : Clock pins are disabled ■ 0x1 : CLK[1] is selected ■ 0x2 : CLK[2] is selected ■ . . ■ . . ■ . . ■ 0x7 : CLK[7] is selected	0x0
15	R	RSVD_15	This bit (RSVD_15) of the EMBEDDED_CTRL_R register is reserved. It always returns 0.	0x0
14:8	R	BUS_WIDTH_PRESET	Bus Width Preset (SD Mode) Each bit of this field specifies the bus width for each embedded device. The shared bus supports mixing of 4-bit and 8-bit bus width devices. ■ D08 : Bus Width Preset for Device 1 ■ D09 : Bus Width Preset for Device 2 ■ D10 : Bus Width Preset for Device 3 ■ D11 : Bus Width Preset for Device 4 ■ D12 : Bus Width Preset for Device 5 ■ D13 : Bus Width Preset for Device 6 ■ D14 : Bus Width Preset for Device 7 Function of each bit is defined as follows: ■ 0 : 4-bit bus width mode ■ 1 : 8-bit bus width mode	0x0
7:6	R	RSVD_7_6	These bits (RSVD_7_6) of the EMBEDDED_CTRL_R register are reserved. They always return 0.	0x0
5:4	R	NUM_INT_PIN	Number of Interrupt Input Pins This field indicates support of interrupt input pins for an embedded system.	0x0
3	R	RSVD_3	This bit (RSVD_3) of the EMBEDDED_CTRL_R register is reserved. It always returns 0.	0x0

Bits	Access	Name	Description	Reset
2:0	R	NUM_CLK_PIN	Number of Clock Pins (SD Mode) This field indicates support of clock pins to select one of devices for shared bus system. Up to 7 clock pins can be supported. ■ 0x0 : Shared bus is not supported ■ 0x1 : 1 SDCLK is supported ■ 0x2 - 2 SDCLK is supported ■ . . ■ . . ■ . . ■ 0x7 : 7 SDCLK is supported	0x0

CQVER

CQVER is a Command Queuing Version register Offset Address : P_VENDOR2_SPECIFIC_AREA Total Reset Value: 0x00000510

Bits	Name	Access	Description	Reset
31:12	R	EMMMC_VER_RSVD	These bits of the CQVER register are reserved. They always return 0.	0x0
11:8	R	EMMC_VER_MAJOR	This bit indicates the eMMC major version (1st digit left of decimal point) in BCD format.	0x5
7:4	R	EMMC_VER_MINOR	This bit indicates the eMMC minor version (1st digit right of decimal point) in BCD format.	0x1
3:0	R	EMMC_VER_SUFFIX	This bit indicates the eMMC version suffix (2nd digit right of decimal point) in BCD format.	0x0

CQCAP

CQCAP is a Command Queuing Capabilities register Offset Address : P_VENDOR2_SPECIFIC_AREA + 0x4 Total Reset Value: 0x00003014

Bits	Access	Name	Description	Reset
31:29	R	CQCCAP_RSVD3	These bits [31:29] of the CQCAP register are reserved. They always return 0.	0x0
28	R	CRYPTO_SUPPORT	Crypto Support This bit indicates whether the Host Controller supports cryptographic operations. Values: ■ 0x0 (FALSE): Crypto not Supported ■ 0x1 (TRUE): Crypto Supported	0x0
27:16	R	CQCCAP_RSVD2	These bits [27:16] of the CQCAP register are reserved. They always return 0.	0x0

Bits	Access	Name	Description	Reset
15:12	R	ITCFMUL	Internal Timer Clock Frequency Multiplier (ITCFMUL) This field indicates the frequency of the clock used for interrupt coalescing timer and for determining the SQS polling period. See ITCFVAL definition for details. Values 0x5 to 0xF are reserved. Values: ■ 0x0 (CLK_1KHz): 1KHz clock ■ 0x1 (CLK_10KHz): 10KHz clock ■ 0x2 (CLK_100KHz): 100KHz clock ■ 0x3 (CLK_1MHz): 1MHz clock ■ 0x4 (CLK_10MHz): 10MHz clock	0x3
11:10	R	CQCCAP_RSVD1	These bits of the CQCAP register are reserved. They always return 0.	0x0
9:0	R	ITCFVAL	Internal Timer Clock Frequency Value (ITCFVAL) This field scales the frequency of the timer clock provided by ITCFMUL. The Final clock frequency of actual timer clock is calculated as ITCFVAL* ITCFMUL.	0x14

CQCFG

CQCFG is a Command Queuing Configuration register Offset Address : P_VENDOR2_SPECIFIC_AREA + 0x8 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
31:13	R	CQCCFG_RSVD3	These bits (CQCCFG_RSVD3) of the CQCFG register are reserved. They always return 0.	0x0
12	R/W	DCMD_EN	This bit indicates to the hardware whether the Task Descriptor in slot #31 of the TDL is a data transfer descriptor or a direct-command descriptor. CQE uses this bit when a task is issued in slot #31, to determine how to decode the Task Descriptor. Values: ■ 0x1 (SLOT31_DCMD_ENABLE): Task descriptor in slot #31 is a DCMD Task Descriptor ■ 0x0 (SLOT31_DCMD_DISABLE): Task descriptor in slot #31 is a data Transfer Task Descriptor	0x0
11:9	R	CQCCFG_RSVD2	These bits (CQCCFG_RSVD2) of the CQCFG register are reserved. They always return 0.	0x0
8	R/W	TASK_DESC_SIZE	Bit Value Description This bit indicates the size of task descriptor used in host memory. This bit can only be configured when Command Queuing Enable bit is 0 (command queuing is disabled). Values: ■ 0x1 (TASK_DESC_128b): Task descriptor size is 128 bits ■ 0x0 (TASK_DESC_64b): Task descriptor size is 64 bits	0x0
7:2	R	CQCCFG_RSVD1	These bits (CQCCFG_RSVD1) of the CQCFG register are reserved. They always return 0.	0x0

Bits	Access	Name	Description	Reset
1	R/W	CR_GENERAL_EN	Crypto General Enable Enable/Disable bit for Crypto Engine. If cryptographic operations are not supported, this status bit is reserved. Values: ■ 0x1 (ENABLE): Enable cryptographic operations for transactions where TD.CE=1 or CRNQP.CE=1 ■ 0x0 (DISABLE): Disable cryptographic operations for all transactions	0x0
0	R/W	CQ_EN	Enable command queuing engine (CQE). When CQE is disable, the software controls the eMMC bus using the registers between the addresses 0x000 to 0x1FF. Before the software writes to this bit, the software verifies that the eMMC host controller is in idle state and there are no ongoing commands or data transfers. When software wants to exit command queuing mode, it clears all previous tasks (if any) before setting this bit to 0. Values: ■ 0x1 (CQE_ENABLE): Enable command queuing ■ 0x0 (CQE_DISABLE): Disable command queuing	0x0

CQCTL

CQCTL is a Command Queuing Control register Offset Address : P_VENDOR2_SPECIFIC_AREA + 0xc Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
31:9	R	CQCTL_RSVD2	These bits (CQCTL_RSVD2) of the CQCTL register are reserved. They always return 0.	0x0
8	R/W	CLR_ALL_TASKS	Clear all tasks This bit can only be written when the controller is halted. This bit does not clear tasks in the device. The software has to use the CMDQ_TASK_MGMT command to clear device's queue. Values: ■ 0x1 (CLEAR_ALL_TASKS): Clears controller ■ 0x0 (NO_EFFECT): Programming 0 has no effect	0x0
7:1	R	CQCTL_RSVD1	These bits (CQCTL_RSVD1) of the CQCTL register are reserved. They always return 0.	0x0

Bits	Access	Name	Description	Reset
0	R/W	HALT	Halt request and resume Values: ■ 0x1 (HALT_CQE): Software writes 1 to this bit when it wants to acquire software control over the eMMC bus and to disable CQE from issuing command on the bus. For example, issuing a Discard Task command (CMDQ_TASK_MGMT). When the software writes 1, CQE completes the ongoing task (if any in progress). After the task is completed and the CQE is in idle state, CQE does not issue new commands and indicates to the software by setting this bit to 1. The software can poll on this bit until it is set to 1 and only then send commands on the eMMC bus. ■ 0x0 (RESUME_CQE): Software writes 0 to this bit to exit from the halt state and resume CQE activity.	0x0

CQIS

CQIS is a Command Queuing Interrupt Status register Offset Address : P_VENDOR2_SPECIFIC_AREA + 0x10 Total Reset Value: 0x0

Bits	Name	Access	Description	Reset
31:6	R	CQIS_RSVD1	These bits of the CQIS register are reserved. They always return 0.	0x0
5	R/W1C	ICCE	Invalid Crypto Configuration Error If cryptographic operations is supported by host controller (CQCAP.CS=1) and (CQCFG.GCE=1); and encryption/decryption is enabled (TD.CE=1), this status bit is asserted (if CQISE.ICCE_STE=1) if either of the below conditions are met. ■ CRYPTOCFG disabled (CRYPTOCFG_i_16[TD.CCI].CFGE_i=0) OR nonexistent (TD.CCI>=CRCAP.CFGC) ■ Capability index in Configuration (CRYPTOCFG_i_16[TD.CCI].CAPIDX_i) is greater than maximum number of capabilities supported (CAPIDX_i > CRCAP.CC) ■ Data unit size in Configuration (CRYPTOCFG_i_16[TD.CCI].DUSIZE_i) is not supported by selected capability (CRYPTOCAP_x[CAPIDX_i].SDUSB) If cryptographic operations are not supported, this status bit is reserved. Values: ■ 0x1 (SET): ICCE interrupt is set ■ 0x0 (UNSET): ICCE interrupt is not set	0x0

Bits	Name	Access	Description	Reset
4	R/W1C	GCE	General Crypto Error If cryptographic operations is supported by host controller (CQCAP.CS=1) and (CQCFG.GCE=1); and encryption/decryption is enabled (TD.CE=1), this status bit is asserted (if CQISE.GCE_STE=1) when the crypto hardware encounters any of the following error in the processing of a transaction. ■ Block size is not multiple of 16 bytes ■ Total transfer length (Block size * Block count) is not an integral multiple of DUSize ■ Device data block transfer starting address is not aligned to DUSize ■ CQE task descriptor data transfer length is not multiple of 16 bytes If cryptographic operations are not supported, this status bit is reserved. Values: ■ 0x1 (SET): GCE interrupt is set ■ 0x0 (UNSET): GCE interrupt is not set	0x0
3	R/W1C	TCL	Task cleared interrupt This status bit is asserted (if CQISE.TCL_STE=1) when a task clear operation is completed by CQE. The completed task clear operation is either an individual task clear (by writing CQTCLR) or clearing of all tasks (by writing CQCTL). A value of 1 clears this status bit. Values: ■ 0x1 (SET): TCL Interrupt is set ■ 0x0 (NOTSET): TCL Interrupt is not set	0x0
2	R/W1C	RED	Response error detected interrupt This status bit is asserted (if CQISE.RED_STE=1) when a response is received with an error bit set in the device status field. Configure the CQRMEM register to identify device status bit fields that may trigger an interrupt and that are masked. A value of 1 clears this status bit. Values: ■ 0x1 (SET): RED Interrupt is set ■ 0x0 (NOTSET): RED Interrupt is not set	0x0
1	R/W1C	TCC	Task complete interrupt This status bit is asserted (if CQISE.TCC_STE=1) when at least one of the following conditions are met: ■ A task is completed and the INT bit is set in its Task Descriptor ■ Interrupt caused by Interrupt Coalescing logic due to timeout ■ Interrupt Coalescing logic reached the configured threshold A value of 1 clears this status bit Values: ■ 0x1 (SET): TCC Interrupt is set ■ 0x0 (NOTSET): TCC Interrupt is not set	0x0

Bits	Name	Access	Description	Reset
0	R/W1C	HAC	Halt complete interrupt This status bit is asserted (only if CQISE.HAC_STE=1) when halt bit in the CQCTL register transitions from 0 to 1 indicating that the host controller has completed its current ongoing task and has entered halt state. A value of 1 clears this status bit. Values: ■ 0x1 (SET): HAC Interrupt is set ■ 0x0 (NOTSET): HAC Interrupt is not set	0x0

CQISE

CQISE is a Command Queuing Interrupt Status Enable register Offset Address : P_VENDOR2_SPECIFIC_AREA + 0x14 Total Reset Value: 0x0

Bits	Name	Access	Description	Reset
31:6	R	CQISTE_RSVD1	These bits of the CQISE register are reserved. They always return 0.	0x0
5	R/W	ICCE_STE	Invalid Crypto Configuration Error interrupt status enable Values: ■ 0x1 (INT_STS_ENABLE): CQIS.ICCE is set when its interrupt condition is active ■ 0x0 (INT_STS_DISABLE): CQIS.ICCE is disabled	0x0
4	R/W	GCE_STE	General Crypto Error interrupt status enable Values: ■ 0x1 (INT_STS_ENABLE): CQIS.GCE is set when its interrupt condition is active ■ 0x0 (INT_STS_DISABLE): CQIS.GCE is disabled	0x0
3	R/W	TCL_STE	Task cleared interrupt status enable Values: ■ 0x1 (INT_STS_ENABLE): CQIS.TCL is set when its interrupt condition is active ■ 0x0 (INT_STS_DISABLE): CQIS.TCL is disabled	0x0
2	R/W	RED_STE	Response error detected interrupt status enable Values: ■ 0x1 (INT_STS_ENABLE): CQIS.RED is set when its interrupt condition is active ■ 0x0 (INT_STS_DISABLE): CQIS.RED is disabled	0x0
1	R/W	TCC_STE	Task complete interrupt status enable Values: ■ 0x1 (INT_STS_ENABLE): CQIS.TCC is set when its interrupt condition is active ■ 0x0 (INT_STS_DISABLE): CQIS.TCC is disabled	0x0
0	R/W	HAC_STE	Halt complete interrupt status enable Values: ■ 0x1 (INT_STS_ENABLE): CQIS.HAC is set when its interrupt condition is active ■ 0x0 (INT_STS_DISABLE): CQIS.HAC is disabled	0x0

CQISGE

CQISGE is a Command Queuing Interrupt signal enable register Offset Address : P_VENDOR2_SPECIFIC_AREA + 0x18 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
31:6	R	CQISGE_RSVD1	These bits of the CQISGE register are reserved. They always return 0.	0x0
5	R/W	ICCE_SGE	Invalid Crypto Configuration Error interrupt signal enable Values: ■ 0x1 (INT_SIG_ENABLE): CQIS.ICCE interrupt signal generation is active ■ 0x0 (INT_SIG_DISABLE): CQIS.ICCE interrupt signal generation is disabled	0x0
4	R/W	GCE_SGE	General Crypto Error interrupt signal enable Values: ■ 0x1 (INT_SIG_ENABLE): CQIS.GCE interrupt signal generation is active ■ 0x0 (INT_SIG_DISABLE): CQIS.GCE interrupt signal generation is disabled	0x0
3	R/W	TCL_SGE	Task cleared interrupt signal enable Values: ■ 0x1 (INT_SIG_ENABLE): CQIS.TCL interrupt signal generation is active ■ 0x0 (INT_SIG_DISABLE): CQIS.TCL interrupt signal generation is disabled	0x0
2	R/W	RED_SGE	Response error detected interrupt signal enable Values: ■ 0x1 (INT_SIG_ENABLE): CQIS.RED interrupt signal generation is active ■ 0x0 (INT_SIG_DISABLE): CQIS.RED interrupt signal generation is disabled	0x0
1	R/W	TCC_SGE	Task complete interrupt signal enable Values: ■ 0x1 (INT_SIG_ENABLE): CQIS.TCC interrupt signal generation is active ■ 0x0 (INT_SIG_DISABLE): CQIS.TCC interrupt signal generation is disabled	0x0
0	R/W	HAC_SGE	Halt complete interrupt signal enable Values: ■ 0x1 (INT_SIG_ENABLE): CQIS.HAC interrupt signal generation is active ■ 0x0 (INT_SIG_DISABLE): CQIS.HAC interrupt signal generation is disabled	0x0

CQIC

CQIC is a Command Queuing Interrupt Coalescing register Offset Address : P_VENDOR2_SPECIFIC_AREA + 0x1c
Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
31	R/W	INTC_EN	Interrupt Coalescing Enable Bit Values: ■ 0x1 (ENABLE_INT_COALESCING): Interrupt coalescing mechanism is active. Interrupts are counted and timed, and coalesced interrupts are generated ■ 0x0 (DISABLE_INT_COALESCING): Interrupt coalescing mechanism is disabled (Default).	0x0
30:21	R	CQIC_RSVD3	These bits (CQIC_RSVD3) of the CQIC register are reserved. They always return 0.	0x0
20	R	INTC_STAT	Interrupt Coalescing Status Bit This bit indicates to the software whether any tasks (with INT=0) have completed and counted towards interrupt coalescing (that is, this is set if and only if INTC counter > 0). Values: ■ 0x1 (INTC_ATLEAST1_COMP): At least one INT0 task completion has been counted (INTC counter > 0) ■ 0x0 (INTC_NO_TASK_COMP): INT0 Task completions have not occurred since last counter reset (INTC counter == 0)	0x0
19:17	R	CQIC_RSVD2	These bits (CQIC_RSVD2) of the CQIC register are reserved. They always return 0.	0x0
16	W	INTC_RST	Counter and Timer Reset When host driver writes 1, the interrupt coalescing timer and counter are reset. Values: ■ 0x1 (ASSERT_INTC_RESET): Interrupt coalescing timer and counter are reset ■ 0x0 (NO_EFFECT): No Effect	0x0
15	W	INTC_TH_WEN	Interrupt Coalescing Counter Threshold Write Enable When software writes 1 to this bit, the value INTC_TH is updated with the contents written on the same cycle. Values: ■ 0x1 (WEN_SET): Sets INTC_TH_WEN ■ 0x0 (WEN_CLR): Clears INTC_TH_WEN	0x0
14:13	R	CQIC_RSVD1	These bits (CQIC_RSVD1) of the CQIC register are reserved. They always return 0.	0x0

Bits	Access	Name	Description	Reset
12:8	W	INTC_TH	Interrupt Coalescing Counter Threshold field Software uses this field to configure the number of task completions (only tasks with INT=0 in the Task Descriptor), which are required in order to generate an interrupt. Counter Operation: As data transfer tasks with INT=0 complete, they are counted by CQE. The counter is reset by software during the interrupt service routine. The counter stops counting when it reaches the value configured in INTC_TH, and generates interrupt. ■ 0x0: Interrupt coalescing feature disabled ■ 0x1: Interrupt coalescing interrupt generated after 1 task when INT=0 completes ■ 0x2: Interrupt coalescing interrupt generated after 2 tasks when INT=0 completes ■ ■ 0x1f: Interrupt coalescing interrupt generated after 31 tasks when INT=0 completes To write to this field, the INTC_TH_WEN bit must be set during the same write operation.	0x0
7	W	TOUT_VAL_WEN	When software writes 1 to this bit, the value TOUT_VAL is updated with the contents written on the same cycle. Values: ■ 0x1 (WEN_SET): Sets TOUT_VAL_WEN ■ 0x0 (WEN_CLR): clears TOUT_VAL_WEN	0x0
6:0	R/W	TOUT_VAL	Interrupt Coalescing Timeout Value Software uses this field to configure the maximum time allowed between the completion of a task on the bus and the generation of an interrupt. Timer Operation: The timer is reset by software during the interrupt service routine. It starts running when the first data transfer task with INT=0 is completed, after the timer was reset. When the timer reaches the value configured in ICTOVAL field, it generates an interrupt and stops. The timer's unit is equal to 1024 clock periods of the clock whose frequency is specified in the Internal Timer Clock Frequency field CQCAP register. ■ 0x0: Timer is disabled. Timeout-based interrupt is not generated ■ 0x2: Timeout on 02x1024 cycles of timer clock frequency ■ ■ 0x7f: Timeout on 127x1024 cycles of timer clock frequency In order to write to this field, the TOUT_VAL_WEN bit must be set at the same write operation.	0x0

CQTDLBA

CQTDLBA is a Command Queuing Task Descriptor List Base Address register Offset Address :
P_VENDOR2_SPECIFIC_AREA + 0x20 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset

Bits	Access	Name	Description	Reset
31:0	R/W	TDLBA	This register stores the LSB bits (31:0) of the byte address of the head of the Task Descriptor List in system memory. The size of the task descriptor list is 32 * (Task Descriptor size + Transfer Descriptor size) as configured by the host driver. This address is set on 1 KB boundary. The lower 10 bits of this register are set to 0 by the software and are ignored by CQE.	0x0

CQTDLBAU

CQTDLBAU is a Command Queuing Task Descriptor List Base Address Upper register Offset Address : P_VENDOR2_SPECIFIC_AREA + 0x24 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
31:0	R/W	TDLBAU	This register stores the MSB bits (63:32) of the byte address of the head of the Task Descriptor List in system memory. The size of the task descriptor list is 32 * (Task Descriptor size + Transfer Descriptor size) as configured by Host driver. This address is set on 1 KB boundary. The lower 10 bits of this register are set to 0 by the software and are ignored by CQE. This register is reserved when using 32-bit addressing mode.	0x0

CQTDBR

CQTDBR is a Command Queuing DoorBell register Offset Address : P_VENDOR2_SPECIFIC_AREA + 0x28 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
31:0	R/W	DBR	The software configures TDLBA and TDLBAU, and enable CQE in CQCFG before using this register. Writing 1 to bit n of this register triggers CQE to start processing the task encoded in slot n of the TDL. Writing 0 by the software does not have any impact on the hardware, and does not change the value of the register bit. CQE always processes tasks according to the order submitted to the list by CQTDBR write transactions. CQE processes Data Transfer tasks by reading the Task Descriptor and sending QUEUED_TASK_PARAMS (CMD44) and QUEUED_TASK_ADDRESS (CMD45) commands to the device. CQE processes DCMD tasks (in slot #31, when enabled) by reading the Task Descriptor, and generating the command encoded by its index and argument. The corresponding bit is cleared to 0 by CQE in one of the following events: ■ A task execution is completed (with success or error). ■ The task is cleared using CQTCLR register. ■ All tasks are cleared using CQCTL register. ■ CQE is disabled using CQCFG register. Software may initiate multiple tasks at the same time (batch submission) by writing 1 to multiple bits of this register in the same transaction. In the case of batch submission, CQE processes the tasks in order of the task index, starting with the lowest index. If one or more tasks in the batch are marked with QBR, the ordering of execution is based on said processing order.	0x0

CQTCN

CQTCN is a Command Queuing TaskClear Notification register Offset Address : P_VENDOR2_SPECIFIC_AREA + 0x2c Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
31:0	R/W1C	TCN	Task Completion Notification Each of the 32 bits are bit mapped to the 32 tasks. ■ Bit-N(1): Task-N has completed execution (with success or errors) ■ Bit-N(0): Task-N has not completed, could be pending or not submitted. On task completion, software may read this register to know tasks that have completed. After reading this register, software may clear the relevant bit fields by writing 1 to the corresponding bits.	0x0

CQDQS

CQDQS is a Device queue status register Offset Address : P_VENDOR2_SPECIFIC_AREA + 0x30 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
31:0	R	DQS	Device Queue Status Each of the 32 bits are bit mapped to the 32 tasks. ■ Bit-N(1): Device has marked task N as ready for execution ■ Bit-N(0): Task-N is not ready for execution. This task could be pending in device or not submitted. Host controller updates this register with response of the Device Queue Status command.	0x0

CQDPT

CQDPT is a Device pending tasks register Offset Address : P_VENDOR2_SPECIFIC_AREA + 0x34 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
31:0	R	DPT	Device-Pending Tasks Each of the 32 bits are bit mapped to the 32 tasks. ■ Bit-N(1): Task-N has been successfully queued into the device and is awaiting execution ■ Bit-N(0): Task-N is not yet queued. Bit n of this register is set if and only if QUEUED_TASK_PARAMS (CMD44) and QUEUED_TASK_ADDRESS (CMD45) were sent for this specific task and if this task has not been executed. The controller sets this bit after receiving a successful response for CMD45. CQE clears this bit after the task has completed execution. Software reads this register in the task-discard procedure to determine if the task is queued in the device.	0x0

CQTCLR

CQTCLR is a Command Queuing DoorBell register Offset Address : P_VENDOR2_SPECIFIC_AREA + 0x38 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset

Bits	Access	Name	Description	Reset
31:0	R/W	TCLR	Writing 1 to bit n of this register orders CQE to clear a task that the software has previously issued. This bit can only be written when CQE is in Halt state as indicated in CQCFG register Halt bit. When software writes 1 to a bit in this register, CQE updates the value to 1, and starts clearing the data structures related to the task. CQE clears the bit fields (sets a value of 0) in CQTCLR and in CQTDBR once the clear operation is complete. Software must poll on the CQTCLR until it is cleared to verify that a clear operation was done.	0x0

CQSSC1

CQSSC1 is a CQ Send Status Configuration 1 register Offset Address : P_VENDOR2_SPECIFIC_AREA + 0x40 Total Reset Value: 0x00011000

Bits	Access	Name	Description	Reset
31:20	R	RSVD_20_31	These bits of the CQSSC1 register are reserved. They always return 0.	0x0
19:16	R/W	SQSCMD_BLK_CNT	This field indicates when SQS CMD is sent while data transfer is in progress. A value of 'n' indicates that CQE sends status command on the CMD line, during the transfer of data block BLOCK_CNT_n, on the data lines, where BLOCK_CNT is the number of blocks in the current transaction. ■ 0x0: SEND_QUEUE_STATUS (CMD13) command is not sent during the transaction. Instead, it is sent only when the data lines are idle. ■ 0x1: SEND_QUEUE_STATUS command is to be sent during the last block of the transaction. ■ 0x2: SEND_QUEUE_STATUS command when last 2 blocks are pending. ■ 0x3: SEND_QUEUE_STATUS command when last 3 blocks are pending. ■ ■ 0xf: SEND_QUEUE_STATUS command when last 15 blocks are pending. Should be programmed only when CQCFG.CQ_EN is '0'	0x1

Bits	Access	Name	Description	Reset
15:0	R/W	SQSCMD_IDLE_TMR	This field configures the polling period to be used when using periodic SEND_QUEUE_STATUS (CMD13) polling. Periodic polling is used when tasks are pending in the device, but no data transfer is in progress. When a SEND_QUEUE_STATUS response indicates that no task is ready for execution, CQE counts the configured time until it issues the next SEND_QUEUE_STATUS. Timer units are clock periods of the clock whose frequency is specified in the Internal Timer Clock Frequency field CQCAP register. The minimum value is 0001h (1 clock period) and the maximum value is FFFFh (65535 clock periods). For example, a CQCAP field value of 0 indicates a 19.2 MHz clock frequency (period = 52.08 ns). If the setting in CQSSC1.CIT is 1000h, the calculated polling period is $4096 * 52.08 \text{ ns} = 213.33 \text{ ns}$. Should be programmed only when CQCFG.CQ_EN is '0'.	0x1000

CQSSC2

CQSSC2 is a CQ Send Status Configuration 2 register Offset Address : P_VENDOR2_SPECIFIC_AREA + 0x44 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
31:16	R	RSVD_16_31	These bits of the CQSSC2 register are reserved. They always return 0.	0x0
15:0	R/W	SQSCMD_RCA	This field provides CQE with the contents of the 16-bit RCA field in SEND_QUEUE_STATUS (CMD13) command argument. CQE copies this field to bits 31:16 of the argument when transmitting SEND_QUEUE_STATUS (CMD13) command.	0x0

CQCRDCT

CQCRDCT is a Command response for direct command register Offset Address : P_VENDOR2_SPECIFIC_AREA + 0x48 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
31:0	R	DCMD_RESP	This register contains the response of the command generated by the last direct command (DCMD) task that was sent. Contents of this register are valid only after bit 31 of CQTDBR register is cleared by the controller.	0x0

CQRMEM

CQRMEM is a Command response mode error mask register Offset Address : P_VENDOR2_SPECIFIC_AREA + 0x50 Total Reset Value: 0xfdf9a080

Bits	Access	Name	Description	Reset

Bits	Access	Name	Description	Reset
31:0	R/W	RESP_ERR_MASK	The bits of this field are bit mapped to the device response. This bit is used as an interrupt mask on the device status filed that is received in R1/R1b responses. ■ 1: When a R1/R1b response is received, with a bit i in the device status set, a RED interrupt is generated. ■ 0: When a R1/R1b response is received, bit i in the device status is ignored. The reset value of this register is set to trigger an interrupt on all "Error" type bits in the device status. Note: Responses to CMD13 (SQS) encode the QSR so that they are ignored by this logic.	0xfdf9a080

CQTERRI

CQTERRI is a CQ Task Error Information register Offset Address : P_VENDOR2_SPECIFIC_AREA + 0x54 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31	R	TRANS_ERR_FIELDS_VALID	This bit is updated when an error is detected while a data transfer transaction was in progress. Values: ■ 0x1 (SET): data transfer related error detected. Check contents of TRANS_ERR_TASKID and TRANS_ERR_CMD_INDX fields ■ 0x0 (NOT_SET): Ignore contents of TRANS_ERR_TASKID and TRANS_ERR_CMD_INDX	0x0
30:29	R	RSVD_30_29	These bits (RSVD_30_29) of the CQTERRI register are reserved. They always return 0.	0x0
28:24	R	TRANS_ERR_TASKID	This field captures the ID of the task that was executed and whose data transfer has errors.	0x0
23:22	R	RSVD_23_22	These bits (RSVD_23_22) of the CQTERRI register are reserved. They always return 0.	0x0
21:16	R	TRANS_ERR_CMD_INDX	This field captures the index of the command that was executed and whose data transfer has errors.	0x0
15	R	RESP_ERR_FIELDS_VALID	This bit is updated when an error is detected while a command transaction was in progress. Values: ■ 0x1 (SET): Response-related error is detected. Check contents of RESP_ERR_TASKID and RESP_ERR_CMD_INDX fields ■ 0x0 (NOT_SET): Ignore contents of RESP_ERR_TASKID and RESP_ERR_CMD_INDX	0x0
14:13	R	RSVD_13_14	These bits (RSVD_13_14) of the CQTERRI register are reserved. They always return 0.	0x0

Bits	Access	Name	Description	Reset
12:8	R	RESP_ERR_TASKID	This field captures the ID of the task which was executed on the command line when the error occurred.	0x0
7:6	R	RSVD_6_7	These bits (RSVD_6_7) of the CQTERRI register are reserved. They always return 0.	0x0
5:0	R	RESP_ERR_CMD_INDX	This field captures the index of the command that was executed on the command line when the error occurred.	0x0

CQCRI

CQCRI is a CQ Command response index Offset Address : P_VENDOR2_SPECIFIC_AREA + 0x58 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:6	R	RSVD_31_6	These bits of the CQCRI register are reserved. They always return 0.	0x0
5:0	R	CMD_RESP_INDX	Last Command Response index This field stores the index of the last received command response. Controller updates the value every time a command response is received.	0x0

CQCRA

CQCRA is a CQ Command response argument register Offset Address : P_VENDOR2_SPECIFIC_AREA + 0x5c Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:0	R	CMD_RESP_ARG	Last Command Response argument This field stores the argument of the last received command response. Controller updates the value every time a command response is received.	0x0

PHY_CNFG

PHY_CNFG is a SD/eMMC PHY General Configuration Offset Address : DWC_MSHC_PTR_PHY_REGS + 0x0 Total Reset Value:0x0

Bits	Name	Access	Description	Reset
31:24			Reserved Field: Yes	
23:20	R/W	PAD_SN	NMOS TX drive strength control. Common config for all for SD/eMMC Pads.	0x0
19:16	R/W	PAD_SP	PMOS TX drive strength control. Common config for all for SD/eMMC Pads.	0x0
15:2			Reserved Field: Yes	

Bits	Name	Access	Description	Reset
1	R	PHY_PWRGOOD	Phy's Power Good status is captured here. Ensure this is '1' before stating transactions.	0x0
0	R/W	PHY_RSTN	Active-Low reset control for PHY, write '0' to reset PHY, Write '1' to deassert reset.	0x0

CMDPAD_CNFG

CMDPAD_CNFG is a SD/eMMC PHY CMD/RESP PAD Setting Offset Address : DWC_MSHC_PTR_PHY_REGS + 0x4 Total Reset Value:0x0440

Bits	Access	Name	Description	Reset
15:13			Reserved Field: Yes	
12:9	R/W	TXSLEW_CTRL_N	Slew control for N-Type CMD Pad's TX	0x2
8:5	R/W	TXSLEW_CTRL_P	Slew control for P-Type CMD Pad's TX	0x2
4:3	R/W	WEAKPULL_EN	Pull-up/Pul-down enable control for CMD PAD Values: ■ 0x0 (DISABLED): Pull-up and pull-down functionality disabled ■ 0x1 (PULLUP): Weak pull up enabled ■ 0x2 (PULLDOWN): Weak pull down enabled ■ 0x3 (ILLEGAL): Should not be used	0x0
2:0	R/W	RXSEL	Reciver type select for PAD. Controls the RXSEL value of SD/eMMC PHY CMD PAD	0x0

ARGUMENT_R

STBPAD_CNFG is a SD/eMMC PHY Data PAD Setting Offset Address : DWC_MSHC_PTR_PHY_REGS + 0x6 Total Reset Value:0x0440

Bits	Access	Name	Description	Reset
15:13			Reserved Field: Yes	
12:9	R/W	TXSLEW_CTRL_N	Slew control for N-Type DATA Pad's TX	0x2
8:5	R/W	TXSLEW_CTRL_P	Slew control for P-Type DATA Pad's TX	0x2
4:3	R/W	WEAKPULL_EN	Pull-up/Pull-down enable control for DATA PADs Values: ■ 0x0 (DISABLED): Pull-up and pull-down functionality disabled ■ 0x1 (PULLUP): Weak pull up enabled ■ 0x2 (PULLDOWN): Weak pull down enabled ■ 0x3 (ILLEGAL): Should not be used	0x0
2:0	R/W	RXSEL	Reciver type select for PAD. Controls the RXSEL value of SD/eMMC PHY DATA PADs	0x0

ARGUMENT_R**CLKPAD_CNFG ■ Name: SD/eMMC PHY Clock PAD Setting Offset Address :

DWC_MSHC_PTR_PHY_REGS + 0x8 Total Reset Value:0x0440

Bits	Access	Name	Description	Reset

Bits	Access	Name	Description	Reset
15:13			Reserved Field: Yes	
12:9	R/W	TXSLEW_CTRL_N	Slew control for N-Type CLK Pad's TX	0x2
8:5	R/W	TXSLEW_CTRL_P	Slew control for P-Type CLK Pad's TX	0x2
4:3	R/W	WEAKPULL_EN	Pull-up/Pul-down enable control for CLK PAD Values: ■ 0x0 (DISABLED): Pull-up and pull-down functionality disabled ■ 0x1 (PULLUP): Weak pull up enabled ■ 0x2 (PULLDOWN): Weak pull down enabled ■ 0x3 (ILLEGAL): Should not be used	0x0
2:0	R/W	RXSEL	Reciver type select for PAD. Controls the RXSEL value of SD/eMMC PHY CLK PAD	0x0

STBPAD_CNFG**STBPAD_CNFG

■ Name: SD/eMMC PHY Strobe PAD Setting Offset Address : DWC_MSHC_PTR_PHY_REGS + 0xA Total Reset Value:0x0440

Bits	Name	Access	Description	Reset
15:13			Reserved Field: Yes	
12:9	R/W	TXSLEW_CTRL_N	Slew control for N-Type Strobe Pad's TX	0x2
8:5	R/W	TXSLEW_CTRL_P	Slew control for P-Type Strobe Pad's TX	0x2
4:3	R/W	WEAKPULL_EN	Pull-up/Pul-down enable control for STROBE PAD Values: ■ 0x0 (DISABLED): Pull-up and pull-down functionality disabled ■ 0x1 (PULLUP): Weak pull up enabled ■ 0x2 (PULLDOWN): Weak pull down enabled ■ 0x3 (ILLEGAL): Should not be used	0x0
2:0	R/W	RXSEL	Reciver type select for PAD. Controls the RXSEL value of SD/eMMC PHY STROBE PAD	0x0

RSTNPAD_CNFG

RSTNPAD_CNFG is a SD/eMMC PHY RSTN PAD Setting Offset Address : DWC_MSHC_PTR_PHY_REGS + 0xC Total Reset Value:0x0440

Bits	Name	Access	Description	Reset
15:13			Reserved Field: Yes	
12:9	R/W	TXSLEW_CTRL_N	Slew control for N-Type RST_N Pad's TX	0x2
8:5	R/W	TXSLEW_CTRL_P	Slew control for P-Type RST_N Pad's TX	0x2

Bits	Name	Access	Description	Reset
4:3	R/W	WEAKPULL_EN	Pull-up/Pul-down enable control for RST_N PAD(s) Values: ■ 0x0 (DISABLED): Pull-up and pull-down functionality disabled ■ 0x1 (PULLUP): Weak pull up enabled ■ 0x2 (PULLDOWN): Weak pull down enabled ■ 0x3 (ILLEGAL): Should not be used	0x0
2:0	R/W	RXSEL	Reciver type select for PAD. Controls the RXSEL value of SD/eMMC PHY RST_N PAD(s)	0x0

PADTEST_CNFG

PADTEST_CNFG is a SD/eMMC PHY PAD TEST interface Setting Offset Address : DWC_MSHC_PTR_PHY_REGS + 0xE Total Reset Value:0x0

Bits	Name	Access	Description	Reset
15:10			Reserved Field: Yes	
9:4	R/W	TEST_OE	test interface OE control. Drive's PHY's itest_oe inputs.	0x0
3:1	R	RSVD_1	RSVD1 field is reserved Reserved Field: Yes	0x0
0	R/W	TESTMODE_EN	enables test mode interface for all PADS. Functional interface is disabled. Values: ■ 0x0 (PAD_FUNCMODE): PAD's functional mode I/F is active ■ 0x1 (PAD_TESTMODE): PAD's test mode interface is active	0x0

PADTEST_OUT

PADTEST_OUT is a SD/eMMC PHY PAD TEST Data out value Offset Address : DWC_MSHC_PTR_PHY_REGS + 0x10 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
15:6			Reserved Field: Yes	
5:0	R/W	TESTDATA_OUT	Data written here is reflected on corresponding itest_a. Individual bits are mapped to corresponding itest_a inputs of the PHY	0x0

PADTEST_IN

PADTEST_IN is a SD/eMMC PHY PAD TEST Data in value Offset Address : DWC_MSHC_PTR_PHY_REGS + 0x12 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
15:6			Reserved Field: Yes	
5:0	R	TESTDATA_IN	individual bits here capture data available on corresponding otest_y output. should be used for DC test and low freq data patterns.	0x0

PRBS_CNFG

PRBS_CNFG is a Controller PRBS Config register Offset Address : DWC_MSHC_PTR_PHY_REGS + 0x18 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
15:0	R/W	INIT_SEED	Value programmed here is used as SEED for PRBS engine	0xffff

PHYLPBK_CNFG

PHYLPBK_CNFG is a Loopback Config register Offset Address : DWC_MSHC_PTR_PHY_REGS + 0x1A Total Reset Value:0x0

Bits	Access	Name	Description	Reset
7:2			Reserved Field: Yes	
1	R/W	OUT_EN_PHYLPBK_MODE	CMD/DATA output enable in PHY loopback mode Values: ■ 0x1 (OUTPUT_EN): CMD/DATA output is enable PHY Loopback mode ■ 0x0 (OUTPUT_DIS): CMD/DATA output is disable PHY Loopback mode	0x0
0	R/W	PHYLPBK_EN	PHY Local loop back mode is enable Values: ■ 0x1 (PHYLPBK_MODE_EN): Controller is now in PHY Loopback mode ■ 0x0 (PHYLPBK_MODE_DIS): Controller is not in PHY loopback mode	0x0

COMMDL_CNFG

COMMDL_CNFG is a Common DelayLine config settings register Offset Address : DWC_MSHC_PTR_PHY_REGS + 0x1C Total Reset Value:0x0

Bits	Access	Name	Description	Reset
7:2			Reserved Field: Yes	
1	R/W	DLOUT_EN	When '1' DL outputs can be sampled on PADs. Drives idlout_en for all PADs Values: ■ 0x1 (DL_OUTEN): DelayLine outputs on PAD enabled ■ 0x0 (DL_OUTDIS): DelayLine outputs on PAD disabled	0x0
0	R/W	DLSTEP_SEL	DelayLine's per step delay selection, Drives PHY's idl_step input	0x0

SDCLKDL_CNFG

SDCLKDL_CNFG is a SD/eMMC DelayLine settings Offset Address : DWC_MSHC_PTR_PHY_REGS + 0x1D Total Reset Value:0x0

Bits	Access	Name	Description	Reset
7:5			Reserved Field: Yes	

Bits	Access	Name	Description	Reset
4	R/W	UPDATE_DC	Prepares DealyLine for code update when '1'. Its recommended that this bit is 1 when SDCLKDL_DC is being written. Ensure this is '0' when not updating code. Note: Turn-off card clock using CLK_CTRL_R.SD_CLK_EN before programing this field. Values: ■ 0x1 (BYPASSMODE): output of DelayLine is DelayLine output active ■ 0x0 (DLMODE): DelayLine output is enabled	0x0
3:2	R/W	INPSEL_CNFG	Drives SD/eMMC CLK DelayLine's config input. Value here selects the input source to DelayLine	0x0
1	R/W	BYPASS_EN	Drives SD/eMMC CLK DelayLine's bypassen input Values: ■ 0x1 (BYPASSMODE): DelayLine is bypass mode ■ 0x0 (DLMODE): Delay line active mode	0x0
0	R/W	EXTDLY_EN	Drives SD/eMMC CLK DelayLine's extdlyen input Values: ■ 0x1 (EXTDL_MODE): DelayLine works with extended delay range setting ■ 0x0 (DEF_MODE): Delay line default range setting	0x0

SDCLKDL_DC

SDCLKDL_DC is a SD/eMMC DelayLine Delay code setting Offset Address : DWC_MSHC_PTR_PHY_REGS + 0x1E
Total Reset Value:0x0

Bits	Access	Name	Description	Reset
7			Reserved Field: Yes	
6:0	R/W	CCKDL_DC	Drives SD/eMMC CLK DelayLine's Delay Code input. Value here Selects the number of active stages in the card clock delay line.Note: Turn-off card clock using CLK_CTRL_R.SD_CLK_EN before programing this field.	0x0

SMPLDL_CNFG

SMPLDL_CNFG is a SD/eMMC cclk_rx DelayLine settings Offset Address : DWC_MSHC_PTR_PHY_REGS + 0x20
Total Reset Value:0x0e

Bits	Access	Name	Description	Reset
7:5			Reserved Field: Yes	
4	R/W	INPSEL_OVERRIDE	PHY's Sampling delay line config is controlled by controller using sample_cclk_sel, this signal overrides sample_cclk_sel such that INPSEL_CFG field directly control's PHY's config input. ■ 0x0 : Controller logic drive Sampling delay line config. ■ 0x1 : SMPLDL_CNFG.INPSEL_CNFG drives sampling delay line config.	0x0

Bits	Access	Name	Description	Reset
3:2	R/W	INPSEL_CNFG	Drives CCLK_RX DelayLine's config input. Value here selects the input source to DelayLine	0x3
1	R/W	BYPASS_EN	Drives CCLK_RX DelayLine's bypassen input Values: ■ 0x1 (BYPASSMODE): DelayLine is bypass mode ■ 0x0 (DLMODE): Delay line active mode	0x1
0	R/W	EXTDLY_EN	Drives CCLK_RX DelayLine's extdlyen input Values: ■ 0x1 (EXTDL_MODE): DelayLine works with extended delay range setting ■ 0x0 (DEF_MODE): Delay line default range setting	0x0

ATDL_CNFG

ATDL_CNFG is a SD/eMMC drift_cclk_rx DelayLine configuration settings Offset Address :
DWC_MSHC_PTR_PHY_REGS + 0x21 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
7:4			Reserved Field: Yes	
3:2	R/W	INPSEL_CNFG	Drives drift_cclk_rx DelayLine's config input. Value here selects the input source to DelayLine	0x0
1	R/W	BYPASS_EN	Drives drift_cclk_rx DelayLine's bypassen input Values: ■ 0x1 (BYPASSMODE): DelayLine is bypass mode ■ 0x0 (DLMODE): Delay line active mode	0x0
0	R/W	EXTDLY_EN	Drives drift_cclk_rx DelayLine's extdlyen input Values: ■ 0x1 (EXTDL_MODE): DelayLine works with extended delay range setting ■ 0x0 (DEF_MODE): Delay line default range setting	0x0

DLL_CTRL

DLL_CTRL is a SD/eMMC PHY DLL control setting Offset Address : DWC_MSHC_PTR_PHY_REGS + 0x24 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
7:3	R	RSVD_3_7	These bits of the register are reserved.Reserved Field: Yes	0x0
2	R/W	SLV_SWDC_UPDATE	Corresponding output drives PHY's DLL Slave's dc update input. This is used to turn-off Slave Delay line's output when changing its delay code using DLL_OFFST register Values: ■ 0x1 (DL_OUT_OFF): Update in progress ■ 0x0 (DL_OUT_ON): Update completed	0x0

Bits	Access	Name	Description	Reset
1	R/W	OFFST_EN	Enables offset mode of PHY when DLL is enabled. when DLL is disabled this allows direct control of delay generated by DLL's Slave Values: ■ 0x1 (OFFSTEN): Offset value is valid ■ 0x0 (OFFSTDIS): offset value is invalid	0x0
0	R/W	DLL_EN	Enable's DLL when '1' Values: ■ 0x1 (DLENABLE): PHY DLL is enabled ■ 0x0 (DLLDISABLE): PHY DLL is disabled	0x0

DLL_CNFG1

DLL_CNFG1 is a DLL Config register 1 Offset Address : DWC_MSHC_PTR_PHY_REGS + 0x25 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
7:6			Reserved Field: Yes	
5:4	R/W	SLVDLY	Sets the value of DLL slave's update delay input slv_update_dly	0x0
3			Reserved Field: Yes	
2:0	R/W	WAITCYCLE	Sets the value of DLL's wait cycle input	0x0

DLL_CNFG2

DLL_CNFG2 is a DLL Config register 2

Offset Address : DWC_MSHC_PTR_PHY_REGS + 0x26 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
7			Reserved Field: Yes	
6:0	R/W	JUMPSTEP	Sets the value of DLL's jump step input	0x0

DLLDL_CNFG

DLLDL_CNFG is a DLL Config register 2 Offset Address : DWC_MSHC_PTR_PHY_REGS + 0x28 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
7	R/W	SLV_BYPASS	Bypass enable control for Slave DL	0x0
6:5	R/W	SLV_INPSEL	Clock source select for Slave DL	0x0
4	R/W	SLV_EXTDLYEN	Enable Extended delay mode for Slave	0x0
3	R/W	MST_BYPASS	Bypass enable control for Master DL	0x0
2:1	R/W	MST_INPSEL	Clock source select for Master DL	0x0
0	R/W	MST_EXTDLYEN	Enable Extended delay mode for master	0x0

DLL_OFFST

DLL_OFFST is a DLL Offset setting register Offset Address : DWC_MSHC_PTR_PHY_REGS + 0x29 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
7			Reserved Field: Yes	
6:0	R/W	OFFST	Sets the value of DLL's offset input	0x0

DLLMST_TSTDC

DLLMST_TSTDC is a DLL Master test code setting register Offset Address : DWC_MSHC_PTR_PHY_REGS + 0x2A Total Reset Value:0x0

Bits	Access	Name	Description	Reset
7			Reserved Field: Yes	
6:0	R/W	MSTTST_DC	Sets the value of DLL's Master test code input when DLL is disabled.	0x0

DLLLBT_CNFG

DLLLBT_CNFG is a DLL LBT setting register Offset Address : DWC_MSHC_PTR_PHY_REGS + 0x2C Total Reset Value:0x0

Bits	Access	Name	Description	Reset
15:0	R/W	LBT_LOADVAL	Sets the value of DLL's olbt_loadval input. Controls the lbt timer's timeout value at which DLL runs a revalidation cycle.	0x0

DLL_STATUS

DLL_STATUS is a DLL Status register Offset Address : DWC_MSHC_PTR_PHY_REGS + 0x2E Total Reset Value:0x0

Bits	Access	Name	Description	Reset
7:2			Reserved Field: Yes	
1	R	ERROR_STS	Captures the value of DLL's lock error status information. Value is valid only when LOCK_STS is set. ■ IF LOCK_STS =1 and ERR_STS = 0 then DLL is locked and no errors are generated ■ IF LOCK_STS =1 and ERR_STS = 1 then DLL is locked to default and has errors. Transactions at this phase can fail Values: ■ 0x1 (DLL_ERROR): DLL is locked and ready ■ 0x0 (DLL_LOCK_OKAY): DLL has not locked	0x0
0	R	LOCK_STS	Captures the value of DLL's lock status information Values: ■ 0x1 (DLL_IS_LOCKED): DLL is locked and ready ■ 0x0 (DLL_NOT_LOCKED): DLL has not locked	0x0

DLLDBG_MLKDC

DLLDBG_MLKDC is a DLL Master lock code debug register Offset Address : DWC_MSHC_PTR_PHY_REGS + 0x30 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
7			Reserved Field: Yes	
6:0	R	MSTLKDC	Captures the value Delay Code to which DLL's Master has locked	0x0

DLLDBG_SLKDC

DLLDBG_SLKDC is a DLL Master Slave code debug register Offset Address : DWC_MSHC_PTR_PHY_REGS + 0x32
Total Reset Value:0x0

Bits	Access	Name	Description	Reset
7			Reserved Field: Yes	
6:0	R	SLVLKDC	Captures the value Delay Code to which DLL's Slave has locked	0x0

MSHC_VER_ID_R

MSHC_VER_ID_R is a MSHC version Offset Address : P_VENDOR_SPECIFIC_AREA[11:0] Total Reset
Value:0x30313930

Bits	Access	Name	Description	Reset
31:0	R	MSHC_VER_ID	Current release number This field indicates the Synopsys DesignWare Cores DWC_mshc/DWC_mshc_lite current release number that is read by an application. An application reading this register in conjunction with the MSHC_VER_TYPE_R register, gathers details of the current release.	0x30313930

MSHC_VER_TYPE_R

MSHC_VER_TYPE_R is a MSHC version type Offset Address : P_VENDOR_SPECIFIC_AREA[11:0] + 0x4 Total
Reset Value: 0x67612a2a

Bits	Access	Name	Description	Reset
31:0	R	MSHC_VER_TYPE	Current release type This field indicates the Synopsys DesignWare Cores DWC_mshc/DWC_mshc_lite current release type that is read by an application. An application reading this register in conjunction with the MSHC_VER_ID_R register, gathers details of the current release.	0x67612a2a

MSHC_CTRL_R

MSHC_CTRL_R is a MSHC Control register Offset Address : P_VENDOR_SPECIFIC_AREA[11:0] + 0x8 Total Reset
Value:0x01

Bits	Access	Name	Description	Reset

Bits	Access	Name	Description	Reset
7	R/W	NEDGE_SMPL_EN	Samples CMD/DATA with respect to negative edge of cclk_rx for low speed SDR only support. Note: Negedge sampling shall be enable only if the total output delay is greater that ($t_{\text{clockperiod}} ? t_{\text{setup}} ? t_{\text{diff}}$) and limited to ($1.5 * t_{\text{clockperiod}} ? t_{\text{setup}} ? t_{\text{diff}}$). Values: ■ 0x0 (PEDGE_SMPL): Samples CMD/DATA with respect to positive edge of cclk_rx ■ 0x1 (NEDGE_SMPL): Samples CMD/DATA with respect to negative edge of cclk_rx	0x0
6	R/W	PEDGE_DRV_EN	Launches CMD/DATA with respect to positive edge of cclk_tx for low speed SDR only support. Values: ■ 0x0 (NEDGE_DRV): Launches CMD/DATA with respect to negative edge of cclk_tx ■ 0x1 (PEDGE_DRV): Launches CMD/DATA with respect to positive edge of cclk_tx	0x0
5	R	SLV_ERR_RESP_NONEXIS_REG	When 0 this bit gives slave error response for non existent register access. Currently this is a read-only bit with default value 0.	0x0
4	R/W	SW_CG_DIS	Internal clock gating disable control This bit must be used to disable IP's internal clock gating when required. when disabled clocks are not gated. Clocks to the core (except hclk) must be stopped when programming this bit. Values: ■ 0x0 (ENABLE): Internal clock gates are active and clock gating is controlled internally ■ 0x1 (DISABLE): Internal clock gating is disabled, clocks are not gated internally	0x0
3:2	R	RSVD1	These bits (RSVD1) of the MSHC_CTRL_R register are reserved. They always return 0.	0x0
1	R/W	NEGEDGE_DATAOUT_EN	Negedge Data Out This bit enables the data to be sent out on negedge of cclk_tx. This is valid only for speed mode HS400. Values: ■ 0x0 (DISABLED): Negedge data sent out disabled ■ 0x1 (ENABLED): Negedge data sent out enabled	0x0

Bits	Access	Name	Description	Reset
0	R/W	CMD_CONFLICT_CHECK	Command conflict check This bit enables command conflict check. Note: DWC_mshc controller monitors the CMD line whenever a command is issued and checks whether the value driven on sd_cmd_out matches the value on sd_cmd_in at next subsequent edge of cclk_tx to determine command conflict error. This bit is cleared only if the feed back delay (including IO Pad delay) is more than (t_card_clk_period - t_setup), where t_setup is the setup time of a flop in DWC_mshc. The I/O pad delay is consistent across CMD and DATA lines, and it is within the value:(2*t_card_clk_period - t_setup) Values: ■ 0x0 (DISABLE_CMD_CONFLICT_CHK): Disable command conflict check ■ 0x1 (CMD_CONFLICT_CHK_LAT1): Check for command conflict after 1 card clock cycle	0x1

MBIU_CTRL_R

MBIU_CTRL_R is an MBIU Control register Offset Address : P_VENDOR_SPECIFIC_AREA[11:0] + 0x10 Total Reset Value:0x0f

Bits	Access	Name	Description	Reset
7:4	R	RSVD	Reserved field	0x0
3	R/W	BURST_INCR16_EN	INCR16 Burst Controls generation of INCR16 transfers on Master interface. Values: ■ 0x0 (FALSE): AHB INCR16 burst type is not generated on Master I/F ■ 0x1 (TRUE): AHB INCR16 burst type can be generated on Master I/F	0x1
2	R/W	BURST_INCR8_EN	INCR8 Burst Controls generation of INCR8 transfers on Master interface. Values: ■ 0x0 (FALSE): AHB INCR8 burst type is not generated on Master I/F ■ 0x1 (TRUE): AHB INCR8 burst type can be generated on Master I/F	0x1

Bits	Access	Name	Description	Reset
1	R/W	BURST_INCR4_EN	INCR4 Burst Controls generation of INCR4 transfers on Master interface. Values: ■ 0x0 (FALSE): AHB INCR4 burst type is not generated on Master I/F ■ 0x1 (TRUE): AHB INCR4 burst type can be generated on Master I/F	0x1
0	R/W	UNDEFL_INCR_EN	Undefined INCR Burst Controls generation of undefined length INCR transfer on Master interface. Values: ■ 0x0 (FALSE): Undefined INCR type burst is the least preferred burst on AHB Master I/F ■ 0x1 (TRUE): Undefined INCR type burst is the most preferred burst on AHB Master I/F	0x1

EMMC_CTRL_R

EMMC_CTRL_R is an eMMC Control register Offset Address : P_VENDOR_SPECIFIC_AREA[11:0] + 0x2C Total
Reset Value:0x000C

Bits	Access	Name	Description	Reset
15:11	R	RSVD	These bits (RSVD) of the EMMC_CTRL_R register are reserved. They always return 0.	0x0
10	R/W	CQE_PREFETCH_DISABLE	Enable or Disable CQE's PREFETCH feature This field allows Software to disable CQE's data prefetch feature when set to 1. Values: ■ 0x0 (PREFETCH_ENABLE): CQE can Prefetch data for successive WRITE transfers and pipeline successive READ transfers ■ 0x1 (PREFETCH_DISABLE): Prefetch for WRITE and Pipeline for READ are disabled	0x0
9	R/W	CQE_ALGO_SEL	Scheduler algorithm selected for execution This bit selects the Algorithm used for selecting one of the many ready tasks for execution. Values: ■ 0x0 (PRI_REORDER_PLUS_FCFS): Priority based reordering with FCFS to resolve equal priority tasks ■ 0x1 (FCFS_ONLY): First come First serve, in the order of DBR rings	0x0

Bits	Access	Name	Description	Reset
8	R/W	ENH_STROBE_ENABLE	Enhanced Strobe Enable This bit instructs DWC_mshc to sample the CMD line using data strobe for HS400 mode. Values: ■ 0x1 (ENH_STB_FOR_CMD): CMD line is sampled using data strobe for HS400 mode ■ 0x0 (NO_STB_FOR_CMD): CMD line is sampled using cclk_rx for HS400 mode	0x0
7:4			Reserved Field: Yes	
3	R/W	EMMC_RST_N_OE	Output Enable control for EMMC Device Reset signal PAD control. This field driven sd_rst_n_oe output of DWC_mshc Values: ■ 0x1 (ENABLE): sd_rst_n_oe is 1 ■ 0x0 (DISABLE): sd_rst_n_oe is 0	0x1
2	R/W	EMMC_RST_N	EMMC Device Reset signal control. This register field controls the sd_rst_n output of DWC_mshc Values: ■ 0x1 (RST_DEASSERT): Reset to eMMC device is deasserted ■ 0x0 (RST_ASSERT): Reset to eMMC device asserted (active low)	0x1
1	R/W	DISABLE_DATA_CRC_CHK	Disable Data CRC Check This bit controls masking of CRC16 error for Card Write in eMMC mode. This is useful in bus testing (CMD19) for an eMMC device. In bus testing, an eMMC card does not send CRC status for a block, which may generate CRC error. This CRC error can be masked using this bit during bus testing. Values: ■ 0x1 (DISABLE): DATA CRC check is disabled ■ 0x0 (ENABLE): DATA CRC check is enabled	0x0
0	R/W	CARD_IS_EMMC	eMMC Card present This bit indicates the type of card connected. An application program this bit based on the card connected to MSHC. Values: ■ 0x1 (EMMC_CARD): Card connected to MSHC is an eMMC card ■ 0x0 (NON_EMMC_CARD): Card connected to MSHC is a non-eMMC card	0x0

BOOT_CTRL_R

BOOT_CTRL_R is an eMMC Boot Control register Offset Address : P_VENDOR_SPECIFIC_AREA[11:0] + 0x2E Total Reset Value:0x0

Bits	Access	Name	Description	Reset
15:12	R/W	BOOT_TOUT_CNT	Boot Ack Timeout Counter Value. This value determines the interval by which boot ack timeout (50 ms) is detected when boot ack is expected during boot operation. ■ 0xF : Reserved ■ 0xE : TMCLK x 2²⁷ ■ .. - ■ 0x1 : TMCLK x 2¹⁴ ■ 0x0 : TMCLK x 2¹³	0x0
11:9	R	RSVD_11_9	These bits (RSVD_11_9) of the BOOT_CTRL_R register are reserved. They always return 0.	0x0
8	R/W	BOOT_ACK_ENABLE	Boot Acknowledge Enable When this bit set, DWC_mshc checks for boot acknowledge start pattern of 0-1-0 during boot operation. This bit is applicable for both mandatory and alternate boot mode. Values: ■ 0x1 (TRUE): Boot Ack enable ■ 0x0 (FALSE): Boot Ack disable	0x0
7	W	VALIDATE_BOOT	Validate Mandatory Boot Enable bit This bit is used to validate the MAN_BOOT_EN bit. Values: ■ 0x1 (TRUE): Validate Mandatory boot enable bit ■ 0x0 (FALSE): Ignore Mandatory boot Enable bit	0x0
6:1	R	RSVD_6_1	These bits (RSVD_6_1) of the BOOT_CTRL_R register are reserved. They always return 0.	0x0
0	R/W	MAN_BOOT_EN	Mandatory Boot Enable This bit is used to initiate the mandatory boot operation. The application sets this bit along with VALIDATE_BOOT bit. Writing 0 is ignored. The DWC_mshc clears this bit after the boot transfer is completed or terminated. Values: ■ 0x1 (MAN_BOOT_EN): Mandatory boot enable ■ 0x0 (MAN_BOOT_DIS): Mandatory boot disable	0x0

GP_IN_R

GP_IN_R is a General Purpose Input register Offset Address : P_VENDOR_SPECIFIC_AREA[11:0] + 0x30 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:y	R	RSVD	These bits of the GP_IN_R register are reserved. They always return 0.	0x0
x:0	R	GP_IN	It reflects the value of gp_in ports.	0x0

GP_OUT_R

GP_OUT_R is a General Purpose Output register Offset Address : P_VENDOR_SPECIFIC_AREA[11:0] + 0x34 Total Reset Value:0x0

Bits	Access	Name	Description	Reset
31:y	R	RSVD	These bits of the GP_OUT_R register are reserved. They always return 0.	0x0
x:0	R/W	GP_OUT	The value of this register is reflected on gp_out ports.	0x0

AT_CTRL_R

AT_CTRL_R is a Tuning and Auto-tuning control register Offset Address : P_VENDOR_SPECIFIC_AREA[11:0] + 0x40 Total Reset Value:0x1f000001

Bits	Access	Name	Description	Reset
x:24	R/W	SWIN_TH_VAL	Sampling window threshold value setting The maximum value that can be set here depends on the length of delayline used for tuning. A delayLine with 128 taps can use values from 0x0 to 0x7F. This field is valid only when SWIN_TH_EN is '1'. Should be programmed only when SAMPLE_CLK_SEL is '0' ■ 0x0 : Threshold values is 0x1, windows of length 1 tap and above can be selected as sampling window. ■ 0x1 : Threshold values is 0x2, windows of length 2 taps and above can be selected as sampling window. ■ 0x2 : Threshold values is 0x1, windows of length 3 taps and above can be selected as sampling window. ■ ■ 0x7F : Threshold values is 0x1, windows of length 127 taps and above can be selected as sampling window.	0x1f
23:21			Reserved Field: Yes	
20:19	R/W	POST_CHANGE_DLY	Time taken for phase switching and stable clock output. Specifies the maximum time (in terms of cclk cycles) that the delay line can take to switch its output phase after a change in tuning_cclk_sel or autotuning_cclk_sel. Values: ■ 0x0 (LATENCY_LT_1): Less than 1-cycle latency ■ 0x1 (LATENCY_LT_2): Less than 2-cycle latency ■ 0x2 (LATENCY_LT_3): Less than 3-cycle latency ■ 0x3 (LATENCY_LT_4): Less than 4-cycle latency	0x0
18:17	R/W	PRE_CHANGE_DLY	Maximum Latency specification between cclk_tx and cclk_rx. Values: ■ 0x0 (LATENCY_LT_1): Less than 1-cycle latency ■ 0x1 (LATENCY_LT_2): Less than 2-cycle latency ■ 0x2 (LATENCY_LT_3): Less than 3-cycle latency ■ 0x3 (LATENCY_LT_4): Less than 4-cycle latency	0x0

Bits	Access	Name	Description	Reset
16	R/W	TUNE_CLK_STOP_EN	Clock stopping control for Tuning and auto-tuning circuit. When enabled, clock gate control output of DWC_mshc (clk2card_on) is pulled low before changing phase select codes on tuning_cclk_sel and autotuning_cclk_sel. This effectively stops the Device/Card clock, cclk_rx and also drift_cclk_rx. Changing phase code when clocks are stopped ensures glitch free phase switching. Set this bit to 0 if the PHY or delayline can guarantee glitch free switching. Values: ■ 0x1 (ENABLE_CLK_STOPPING): Clocks stopped during phase code change ■ 0x0 (DISABLE_CLK_STOPPING): Clocks not stopped. PHY ensures glitch free phase switching.	0x0
15:12	R	RSVD3	These bits (RSVD3) of the AT_CTRL_R register are reserved. They always return 0.	0x0
11:8	R/W	WIN_EDGE_SEL	This field sets the phase for Left and Right edges for drift monitoring. [Left edge offset + Right edge offset] must not be less than total taps of delayLine. ■ 0x0: User selection disabled. Tuning calculated edges are used. ■ 0x1: Right edge Phase is center + 2 stages, Left edge Phase is center - 2 stages. ■ 0x2: Right edge Phase is center + 3 stages, Left edge Phase is center - 3 stages. ■ ... ■ 0xF: Right edge Phase is center + 16 stages, Left edge Phase is center - 16 stages.	0x0
7:5	R	RSVD2	These bits (RSVD2) of the AT_CTRL_R register are reserved. They always return 0.	0x0
4	R/W	SW_TUNE_EN	This fields enables software-managed tuning flow. Values: ■ 0x1 (SW_TUNING_ENABLE): Software-managed tuning enabled. AT_STAT_R.CENTER_PH_CODE Field is now writable. ■ 0x0 (SW_TUNING_DISABLE): Software-managed tuning disabled.	0x0
3	R/W	RPT_TUNE_ERR	Framing errors are not generated when executing tuning. This debug bit allows users to report these errors. Values: ■ 0x1 (DEBUG_ERRORS): Debug mode for reporting framing errors ■ 0x0 (ERRORS_DISABLED): Default mode where as per SD-HCI no errors are reported.	0x0

Bits	Access	Name	Description	Reset
2	R/W	SWIN_TH_EN	Sampling window Threshold enable Selects the tuning mode Field should be programmed only when SAMPLE_CLK_SEL is '0' Values: ■ 0x1 (THRESHOLD_MODE): Tuning engine selects the first complete sampling window that meets the threshold set by SWIN_TH_VAL field ■ 0x0 (LARGEST_WIN_MODE): Tuning engine sweeps all taps and settles at the largest window	0x0
1	R/W	CI_SEL	Selects the interval when the corrected center phase select code can be driven on tuning_cclk_sel output. Values: ■ 0x0 (WHEN_IN_BLK_GAP): Driven in block gap interval ■ 0x1 (WHEN_IN_IDLE): Driven at the end of the transfer	0x0
0	R/W	AT_EN	Setting this bit enables Auto tuning engine. This bit is enabled by default when core is configured with mode3 retuning support. Clear this bit to 0 when core is configured to have Mode3 re-tuning but SW wishes to disable mode3 re-tuning. This field should be programmed only when CLK_CTRL_R.SD_CLK_EN is 0. Values: ■ 0x1 (AT_ENABLE): AutoTuning is enabled ■ 0x0 (AT_DISABLE): AutoTuning is disabled	0x1

AT_STAT_R

AT_STAT_R is a Tuning and Auto-tuning status register Offset Address : P_VENDOR_SPECIFIC_AREA[11:0] + 0x44
Total Reset Value:0x00000006

Bits	Access	Name	Description	Reset
31:24	R	RSDV1	These bits of the AT_STAT_R register are reserved. They always return 0.	0x0
23:16	R	L_EDGE_PH_CODE	Left Edge Phase code. Reading this field returns the phase code value used by Auto-tuning engine to sample data on Left edge of sampling window.	0x0
15:8	R	R_EDGE_PH_CODE	Right Edge Phase code. Reading this field returns the phase code value used by Auto-tuning engine to sample data on Right edge of sampling window.	0x0
7:0	R/W	CENTER_PH_CODE	Centered Phase code. Reading this field returns the current value on tuning_cclk_sel output. Setting AT_CTRL_R.SW_TUNE_EN enables software to write to this field and its contents are reflected on tuning_cclk_sel.	0x6

12.5 GPIO

12.5.1 Overview

The chip supports 3 GPIO groups, namely GPIO group0–GPIO group2. Each GPIO group provides 32 programmable input/output pins (GPIO group2 has only 8 input/output pins).

Each pin can be configured as input or output and these pins are used to generate input signals or output signals for special purposes. As input, each GPIO pin can act as an interrupt source in GPIO group0 and GPIO group1; as output, each GPIO pin can be set to 0 or 1 separately.

The GPIO can generate maskable interrupts based on the level or transition value. The general purpose input output interrupt (GPIOINTR) signal provides an indicator to the interrupt controller, indicating that an interrupt occurs.

The chip provides 72 GPIOs:

- GPIO group0: GPIO0-31, that can be programmed to accept external signals as interrupt sources on any of the bits of the signal.
- GPIO group1: GPIO32-63, that can be programmed to accept external signals as interrupt sources on any of the bits of the signal.
- GPIO group2: GPIO64-71

12.5.2 Feature

The GPIO has the following features:

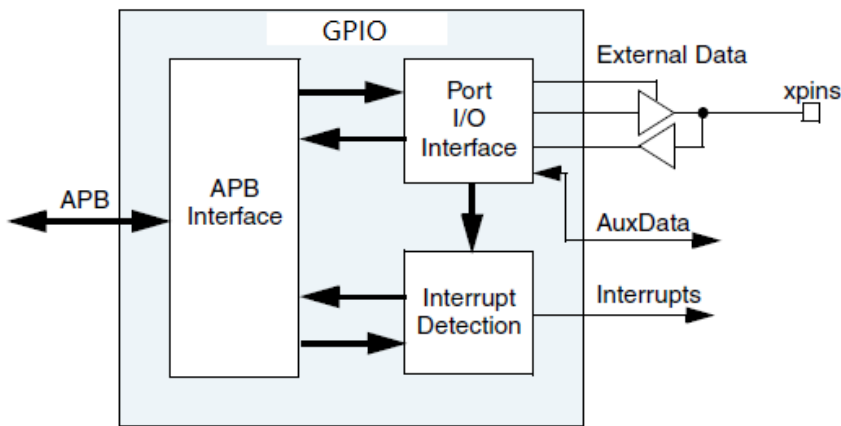
- Separate data registers and data direction registers for each signal
- Configurable hardware and software control for each signal, or for each bit of each signal Separate auxiliary data input, data output, and data control for each I/O in Hardware Control mode. As output, each GPIO can be set to 0 or 1 separately.
- Independently controllable signal bits
- Configurable interrupt mode for GPIO group0 and GPIO group1. As input, each GPIO pin can act as an interrupt source. IO both edge interrupts.
- Configurable debounce logic with an external slow clock to debounce interrupts
- Option to generate single or multiple interrupts
- GPIO Component Type register
- GPIO Component Version register
- Configurable reset values on output signals
- Configurable synchronization of interrupt signals
- 32-bit APB width

12.5.3 Block Diagram

Figure 12-5-1 shows the following functional groupings of the main interfaces to the GPIO block:

- APB interface to or from APB bridge
- External data Interface to or from I/O pads
- Auxiliary hardware data interface to or from auxiliary data sink or source
- Interrupt interface to or from interrupt controller

[Figure 12-5-1](#) GPIO Block Diagram



12.5.4 Function Description

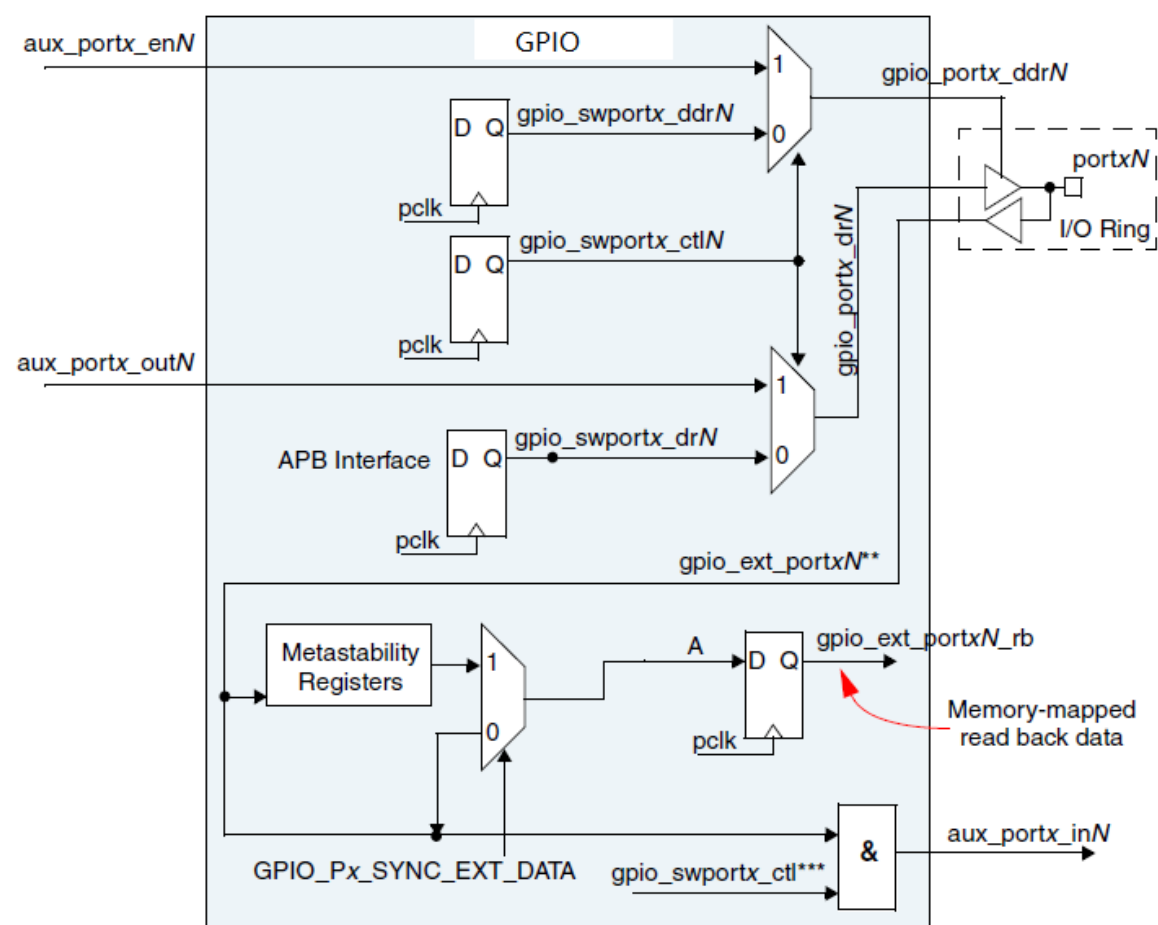
12.5.4.1 Data and Control Flow

The GPIO controls the output data and direction of external I/O pads. It also can read back the data on external pads using memory-mapped registers.

There are two methods for generating the default source of the input data, the output data, and the control of each signal—through software or hardware control. Software control occurs over the APB bus interface; hardware control is through the auxiliary hardware control interface.

The data and control flow for a signal are shown in Figure 12-5-2.

Figure 12-5-2 Control RTL Block Diagram



12.5.4.1.1 Software Control Mode

When a signal is configured for software control, the data and direction control for the signal are sourced from the data register (gpio_swporta_dr) and direction control register (gpio_swporta_ddr).

Under software control, the direction of the external I/O pad is controlled by a write to the Porta data direction register (gpio_swporta_ddr). The data written to this memory-mapped register gets mapped onto an output signal, gpio_porta_ddr, of the GPIO peripheral. This output signal controls the direction of an external I/O pad.

The data written to the Portx data register (gpio_swporta_dr) drives the output buffer of the I/O pad.

External data are input on the external data signal, gpio_ext_porta. Depending on whether gpio_ext_porta is configured as an input or output, the following occurs:

- Input – Reads the values on the signal
- Output – Reads the data register for Porta

The gpio_ext_porta register is read-only, meaning that it cannot be written from the APB software interface.

12.5.4.1.2 Hardware Control Mode

If a signal is configured for hardware control, it has external, auxiliary hardware signals controlling the data and the direction of Ports A through D. In this mode, the auxiliary data input signal (aux_porta_out) and direction control signal (aux_portx_en) are selected.

The data direction of the external I/O pad, gpio_porta_ddr, is controlled through the auxiliary signal direction control signal, aux_porta_en.

For lines that are set to Output, the gpio_porta_dr and gpio_porta_ddr output signals drive the data and direction control onto the bidirectional pad that exists within the I/O ring of the SoC device. Figure 12-6-2 shows how the GPIO peripheral controls the data and direction signals of an I/O PAD and data generation for the auxiliary source.

The gpio_swporta_ctl signal masks the value on the gpio_ext_porta external signal in order to generate aux_porta_in.

The net result is that when hardware mode is selected, the value on aux_porta_in output signal is equal to the value on the gpio_ext_porta input signal. When software mode is selected, the aux_portx_in output signal is always driven low.

Setting bit 0 of gpio_swporta_ctl to 1 selects hardware mode for the entire signal if the parameter

GPIO_PORTX_SINGLE_CTL is 1. If GPIO_PORTX_SINGLE_CTL is 0, setting bit n of gpio_swporta_ctl to 1 selects hardware mode for bit n of Porta. Setting bit 0 of gpio_swporta_ctl to 0 selects software mode for the entire signal if GPIO_PORTA_SINGLE_CTL is 1, while setting bit n of gpio_swporta_ctl to 0 selects software mode for bit n of Portx if GPIO_PORTA_SINGLE_CTL is 0.

12.5.4.1.3 Reading External Signals

The data on the external gpio signal is read by an APB read of the memory-mapped register, gpio_ext_porta. An APB read to the gpio_ext_porta register provides either the data on the gpio_ext_porta control lines or the contents of the gpio_swporta_dr, depending on the value of gpio_swporta_ddr.

Figure 12-6-2 shows how the hardware/software option is multiplexed, where the control lines for the multiplexing come from a memory-mapped register. It also shows the synchronization registers and the individual bit control of each data and data direction bit.

Figure 12-6-3 shows a timing diagram of a read to the gpio_ext_porta memory map registers when the direction is set to Input and the metastability registers are included.

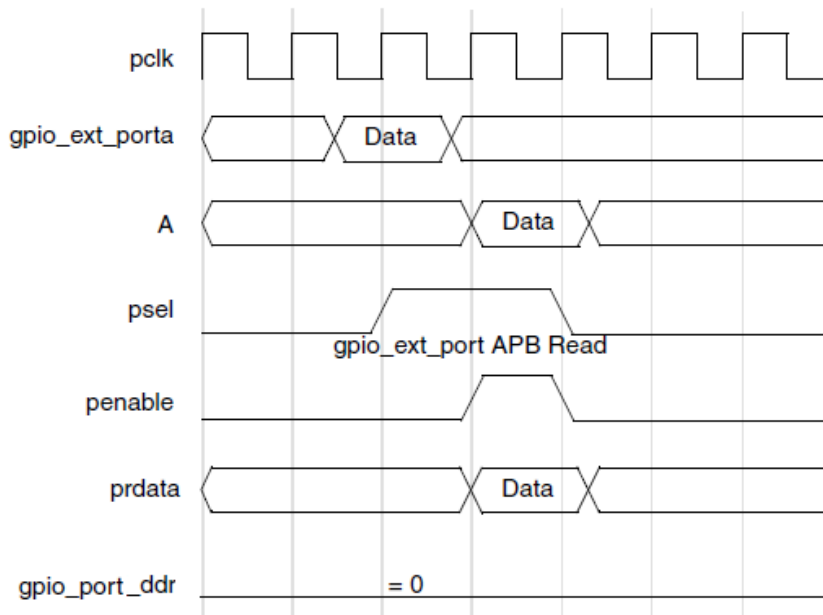


Figure 12-5-3 Read Back of External gpio_ext_porta Data Timing

12.5.4.1.4 Synchronization of Data to System Clock

Synchronization of gpio_ext_porta to pclk prior to an APB read is enabled if the corresponding signal configuration parameter GPIO_PA_SYNC_EXT_DATA is set; this is shown in Figure 12-6-2.

12.5.4.2 Interrupts

GPIO0-63 can be programmed to accept external signals as interrupt sources on any of the bits of the signal. The type of interrupt is programmable with one of the following settings:

- Active-high and level
- Active-low and level
- Rising edge
- Falling edge

The interrupts can be masked by programming the gpio_intmask register. The interrupt status can be read before masking (called raw status) and after masking.

The interrupts are also combined into a single interrupt output signal, which has the same polarity as the individual interrupts. Either individual interrupts (gpio_intr or gpio_intr_n) or a single combined interrupt (gpio_intr_flag or gpio_intr_flag_n) can be generated. In order to mask the combined interrupt, all individual interrupts have to be masked. The single combined interrupt does not have its own mask bit.

Whenever GPIO is configured for interrupts, the data direction must be set to Input and the mode must be set to Software for interrupts to be latched. If the data direction register is reprogrammed to Output or the mode register is programmed to enable Hardware mode, then any pending interrupts are not lost. However, no new interrupts are generated.

Figure 12-6-4 illustrates how the interrupts are generated and how the data flows. The signal names in the diagram correspond to either I/O signals or memory-mapped registers.

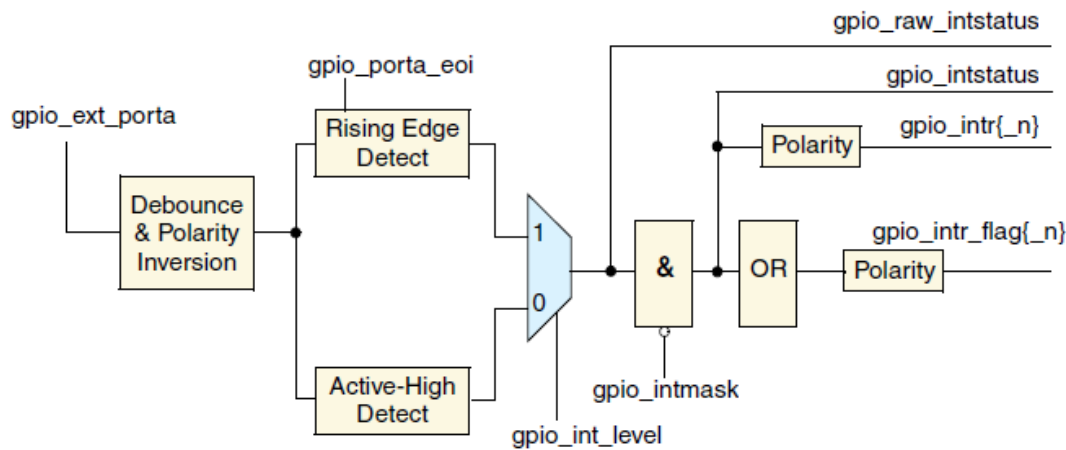


Figure 12-5-4 Interrupt RTL Block Diagram

Two interrupt request connection schemes are supported, and one scheme is chosen during configuration. The simplest connection scheme is where the combined interrupt `gpio_intr_flag` is generated by ORing together the bits of the `gpio_intr` bus. When only the combined interrupt request is used, then the `gpio_status` register must be read in the interrupt service routine (ISR) to find the source of the interrupt. When the individual interrupts lines are connected directly to the interrupt controller, then the `gpio_status` register does not have to be read by the ISR.

For edge-detected interrupts, the ISR can clear the interrupt by writing a 1 to the `gpio_porta_eoi` register for the corresponding bit to disable the interrupt. This write also clears the interrupt status and raw status registers. It is recommended that the interrupt source be cleared prior to writing to the `gpio_porta_eoi` register. Writing to the `gpio_porta_eoi` register has no effect on level-sensitive interrupts.

If level-sensitive interrupts cause the processor to interrupt, then the ISR can poll the `gpio_rawint` status register until the interrupt source disappears, or it can write to the `gpio_intmask` register to mask the interrupt before exiting the ISR. If the ISR exits without masking or disabling the interrupt prior to exiting, then the level-sensitive interrupt repeatedly requests an interrupt until the interrupt is cleared at the source.

If the `gpio_intr_flag` connection scheme is used and the interrupt service routine reads the `gpio_intr_status` register to find multiple pending interrupt requests, then it is up to the processor to prioritize these pending interrupt requests. There are no restrictions on the number of edge-detected interrupts that can be cleared simultaneously by writing multiple 1's to the `gpio_porta_eoi` register.

12.5.4.2.1 Debounce Operation

The external signal can be debounced to remove any spurious glitches that are less than one period of the external debouncing clock.

Figure 12-5-5 shows an RTL diagram of the debounce circuitry. The timing diagrams show an active-high input signal on `gpio_ext_portaN`. The polarity of the input signal detection is controlled by the memory-mapped signal, `gpio_int_polarity`. For a falling-edge- or active-low-sensitive input, the input is then inverted and the same debounce logic is used as for rising-edge or active-high level-sensitive interrupts.

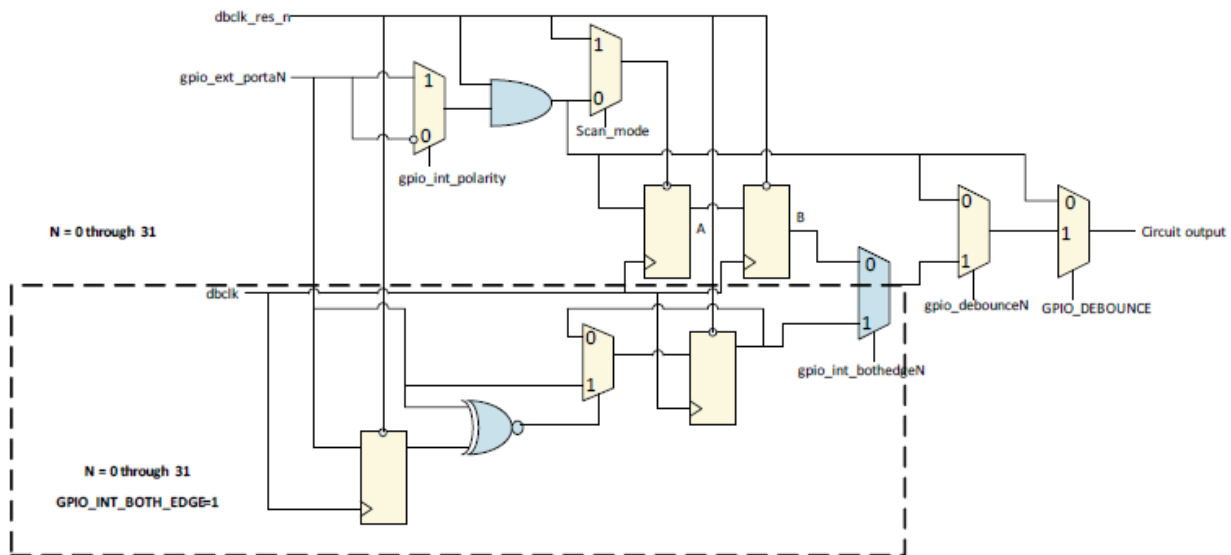


Figure 12-5-5 Debounce RTL Diagram

A timing diagram of the single-edge debounce circuitry is shown in Figure 12-5-6.

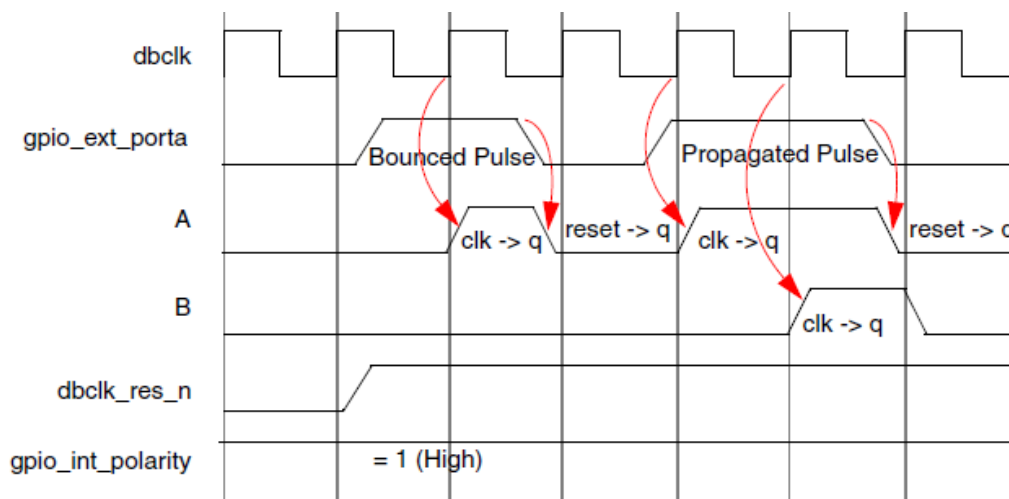


Figure 12-5-6 Debounce Timing With Asynchronous Reset Flip-Flops and GPIO_INT_BOTH_EDGE = 0

When input interrupt signals are debounced using a debounce clock, the signals must be active for a minimum of two cycles of the debounce clock to guarantee that they are registered. Any input pulse widths less than a debounce clock period are bounced. A pulse width between one and two debounce clock widths may or may not propagate, depending on its phase relationship to the debounce clock. If the input pulse spans two rising edges of the debounce clock, it is registered. If it spans only one rising edge, it is not registered.

The timing diagram in Figure 12-6-6 shows both cases: the input signal being bounced, and later, a propagated input signal. If the GPIO supports debounce, then debouncing input signals on Port A can be enabled or disabled under software control.

The dbclk_res_n signal is asynchronously asserted and synchronously de-asserted to the debounce clock, dbclk. The system reset signal, presetn, is asynchronously asserted and synchronously de-asserted to pclk; synchronization must be external to the component. The pclk and dbclk signals are assumed to be asynchronous to each other.

The debounce circuitry works with only asynchronous reset flip-flops.

Figure 12-5-7 shows the debounce timing with parameter GPIO_INT_BOTH_EDGE = 1 and the gpio_int_bothedge register programmed to detect both edges.

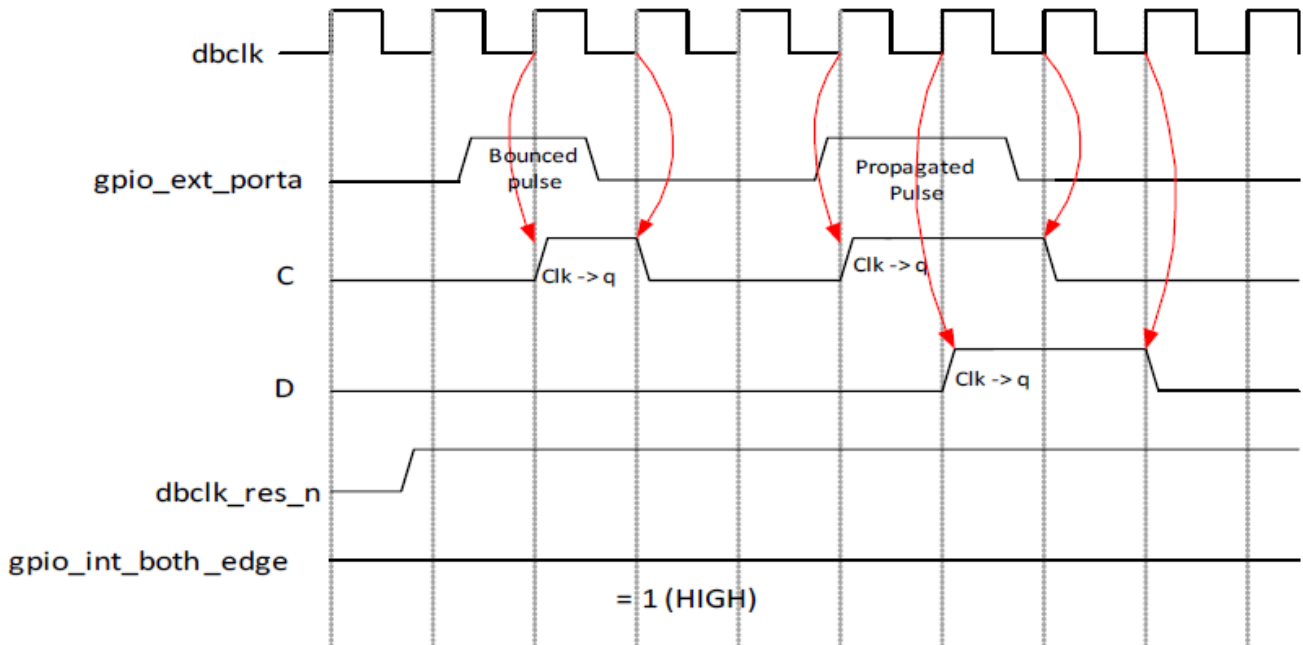


Figure 12-5-7 Debounce Timing With GPIO_INT_BOTH_EDGE = 1

12.5.4.2.2 Synchronization of Interrupt Signals to the System Clock

Interrupt signals can be internally synchronized to a system clock, `pclk_intr`. Synchronization to `pclk_intr` must occur for edge-detect signals. Edge-detected interrupts to the processor are always synchronous to the system bus clock. With level-sensitive interrupts, synchronization is optional and under software control.

The `pclk_intr` signal is needed for systems that may have the GPIO `pclk` bus clock gated off, but the system still wants to detect interrupts. It is assumed that this clock is synchronous to `pclk`. If interrupt detection is required only when `pclk` is running, then `pclk_intr` and `pclk` can be connected to the same clock source. If the system employs a gated `pclk` to the GPIO, `pclk_intr` needs to be running for interrupt detection to occur.

The `gpio_intrclk_en` output signal is asserted when either edge-sensitive interrupts or level-sensitive interrupts requiring synchronization are enabled in the GPIO block. Both cases require a clock for detection. Therefore, this signal can cause the external clock generator block to generate `pclk_intr`.

12.5.4.2.3 Interrupt Edge Detection - Single Edge

Figure 12-5-8 shows an RTL diagram of the synchronization and edge detection of interrupt sources on `gpio_ext_porta` signals, when `GPIO_INT_BOTH_EDGE = 0`.

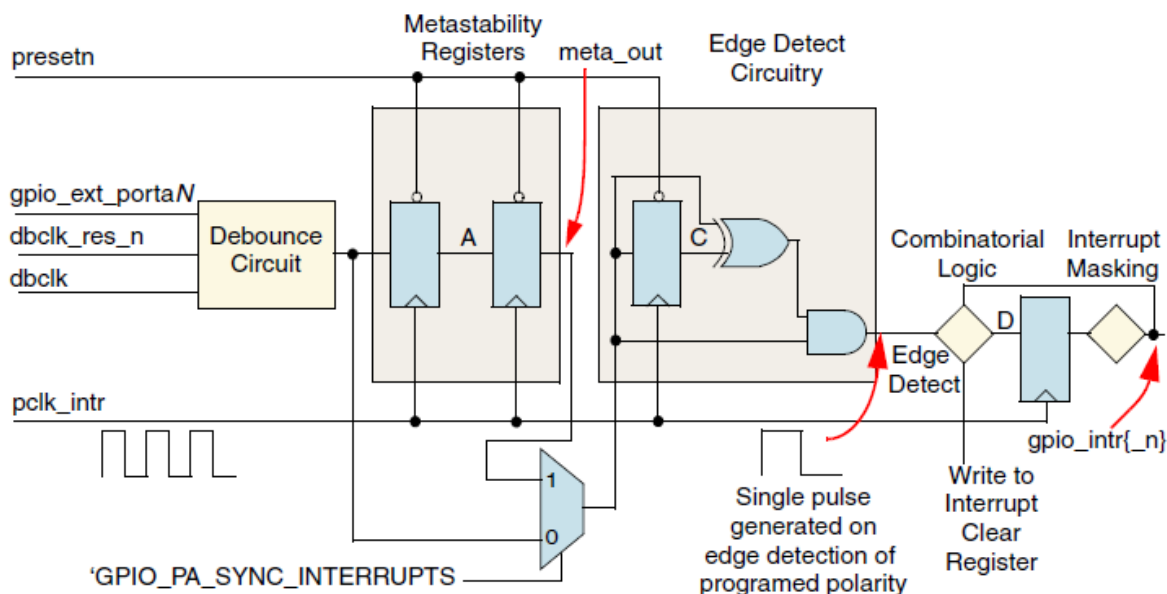


Figure 12-5-8 Synchronization and Edge Detect Interrupt Generation When GPIO_INT_BOTH_EDGE = 0

The MUX allows inclusion or exclusion of the metastability registers at configuration depending on the value of GPIO_PA_SYNC_INTERRUPTS. If this parameter is a 1, the registers are included.

Figure 12-5-9 shows a timing diagram in which an interrupt is generated on the rising edge of an input on GPIO; this is where the debounce logic is disabled and Metastability registers are included. It also shows how an interrupt is cleared by a write to the interrupt clear register.

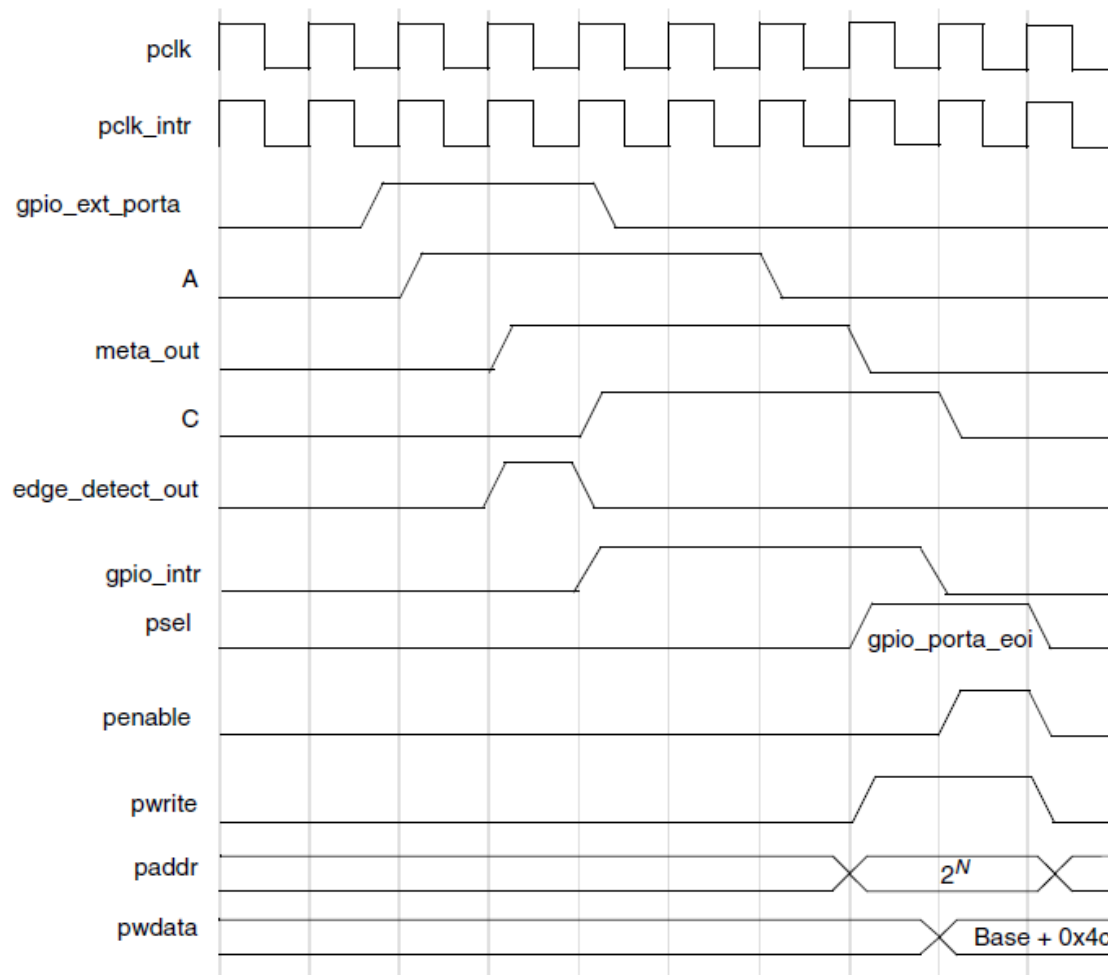


Figure 12-5-9 Interrupt Edge Detection and Interrupt Clear Timing when GPIO_SYNC_PA_INTERRUPTS = 1 (Metastability Included)

A case may arise where the Interrupt Service Routine (ISR) writes to the interrupt clear register in order to clear an existing interrupt during the same clock cycle in which a new interrupt is detected. In such a case, writing to the interrupt clear register clears only the first interrupt. The second interrupt is not lost, since setting an interrupt has a higher priority than clearing it.

Figure 12-5-10 shows a timing diagram similar to Figure 12-5-9, except that Metastability registers are removed. It also shows how an interrupt is cleared by a write to the interrupt clear register.

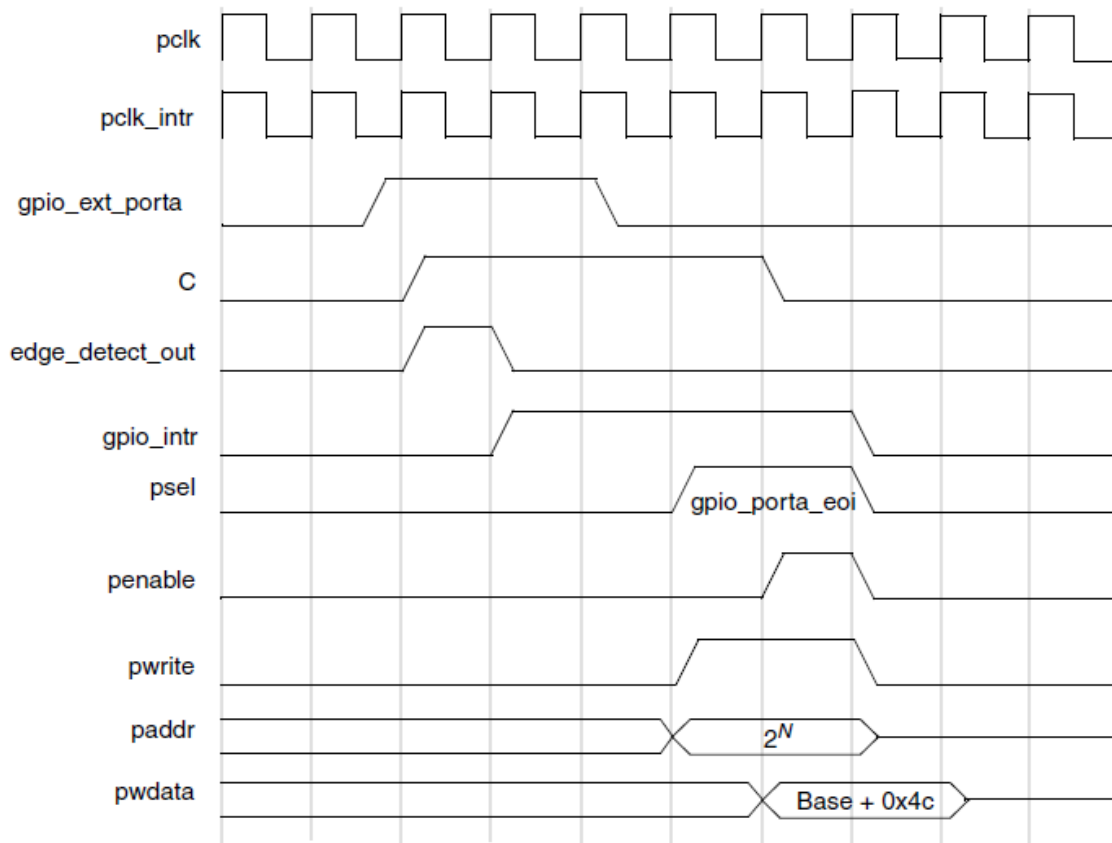


Figure 12-5-10 Interrupt Edge Detection and Interrupt Clear Timing when GPIO_SYNC_PA_INTERRUPTS = 0 (Metastability Removed)

Figure 12-5-11 shows such a case where the debounce logic is unused. In this timing diagram, meta_out and edge_detect_out are the outputs of the second metastability register and the edge detect logic, respectively. The second edge detection occurs on the same cycle as the write to the interrupt clear register. In this example, the write to the interrupt clear register does not clear the second interrupt, and the gpio_intr{_n} signal is not de-asserted.

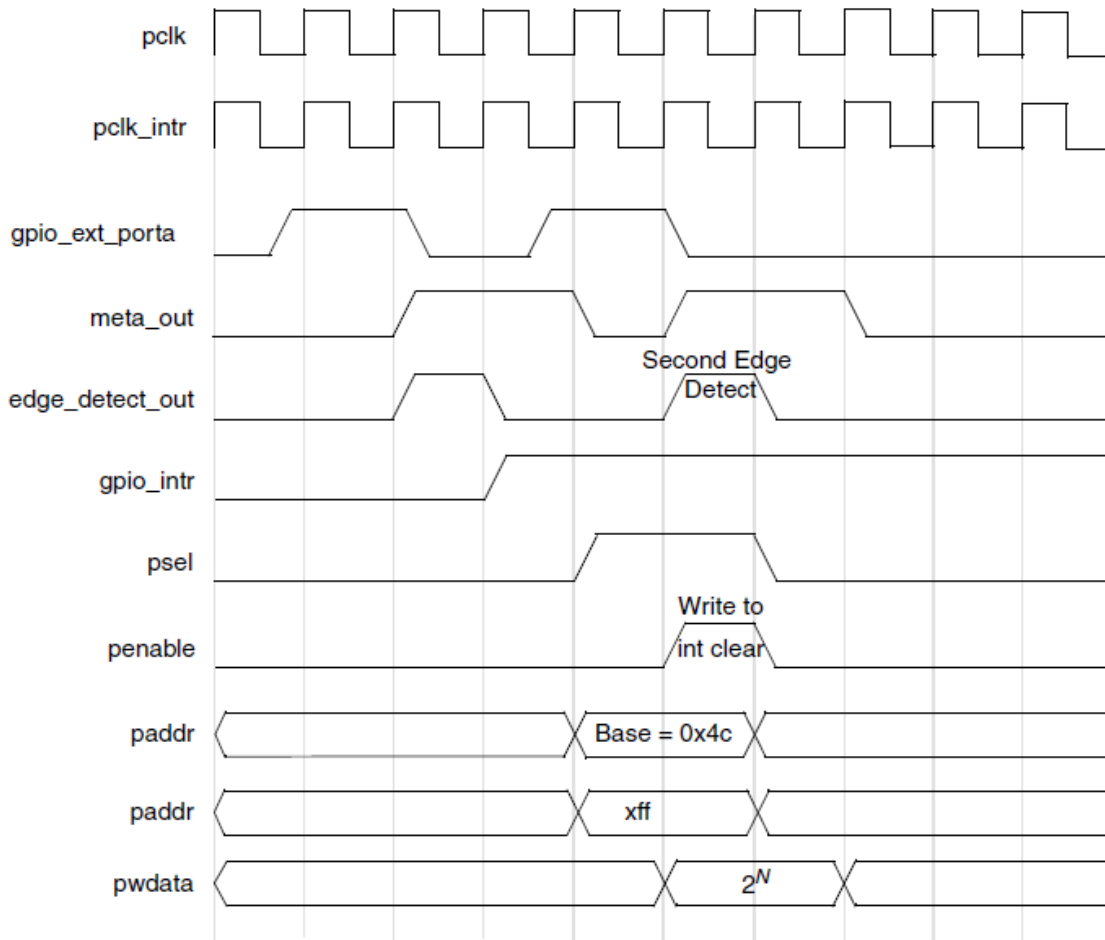


Figure 12-5-11 Write to Interrupt Clear Register, Coincident with Detection of New Interrupt

12.5.4.2.4 Interrupt Edge Detection - Both Edges

Interrupt detection logic for both the rising edge and the falling edge is available only when `GPIO_INT_BOTH_EDGE = 1` and this logic detects the interrupt on both the rising edge and the falling edge when the `gpio_int_bothedge` register is programmed to 1.

Figure 12-5-12 shows the synchronization and edge detect interrupt generation of the interrupt sources on `gpio_ext_portaN` signals when `GPIO_INT_BOTH_EDGE = 1`.

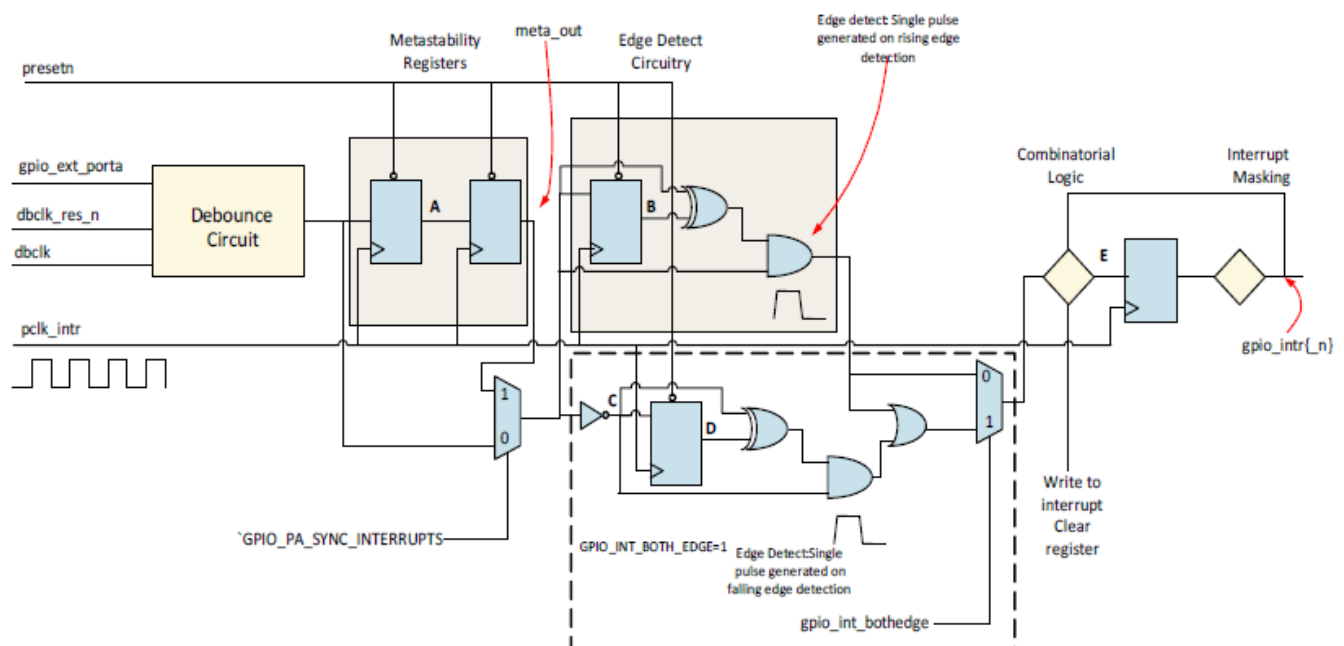


Figure 12-5-12 Synchronization and Edge Detect Interrupt Generation When GPIO_INT_BOTH_EDGE = 1

Figure 12-5-13 shows a timing diagram where an interrupt is generated on both the rising edge and the falling edge of an input on Port A, that is, with GPIO_INT_BOTH_EDGE = 1 and gpio_int_bothedge programmed to detect both edges. In this scenario, debounce logic is disabled and metastability registers are included. This figure also shows how an interrupt is cleared by a write to the interrupt clear register.

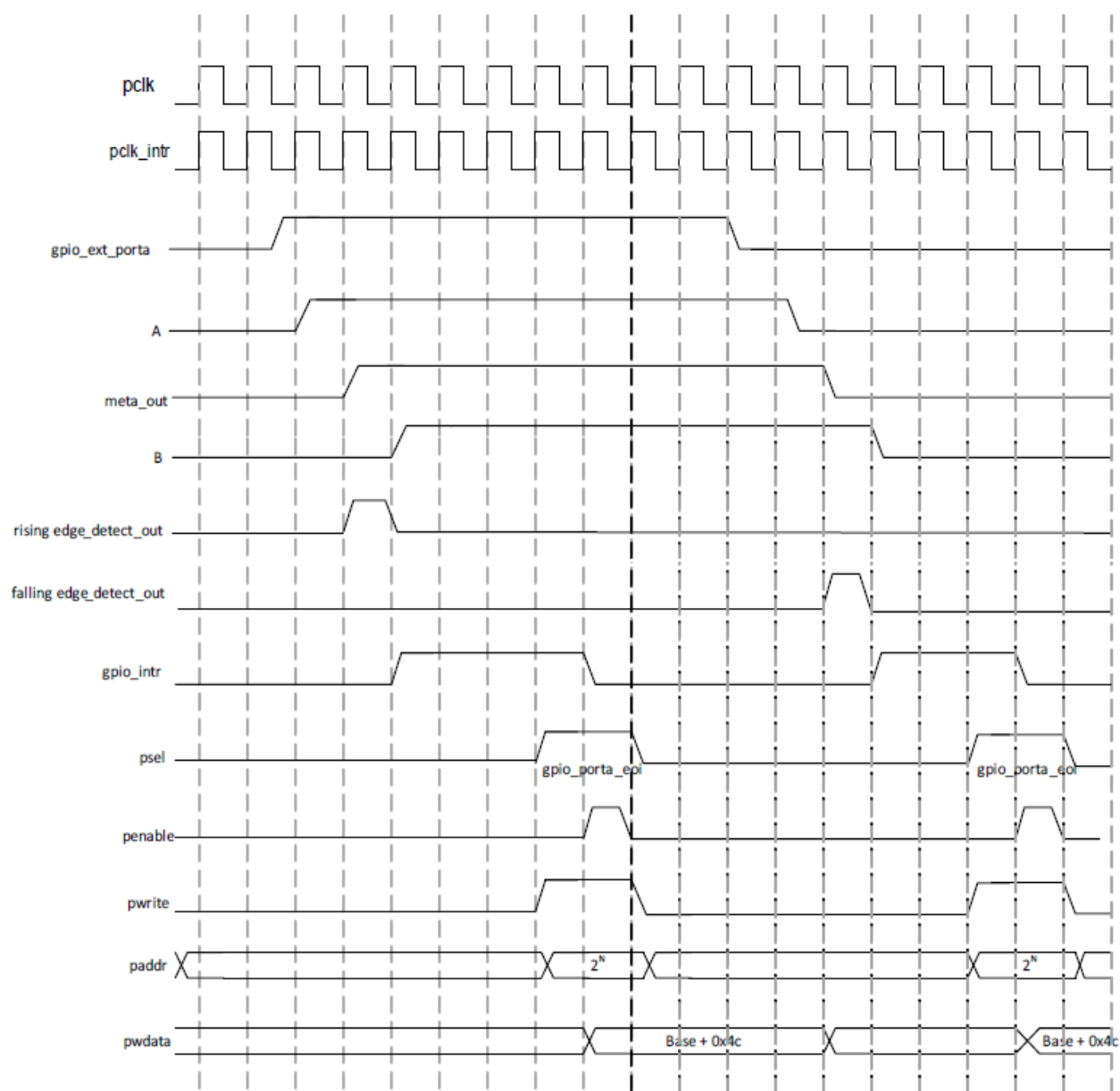


Figure 12-5-13 Interrupt Edge Detection and Interrupt Clear Timing When GPIO_SYNC_PA_INTERRUPTS = 1 and GPIO_INT_BOTH_EDGE = 1 (Metastability Included)

Figure 12-5-14 shows a timing diagram similar to Figure 3-12, except that in this scenario, metastability registers are removed.

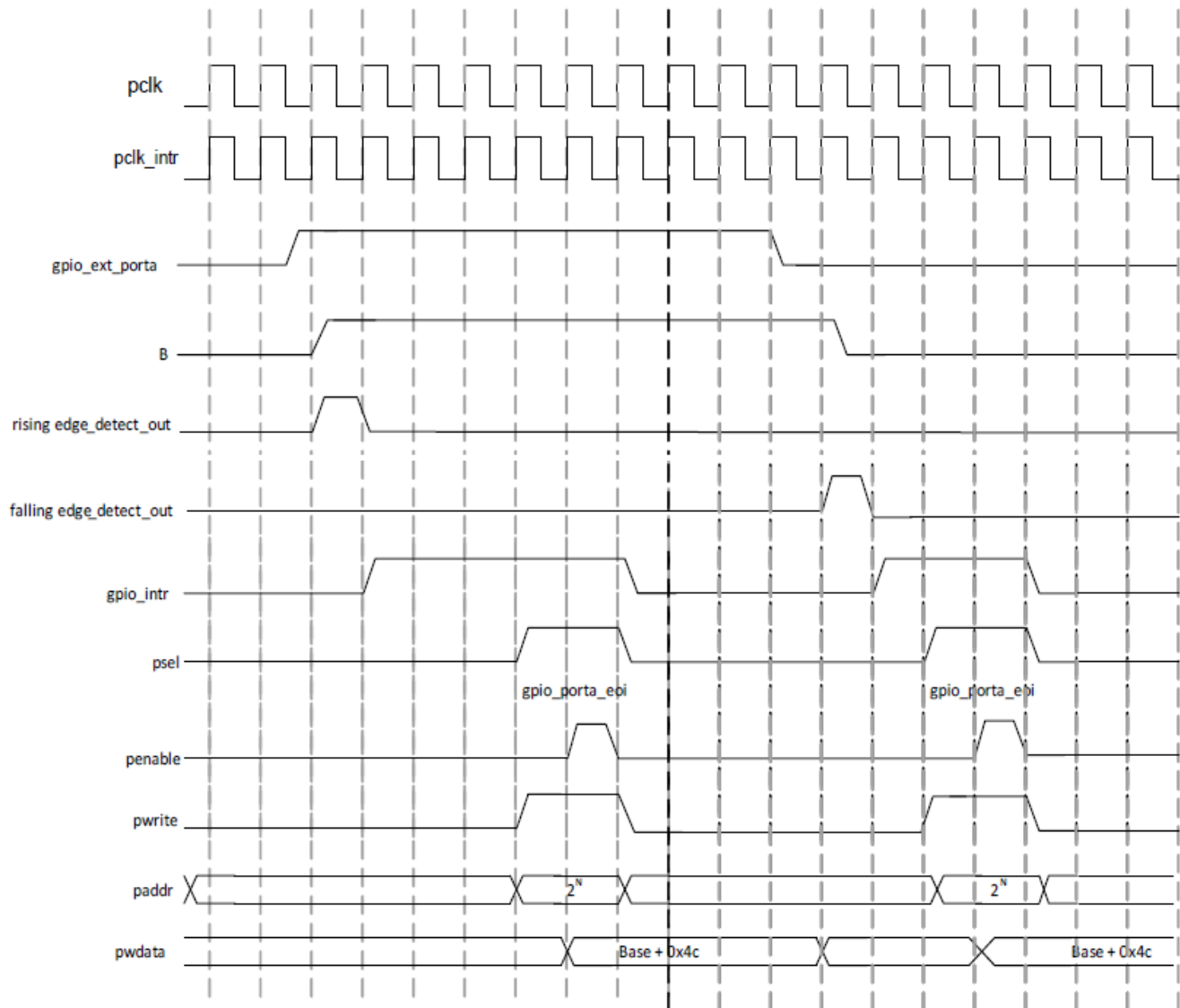


Figure 12-5-14 Interrupt Edge Detection and Interrupt Clear Timing When GPIO_SYNC_PA_INTERRUPTS = 0 and GPIO_INT_BOTH_EDGE = 1 (Metastability Removed)

12.5.4.2.5 Level-Sensitive Interrupts

Figure 12-5-15 shows the generation of level-sensitive interrupts. As for edge-detect interrupts, the user can configure GPIO with or without debounce logic.

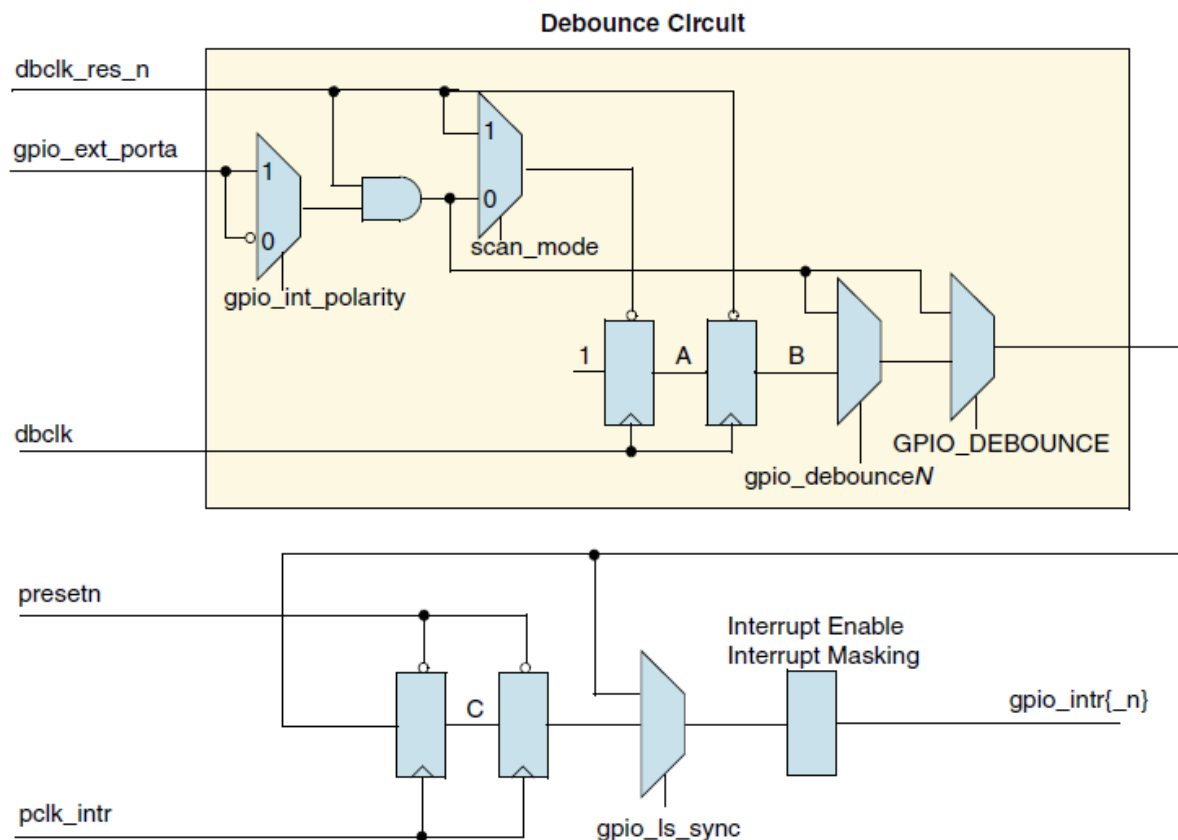


Figure 12-5-15 Level-Sensitive Interrupt RTL Diagram

With level-sensitive interrupts, there is a choice of whether they are synchronized to the interrupt clock pclk_intr or are entirely combinational (aside from the debounce circuitry). The selection is done by programming the gpio_ls_sync (GPIO Level Sensitive Synchronous) register.

This is a memory-mapped bit that inserts two metastability registers clocked off of pclk_intr to synchronize the level-sensitive interrupts to pclk_intr. When gpio_ls_sync is not asserted, then there is no guarantee that the interrupt lines are synchronous to pclk_intr. A processor status register may need to be set to indicate asynchronous interrupts. When gpio_ls_sync is asserted, then the pclk_intr clock must be present to pass the interrupt to the interrupt controller block. The gpio_intrclk_en output signal is asserted when level-sensitive interrupts that are to be synchronized to pclk_intr are selected. The gpio_intrclk_en signal can be used in the clock generation block to turn on pclk_intr.

The input signal is inverted for active-low level-sensitive interrupts. The same detection logic is used here as is used for active-high level-sensitive interrupts.

Figure 12-5-16 shows the generation of an active-low level-sensitive interrupt where the debounce circuitry is disabled.

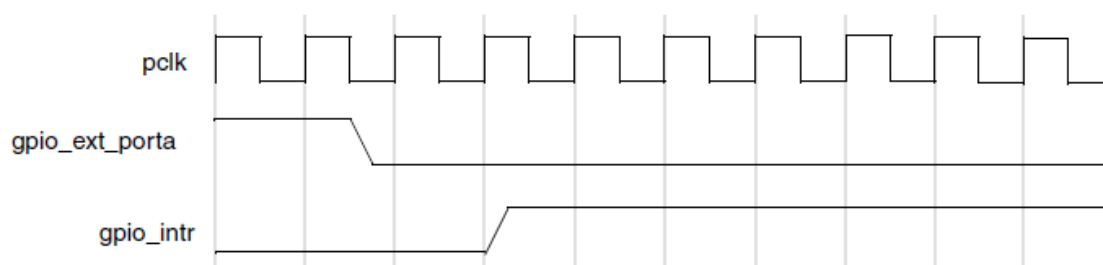


Figure 12-5-16 Active-Low Level-Sensitive Interrupt Generation Timing

12.5.5 Register Description

12.5.5.1 Register Summary

The GPIO base addresses are as follows:

- The base address for GIPO group0 registers is 0x9140_B000.
- The base address for GIPO group1 registers is 0x9140_C000.

Table 12-5-1 Summary of GPIO registers

Offset Address	Register	Description	Page
0x00	GPIO_SWPORTA_DR	Port A data register	12-5-2
0x04	GPIO_SWPORTA_DDR	Port A Data Direction Register	12-5-3
0x08	GPIO_SWPORTA_CTL	Name: Port A data source register	12-5-4
0x0c	GPIO_SWPORTB_DR	Port B data register	12-5-5
0x10	GPIO_SWPORTB_DDR	Port B Data Direction Register	12-5-6
0x14	GPIO_SWPORTB_CTL	Port B data source register	12-5-7
0x30	GPIO_INTEN	Interrupt enable register Note: This register is available only if Port A is configured...	12-5-8
0x34	GPIO_INTMASK	Interrupt mask register Note: This register is available only if Port A is configured to generate...	12-5-9
0x38	GPIO_INTTYPE_LEVEL	Interrupt level Note: This register is available only if Port A is configured to generate interrupts...	12-5-10
0x3c	GPIO_INT_POLARITY	Interrupt polarity Note: This register is available only if Port A is configured to generate interrupts...	12-5-11
0x40	GPIO_INTSTATUS	Interrupt status Note: This register is available only if Port A is configured to generate interrupts...	12-5-12
0x44	GPIO_RAW_INTSTATUS	Raw interrupt status Note: This register is available only if Port A is configured to generate...	12-5-13
0x48	GPIO_DEBOUNCE	Debounce enable Note: This register is available only if Port A is configured to generate interrupts...	12-5-14
0x4c	GPIO_PORTA_EOI	Port A clear interrupt register Note: This register is available only if Port A is configured to...	12-5-15
0x50	GPIO_EXT_PORTA	External port A register	12-5-16
0x54	GPIO_EXT_PORTB	Port B external port register	12-5-17
0x60	GPIO_LS_SYNC	Synchronization level	12-5-18

Offset Address	Register	Description	Page
0x64	GPIO_ID_CODE	GPIO ID code	12-5-19
0x68	GPIO_INT_BOTHEDGE	Interrupt Both Edge type Note: This register is available only if PORT A is configured to generate...	12-5-20
0x6c	GPIO_VER_ID_CODE	GPIO Component Version	12-5-21
0x70	GPIO_CONFIG_REG2	GPIO Configuration Register 2 This register is a read-only register that is present when the configuration...	12-5-22
0x74	GPIO_CONFIG_REG1	GPIO Configuration Register 1 This register is present when the configuration parameter GPIO_ADD_ENCODED_PARAMS...	12-5-23

12.5.5.2 Register Description

12.5.5.2.1 GPIO_SWPORTA_DR

- Name: Port A data register
- Description: Port A data register
- Offset Address: 0x00

Table 12-5-2 Fields for Register: GPIO_SWPORTA_DR

Bits	Access	Name	Description & Reset Value
31:y	R	RSVD_GPIO_SWPORTA_DR	RSVD_GPIO_SWPORTA_DR Reserved bits - read as zero Exists: Always Range Variable[y]: GPIO_PWIDTH_A
x:0	R/W	GPIO_SWPORTA_DR	Values written to this register are output on the I/O signals for Port A if the corresponding data direction bits for Port A are set to Output mode 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. Reset Value: GPIO_SWPORTA_RESET Exists: Always Range Variable[x]: GPIO_PWIDTH_A - 1

12.5.5.2.2 GPIO_SWPORTA_DDR

- Name: Port A Data Direction Register
- Description: Port A Data Direction Register.
- Offset Address: 0x04

Table 12-5-3 Fields for Register: GPIO_SWPORTA_DDR

Bits	Access	Name	Description & Reset Value
31:y	R	RSVD_GPIO_SWPORTA_DDR	RSVD_GPIO_SWPORTA_DDR Reserved bits - read as zero Exists: Always Range Variable[y]: GPIO_PWIDTH_A

Bits	Access	Name	Description & Reset Value
x:0	R/W	GPIO_SWPORTA_DDR	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. Values: 0x0 (IN): Input Direction 0x1 (OUT): Output Direction Value After Reset: {(GPIO_DFLT_DIR_A==1) ? (pow (2, GPIO_PWIDTH_A))-1 : 0} Exists: Always Range Variable[x]: GPIO_PWIDTH_A - 1

12.5.5.2.3 GPIO_SWPORTA_CTL

- Name: Name: Port A data source register
- Description: Name: Port A data source register.
- Offset Address: 0x08

Table 12-5-4 Fields for Register: GPIO_SWPORTA_CTL

Bits	Access	Name	Description & Reset Value
31:y	R	RSVD_GPIO_SWPORTA_CTL	RSVD_GPIO_SWPORTA_CTL Reserved bits - read as zero Exists: Always Range Variable[y]: GPIO_SWPORTA_CTL_REG_SIZE
x:0	R/W	GPIO_SWPORTA_CTL	The data and control source for a signal can come from either software or hardware; this bit selects between them. The default source is configurable through the GPIO_DFLT_DIR_A configuration parameter. If GPIO_PORTA_SINGLE_CTL = 0, the register will contain one bit for each bit of the signal. Upon reset in this case, the value of GPIO_DFLT_SRC_A is replicated across all bits of the signal so that all bits power up with the same operating mode. Furthermore, the default source of each bit of the signal can subsequently be changed by writing to the corresponding bit of this register. This register is not available unless GPIO_HW_PORTA = 1. Reset Value: If GPIO_PORTA_SINGLE_CTL = 1, then the reset value is GPIO_DFLT_SRC_A. If GPIO_PORTA_SINGLE_CTL = 0, then the reset value is {GPIO_PWIDTH_A{GPIO_DFLT_SRC_A in each bit}}. Values: 0x0 (SOFTWARE): Software Mode 0x1 (HARDWARE): Hardware Mode Exists: Always Range Variable[x]: GPIO_SWPORTA_CTL_REG_SIZE - 1

12.5.5.2.4 GPIO_SWPORTB_DR

- Name: Port B data register
- Description: Port B data register.

- Offset Address: 0x0c

Table 12-5-5 Fields for Register: GPIO_SWPORTB_DR

Bits	Access	Name	Description & Reset Value
31:y	R	RSVD_GPIO_SWPORTB_DR	RSVD_GPIO_SWPORTB_DR Reserved bits - read as zero Exists: Always Range Variable[y]: GPIO_PWIDTH_B
x:0	R/W	GPIO_SWPORTB_DR	Values written to this register are output on the I/O signals for Port B if the corresponding data direction bits for Port B are set to Output mode and the corresponding control bit for Port B is set to Software mode. The value read back is equal to the last value written to this register. Reset Value: GPIO_SWPORTB_RESET Exists: Always Range Variable[x]: GPIO_PWIDTH_B - 1

12.5.5.2.5 GPIO_SWPORTB_DDR

- Name: Port B Data Direction Register
- Description: Port B Data Direction Register.
- Offset Address: 0x10

Table 12-5-6 Fields for Register: GPIO_SWPORTB_DDR

Bits	Access	Name	Description & Reset Value
31:y	R	RSVD_GPIO_SWPORTB_DDR	RSVD_GPIO_SWPORTB_DDR Reserved bits - read as zero Exists: Always Range Variable[y]: GPIO_PWIDTH_B
x:0	R/W	GPIO_SWPORTB_DDR	Values written to this register independently control the direction of the corresponding data bit in Port B. The default direction can be configured as input or output after system reset through the GPIO_DFLT_DIR_B parameter. Values: 0x0 (IN): Input Direction (default) 0x1 (OUT): Output Direction Value After Reset: {(GPIO_DFLT_DIR_B==1) ? (pow (2, GPIO_PWIDTH_B))-1 : 0} Exists: Always Range Variable[x]: GPIO_PWIDTH_B - 1

12.5.5.2.6 GPIO_SWPORTB_CTL

- Name: Port B data source register
- Description: Port B data source register.
- Offset Address: 0x14

Table 12-5-7 Fields for Register: GPIO_SWPORTB_CTL

Bits	Access	Name	Description & Reset Value

Bits	Access	Name	Description & Reset Value
31:y	R	RSVD_GPIO_SWPORTB_CTL	RSVD_GPIO_SWPORTB_CTL Reserved bits - read as zero Exists: Always Range Variable[y]: GPIO_SWPORTB_CTL_REG_SIZE
x:0	R/W	GPIO_SWPORTB_CTL	The data and control source for a signal can come from either software or hardware; this bit selects between them. The default source is configurable through the GPIO_DFLT_DIR_B configuration parameter. If GPIO_PORTB_SINGLE_CTL = 0, the register will contain one bit for each bit of the signal. Upon reset in this case, the value of GPIO_DFLT_SRC_B is replicated across all bits of the signal so that all bits power up with the same operating mode. Furthermore, the default source of each bit of the signal can subsequently be changed by writing to the corresponding bit of this register. This register is not available unless GPIO_HW_PORTB = 1. Reset Value: If GPIO_PORTB_SINGLE_CTL = 1, then the reset value is GPIO_DFLT_SRC_B. If GPIO_PORTB_SINGLE_CTL = 0, then the reset value is {GPIO_PWIDTH_B{GPIO_DFLT_SRC_B in each bit}}. Values: 0x0 (SOFTWARE): Software Mode (default) 0x1 (HARDWARE): Hardware Mode Exists: Always Range Variable[x]: GPIO_SWPORTB_CTL_REG_SIZE - 1

12.5.5.2.7 GPIO_INTEN

- Name: Interrupt enable register
- Description: Interrupt enable register. Note: This register is available only if Port A is configured to generate interrupts (GPIO_PORTA_INTR = Include (1)).
- Offset Address: 0x30

Table 12-5-8 Fields for Register: GPIO_INTEN

Bits	Access	Name	Description & Reset Value
31:y	R	RSVD_GPIO_INTEN	RSVD_GPIO_INTEN Reserved bits - read as zero Exists: Always Range Variable[y]: GPIO_PWIDTH_A

Bits	Access	Name	Description & Reset Value
x:0	R/W	GPIO_INTEN	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. Reset Value: 0x0 Values: 0x0 (DISABLED): Interrupt is disabled 0x1 (ENABLED): Interrupt is enabled Exists: Always Range Variable[x]: GPIO_PWIDTH_A - 1

12.5.5.2.8 GPIO_INTMASK

- Name: Interrupt mask register
- Description: Interrupt mask register. Note: This register is available only if Port A is configured to generate interrupts (GPIO_PORTA_INTR = Include (1)).
- Offset Address: 0x34

Table 12-5-9 Fields for Register: GPIO_INTMASK

Bits	Access	Name	Description & Reset Value
31:y	R	RSVD_GPIO_INTMASK	RSVD_GPIO_INTMASK Reserved bits - read as zero Exists: Always Range Variable[y]: GPIO_PWIDTH_A
x:0	R/W	GPIO_INTMASK	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. Reset Value: 0x0 Values: 0x0 (DISABLED): Interrupt bits are unmasked 0x1 (ENABLED): Mask interrupt Exists: Always Range Variable[x]: GPIO_PWIDTH_A - 1

12.5.5.2.9 GPIO_INTTYPE_LEVEL

- Name: Interrupt level
- Description: Interrupt level. Note: This register is available only if Port A is configured to generate interrupts (GPIO_PORTA_INTR = Include (1)).
- Offset Address: 0x38

Table 12-5-10 Fields for Register: GPIO_INTTYPE_LEVEL

Bits	Access	Name	Description & Reset Value

Bits	Access	Name	Description & Reset Value
31:y	R	RSVD_GPIO_INTTYPE_LEVEL	RSVD_GPIO_INTTYPE_LEVEL Reserved bits - read as zero Exists: Always Range Variable[y]: GPIO_PWIDTH_A
x:0	R/W	GPIO_INTTYPE_LEVEL	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. Reset Value: 0x0 Values: 0x0 (LEVEL_SENSITIVE): Interrupt is level sensitive 0x1 (EDGE_SENSITIVE): Interrupt is edge sensitive Exists: Always Range Variable[x]: GPIO_PWIDTH_A - 1

12.5.5.2.10 GPIO_INT_POLARITY

- Name: Interrupt polarity
- Description: Interrupt polarity. Note: This register is available only if Port A is configured to generate interrupts (GPIO_PORTA_INTR = Include (1)).
- Offset Address: 0x3c

Table 12-5-11 Fields for Register: GPIO_INT_POLARITY

Bits	Access	Name	Description & Reset Value
31:y	R	RSVD_GPIO_INT_POLARITY	RSVD_GPIO_INT_POLARITY Reserved bits - read as zero Exists: Always Range Variable[y]: GPIO_PWIDTH_A
x:0	R/W	GPIO_INT_POLARITY	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. Reset Value: 0x0 Values: 0x0 (ACTIVE_LOW): Active Low polarity 0x1 (ACTIVE_HIGH): Active High polarity Exists: Always Range Variable[x]: GPIO_PWIDTH_A - 1

12.5.5.2.11 GPIO_INTSTATUS

- Name: Interrupt status
- Description: Interrupt status. Note: This register is available only if Port A is configured to generate interrupts (GPIO_PORTA_INTR = Include (1)).
- Offset Address: 0x40

Table 12-5-12 Fields for Register: GPIO_INTSTATUS

Bits	Access	Name	Description & Reset Value

Bits	Access	Name	Description & Reset Value
31:y	R	RSVD_GPIO_INTSTATUS	RSVD_GPIO_INTSTATUS Reserved bits - read as zero Exists: Always Volatile: true Range Variable[y]: GPIO_PWIDTH_A
x:0	R	GPIO_INTSTATUS	Interrupt status of Port A. Reset Value: 0x0 Values: 0x0 (INACTIVE): Inactive 0x1 (ACTIVE): Active Exists: Always Volatile: true Range Variable[x]: GPIO_PWIDTH_A - 1

12.5.5.2.12 GPIO_RAW_INTSTATUS

- Name: Raw interrupt status
- Description: Raw interrupt status. Note: This register is available only if Port A is configured to generate interrupts (GPIO_PORTA_INTR = Include (1)).
- Offset Address: 0x44

Table 12-5-13 Fields for Register: GPIO_RAW_INTSTATUS

Bits	Access	Name	Description & Reset Value
31:y	R	RSVD_GPIO_RAW_INTMASK	RSVD_GPIO_RAW_INTMASK Reserved bits - read as zero Exists: Always Volatile: true Range Variable[y]: GPIO_PWIDTH_A
x:0	R	GPIO_RAW_INTSTATUS	Raw interrupt of status of Port A (premasking bits) Reset Value: 0x0 Values: 0x0 (INACTIVE): Inactive 0x1 (ACTIVE): Active Exists: Always Volatile: true Range Variable[x]: GPIO_PWIDTH_A - 1

12.5.5.2.13 GPIO_DEBOUNCE

- Name: Debounce enable
- Description: Debounce enable. Note: This register is available only if Port A is configured to generate interrupts (GPIO_PORTA_INTR = Include (1)) and when the debounce logic is included (GPIO_DEBOUNCE = Include (1)).
- Offset Address: 0x48

Table 12-5-14 Fields for Register: GPIO_DEBOUNCE

Bits	Access	Name	Description & Reset Value
31:y	R	RSVD_GPIO_DEBOUNCE	RSVD_GPIO_DEBOUNCE Reserved bits - read as zero Exists: Always Range Variable[y]: GPIO_PWIDTH_A

Bits	Access	Name	Description & Reset Value
x:0	R/W	GPIO_DEBOUNCE	Controls whether an external signal that is the source of an interrupt needs to be debounced to remove any spurious glitches. Writing a 1 to a bit in this register enables the debouncing circuitry. A signal must be valid for two periods of an external clock before it is internally processed. Reset Value: 0x0 Values: 0x0 (DISABLED): No debounce 0x1 (ENABLED): Enable debounce Exists: Always Range Variable[x]: GPIO_PWIDTH_A - 1

12.5.5.2.14 GPIO_PORTA_EOI

- Name: Port A clear interrupt register
- Description: Port A clear interrupt register. Note: This register is available only if Port A is configured to generate interrupts (GPIO_PORTA_INTR = Include (1)) and when the debounce logic is included (GPIO_DEBOUNCE = Include (1)).
- Offset Address: 0x4c

Table 12-5-15 Fields for Register: GPIO_PORTA_EOI

Bits	Access	Name	Description & Reset Value
31:y	W	RSVD_GPIO_PORTA_EOI	RSVD_GPIO_PORTA_EOI Reserved bits - read as zero Exists: Always Range Variable[y]: GPIO_PWIDTH_A
x:0	W	GPIO_PORTA_EOI	Controls the clearing of edge type interrupts from Port A. When a 1 is written into a corresponding bit of this register, the interrupt is cleared. All interrupts are cleared when Port A is not configured for interrupts. Reset Value: 0x0 Values: 0x0 (DISABLED): No interrupt clear 0x1 (ENABLED): Clear Interrupt Exists: Always Range Variable[x]: GPIO_PWIDTH_A - 1

12.5.5.2.15 GPIO_EXT_PORTA

- Name: External port A register
- Description: External port A register.
- Offset Address: 0x50

Table 12-5-16 Fields for Register: GPIO_EXT_PORTA

Bits	Access	Name	Description & Reset Value

Bits	Access	Name	Description & Reset Value
31:y	R	RSVD_GPIO_EXT_PORTA	RSVD_GPIO_EXT_PORTA Reserved bits - read as zero Exists: Always Volatile: true Range Variable[y]: GPIO_PWIDTH_A
x:0	R	GPIO_EXT_PORTA	This register always reflects the signals value on the External Port A. Reset Value: 0x0 Exists: Always Volatile: true Range Variable[x]: GPIO_PWIDTH_A - 1

12.5.5.2.16 GPIO_EXT_PORTB

- Name: Port B external port register
- Description: Port B external port register.
- Offset Address: 0x54

Table 12-5-17 Fields for Register: GPIO_EXT_PORTB

Bits	Access	Name	Description & Reset Value
31:y	R	RSVD_GPIO_EXT_PORTB	RSVD_GPIO_EXT_PORTB Reserved bits - read as zero Exists: Always Volatile: true Range Variable[y]: GPIO_PWIDTH_B
x:0	R	GPIO_EXT_PORTB	This register always reflects the signals value on the External Port B. Reset Value: 0x0 Exists: Always Volatile: true Range Variable[x]: GPIO_PWIDTH_B - 1

12.5.5.2.17 GPIO_LS_SYNC

- Name: Synchronization level
- Description: Synchronization level.
- Offset Address: 0x60

Table 12-5-18 Fields for Register: GPIO_LS_SYNC

Bits	Access	Name	Description & Reset Value
31:1	R	RSVD_GPIO_LS_SYNC	RSVD_GPIO_LS_SYNC Reserved bits - read as zero Exists: Always
0	*Varies	GPIO_LS_SYNC	Writing a 1 to this register results in all level-sensitive interrupts being synchronized to pclk_intr. Reset Value: 0x0 Values: 0x0 (DISABLED): No synchronization to pclk_int (default)r 0x1 (ENABLED): Synchronize to pclk_intr Exists: Always Memory Access: (GPIO_PORTA_INTR==1) ? read- write : read-only

12.5.5.2.18 GPIO_ID_CODE

- Name: GPIO ID code
- Description: GPIO ID code.
- Offset Address: 0x64

Table 12-5-19 Fields for Register: GPIO_ID_CODE

Bits	Access	Name	Description & Reset Value
31:y	R	RSVD_GPIO_ID_CODE	RSVD_GPIO_ID_CODE Reserved bits - read as zero Exists: Always Range Variable[y]: GPIO_ID_WIDTH
x:0	R	GPIO_ID_CODE	This is a user-specified code that a system can read. It can be used for chip identification, and so on. Reset Value: GPIO_ID_NUM Exists: Always Range Variable[x]: GPIO_ID_WIDTH - 1

12.5.5.2.19 GPIO_INT_BOTHEDGE

- Name: Interrupt Both Edge type
- Description: Interrupt Both Edge type. Note: This register is available only if PORT A is configured to generate interrupts (GPIO_PORTA_INTR = Include(1)) and interrupt detection is configured to generate on both rising and falling edges of external input signal (GPIO_INT_BOTH_EDGE = Include(1)).
- Offset Address: 0x68

Table 12-5-20 Fields for Register: GPIO_INT_BOTHEDGE

Bits	Access	Name	Description & Reset Value
31:y	R	RSVD_GPIO_INT_BOTHEDGE	RSVD_GPIO_INT_BOTHEDGE Reserved bits - read as zero Exists: Always Range Variable[y]: GPIO_PWIDTH_A
x:0	R/W	GPIO_INT_BOTHEDGE	Controls the edge type of interrupt that can occur on Port A. - Whenever a particular bit is programmed to 1, it enables the generation of interrupts on both the rising edge and the falling edge of an external input signal corresponding to that bit on port A. - The values programmed in the registers gpio_inttype_level and gpio_int_polarity for this particular bit are not considered when the corresponding bit of this register is set to 1. - Whenever a particular bit is programmed to 0, the interrupt type depends on the value of the corresponding bits in the gpio_inttype_level and gpio_int_polarity registers. Reset Value: 0x0 Values: 0x0 (DISABLED): single edge sensitive 0x1 (ENABLED): both edge sensitive Exists: Always Range Variable[x]: GPIO_PWIDTH_A - 1

12.5.5.2.20 GPIO_VER_ID_CODE

- Name: GPIO Component Version

- Description: GPIO Component Version.
- Offset Address: 0x6c

Table 12-5-21 Fields for Register: GPIO_VER_ID_CODE

Bits	Access	Name	Description & Reset Value
31:0	R	GPIO_VER_ID_CODE	ASCII value for each number in the version, followed by . <i>For example 32_31_32_2A represents the version 2.12.</i> Reset Value: See the releases table in the Release Notes Exists: Always

12.5.5.2.21 GPIO_CONFIG_REG2

- Name: GPIO Configuration Register 2
- Description: GPIO Configuration Register 2. This register is a read-only register that is present when the configuration parameter GPIO_ADD_ENCODED_PARAMS is set to True. If this configuration is set to False, then this register reads back 0.
Reset Value: ((GPIO_ADD_ENCODED_PARAMS==1'b0) ? 32'h0 : {12'h0, GPIO_ENCODED_PWIDTH_D, GPIO_ENCODED_PWIDTH_C, GPIO_ENCODED_PWIDTH_B, GPIO_ENCODED_PWIDTH_A}).
- Offset Address: 0x70

Table 12-5-22 Fields for Register: GPIO_CONFIG_REG2

Bits	Access	Name	Description & Reset Value
31:20	R	RSVD_GPIO_CONFIG_REG2	RSVD_GPIO_CONFIG_REG2 Reserved bits - read as zero Exists: Always
19:15	R	ENCODED_ID_PWIDTH_D	The value of this register is derived from the GPIO_PWIDTH_D configuration parameter. 0x0 = 8 bits 0x1 = 16 bits 0x2 = 32 bits 0x3 = Reserved Exists: Always
14:10	R	ENCODED_ID_PWIDTH_C	The value of this register is derived from the GPIO_PWIDTH_C configuration parameter. 0x0 = 8 bits 0x1 = 16 bits 0x2 = 32 bits 0x3 = Reserved Exists: Always
9:5	R	ENCODED_ID_PWIDTH_B	The value of this register is derived from the GPIO_PWIDTH_B configuration parameter. 0x0 = 8 bits 0x1 = 16 bits 0x2 = 32 bits 0x3 = Reserved Exists: Always
4:0	R	ENCODED_ID_PWIDTH_A	The value of this register is derived from the GPIO_PWIDTH_A configuration parameter. 0x0 = 8 bits 0x1 = 16 bits 0x2 = 32 bits 0x3 = Reserved Exists: Always

12.5.5.2.22 GPIO_CONFIG_REG1

- Name: GPIO Configuration Register 1
- Description: GPIO Configuration Register 1. This register is present when the configuration parameter GPIO_ADD_ENCODED_PARAMS is set to True. If this parameter is set to False, this register reads back zero

(0).

Reset Value: GPIO_CID_RESET_REG1

- Offset Address: 0x74

Table 12-5-23 Fields for Register: GPIO_CONFIG_REG1

Bits	Access	Name	Description & Reset Value
31:22	R	RSVD_GPIO_CONFIG_REG1	RSVD_GPIO_CONFIG_REG1 Reserved bits - read as zero Exists: Always
21	R	INTERRUPT_BOTH_EDGE_TYPE	The value of this register is derived from the GPIO_INT_BOTH_EDGE configuration parameter Values: 0x0 (DISABLED): Interrupt generation on rising or falling edge 0x1 (ENABLED): Interrupt generation on both rising and falling edge Exists: Always
20:16	R	ENCODED_ID_WIDTH	The value of this register is derived from the GPIO_ID_WIDTH configuration parameter. Exists: Always
15	R	GPIO_ID	The value of this register is derived from the GPIO_ID configuration parameter. Values: 0x0 (DISABLED): GPIO_ID not included 0x1 (ENABLED): GPIO_ID is included Exists: Always
14	R	ADD_ENCODED_PARAMS	The value of this register is derived from the GPIO_ADD_ENCODED_PARAMS configuration parameter. Values: 0x0 (DISABLED): Encoded parameters not added 0x1 (ENABLED): Encoded parameters added Exists: Always
13	R	DEBOUNCE	The value of this register is derived from the GPIO_DEBOUNCE configuration parameter. Values: 0x0 (DISABLED): Exclude debounce capability 0x1 (ENABLED): Include debounce capability Exists: Always
12	R	PORTA_INTR	The value of this register is derived from the GPIO_PORTA_INTR configuration parameter. Values: 0x0 (DISABLED): PORT A is not used as an interrupt source 0x1 (ENABLED): PORT A is required to be used as an interrupt source Exists: Always

Bits	Access	Name	Description & Reset Value
11	R	HW_PORTD	The value of this register is derived from the GPIO_HW_PORTD configuration parameter. Values: 0x0 (DISABLED): Port D has external, auxiliary hardware signals excluded 0x1 (ENABLED): Port D has external, auxiliary hardware signals included Exists: Always
10	R	HW_PORTC	The value of this register is derived from the GPIO_HW_PORTC configuration parameter. Values: 0x0 (DISABLED): Port C has external, auxiliary hardware signals excluded 0x1 (ENABLED): Port C has external, auxiliary hardware signals included Exists: Always
9	R	HW_PORTB	The value of this register is derived from the GPIO_HW_PORTB configuration parameter. Values: 0x0 (DISABLED): Port B has external, auxiliary hardware signals excluded 0x1 (ENABLED): Port B has external, auxiliary hardware signals included Exists: Always
8	R	HW_PORTA	The value of this register is derived from the GPIO_HW_PORTA configuration parameter. Values: 0x0 (DISABLED): Port A has external, auxiliary hardware signals excluded 0x1 (ENABLED): Port A has external, auxiliary hardware signals included Exists: Always
7	R	PORTD_SINGLE_CTL	The value of this register is derived from the GPIO_PORTD_SINGLE_CTL configuration parameter. Values: 0x0 (DISABLED): PORTD is not controlled from a single source 0x1 (ENABLED): PORTD is controlled from a single source Exists: Always
6	R	PORTC_SINGLE_CTL	The value of this register is derived from the GPIO_PORTC_SINGLE_CTL configuration parameter. Values: 0x0 (DISABLED): PORTC is not controlled from a single source 0x1 (ENABLED): PORTC is controlled from a single source Exists: Always

Bits	Access	Name	Description & Reset Value
5	R	PORTB_SINGLE_CTL	The value of this register is derived from the GPIO_PORTB_SINGLE_CTL configuration parameter. Values: 0x0 (DISABLED): PORTB is not controlled from a single source 0x1 (ENABLED): PORTB is controlled from a single source Exists: Always
4	R	PORTA_SINGLE_CTL	The value of this register is derived from the GPIO_PORTA_SINGLE_CTL configuration parameter. Values: 0x0 (DISABLED): PORTA is not controlled from a single source 0x1 (ENABLED): PORTA is controlled from a single source Exists: Always
3:2	R	NUM_PORTS	The value of this register is derived from the GPIO_NUM_PORT configuration parameter. Values: 0x0 (NUM_PORTS_1): Number of ports is 1 0x1 (NUM_PORTS_2): Number of ports is 2 0x2 (NUM_PORTS_3): Number of ports is 3 0x3 (NUM_PORTS_4): Number of ports is 4 Exists: Always
1:0	R	APB_DATA_WIDTH	The value of this register is derived from the GPIO_APB_DATA_WIDTH configuration parameter. Note: 0x3 = Reserved Values: 0x0 (APB_8BITS): APB DATA WIDTH is 8 bits 0x1 (APB_16BITS): APB DATA WIDTH is 16 bits 0x2 (APB_32BITS): APB DATA WIDTH is 32 bits Exists: Always

12.5.6 Interface Description

Table 12-5-24 GPIO Interface Description

Module pin	Direction	PAD name	IOMUX
gpio0_gpio_ext_porta[0]	I/O	PAD_IO_0	IO_0_sel == 3'd1
gpio0_gpio_ext_porta[1]	I/O	PAD_IO_1	IO_1_sel == 3'd1
gpio0_gpio_ext_porta[2]	I/O	PAD_IO_2	IO_2_sel == 3'd1
gpio0_gpio_ext_porta[3]	I/O	PAD_IO_3	IO_3_sel == 3'd1
gpio0_gpio_ext_porta[4]	I/O	PAD_IO_4	IO_4_sel == 3'd1
gpio0_gpio_ext_porta[5]	I/O	PAD_IO_5	IO_5_sel == 3'd1
gpio0_gpio_ext_porta[6]	I/O	PAD_IO_6	IO_6_sel == 3'd1

Module pin	Direction	PAD name	IOMUX
gpio0_gpio_ext_porta[7]	I/O	PAD_IO_7	IO_7_sel == 3'd1
gpio0_gpio_ext_porta[8]	I/O	PAD_IO_8	IO_8_sel == 3'd1
gpio0_gpio_ext_porta[9]	I/O	PAD_IO_9	IO_9_sel == 3'd1
gpio0_gpio_ext_porta[10]	I/O	PAD_IO_10	IO_10_sel == 3'd1
gpio0_gpio_ext_porta[11]	I/O	PAD_IO_11	IO_11_sel == 3'd1
gpio0_gpio_ext_porta[12]	I/O	PAD_IO_12	IO_12_sel == 3'd1
gpio0_gpio_ext_porta[13]	I/O	PAD_IO_13	IO_13_sel == 3'd1
gpio0_gpio_ext_porta[14]	I/O	PAD_IO_14	IO_14_sel == 3'd2
gpio0_gpio_ext_porta[15]	I/O	PAD_IO_15	IO_15_sel == 3'd2
gpio0_gpio_ext_porta[16]	I/O	PAD_IO_16	IO_16_sel == 3'd1
gpio0_gpio_ext_porta[17]	I/O	PAD_IO_17	IO_17_sel == 3'd1
gpio0_gpio_ext_porta[18]	I/O	PAD_IO_18	IO_18_sel == 3'd0
gpio0_gpio_ext_porta[19]	I/O	PAD_IO_19	IO_19_sel == 3'd0
gpio0_gpio_ext_porta[20]	I/O	PAD_IO_20	IO_20_sel == 3'd0
gpio0_gpio_ext_porta[21]	I/O	PAD_IO_21	IO_21_sel == 3'd0
gpio0_gpio_ext_porta[22]	I/O	PAD_IO_22	IO_22_sel == 3'd0
gpio0_gpio_ext_porta[23]	I/O	PAD_IO_23	IO_23_sel == 3'd0
gpio0_gpio_ext_porta[24]	I/O	PAD_IO_24	IO_24_sel == 3'd0
gpio0_gpio_ext_porta[25]	I/O	PAD_IO_25	IO_25_sel == 3'd0
gpio0_gpio_ext_porta[26]	I/O	PAD_IO_26	IO_26_sel == 3'd0
gpio0_gpio_ext_porta[27]	I/O	PAD_IO_27	IO_27_sel == 3'd0
gpio0_gpio_ext_porta[28]	I/O	PAD_IO_28	IO_28_sel == 3'd0
gpio0_gpio_ext_porta[29]	I/O	PAD_IO_29	IO_29_sel == 3'd0
gpio0_gpio_ext_porta[30]	I/O	PAD_IO_30	IO_30_sel == 3'd0
gpio0_gpio_ext_porta[31]	I/O	PAD_IO_31	IO_31_sel == 3'd0
gpio1_gpio_ext_porta[0]	I/O	PAD_IO_32	IO_32_sel == 3'd1
gpio1_gpio_ext_porta[1]	I/O	PAD_IO_33	IO_33_sel == 3'd1
gpio1_gpio_ext_porta[2]	I/O	PAD_IO_34	IO_34_sel == 3'd1
gpio1_gpio_ext_porta[3]	I/O	PAD_IO_35	IO_35_sel == 3'd1
gpio1_gpio_ext_porta[4]	I/O	PAD_IO_36	IO_36_sel == 3'd1
gpio1_gpio_ext_porta[5]	I/O	PAD_IO_37	IO_37_sel == 3'd1
gpio1_gpio_ext_porta[6]	I/O	PAD_IO_38	IO_38_sel == 3'd1
gpio1_gpio_ext_porta[7]	I/O	PAD_IO_39	IO_39_sel == 3'd1
gpio1_gpio_ext_porta[8]	I/O	PAD_IO_40	IO_40_sel == 3'd1

Module pin	Direction	PAD name	IOMUX
gpio1_gpio_ext_porta[9]	I/O	PAD_IO_41	IO_41_sel == 3'd1
gpio1_gpio_ext_porta[10]	I/O	PAD_IO_42	IO_42_sel == 3'd1
gpio1_gpio_ext_porta[11]	I/O	PAD_IO_43	IO_43_sel == 3'd1
gpio1_gpio_ext_porta[12]	I/O	PAD_IO_44	IO_44_sel == 3'd1
gpio1_gpio_ext_porta[13]	I/O	PAD_IO_45	IO_45_sel == 3'd1
gpio1_gpio_ext_porta[14]	I/O	PAD_IO_46	IO_46_sel == 3'd2
gpio1_gpio_ext_porta[15]	I/O	PAD_IO_47	IO_47_sel == 3'd2
gpio1_gpio_ext_porta[16]	I/O	PAD_IO_48	IO_48_sel == 3'd1
gpio1_gpio_ext_porta[17]	I/O	PAD_IO_49	IO_49_sel == 3'd1
gpio1_gpio_ext_porta[18]	I/O	PAD_IO_50	IO_50_sel == 3'd0
gpio1_gpio_ext_porta[19]	I/O	PAD_IO_51	IO_51_sel == 3'd0
gpio1_gpio_ext_porta[20]	I/O	PAD_IO_52	IO_52_sel == 3'd0
gpio1_gpio_ext_porta[21]	I/O	PAD_IO_53	IO_53_sel == 3'd0
gpio1_gpio_ext_porta[22]	I/O	PAD_IO_54	IO_54_sel == 3'd0
gpio1_gpio_ext_porta[23]	I/O	PAD_IO_55	IO_55_sel == 3'd0
gpio1_gpio_ext_porta[24]	I/O	PAD_IO_56	IO_56_sel == 3'd0
gpio1_gpio_ext_porta[25]	I/O	PAD_IO_57	IO_57_sel == 3'd0
gpio1_gpio_ext_porta[26]	I/O	PAD_IO_58	IO_58_sel == 3'd0
gpio1_gpio_ext_porta[27]	I/O	PAD_IO_59	IO_59_sel == 3'd0
gpio1_gpio_ext_porta[28]	I/O	PAD_IO_60	IO_60_sel == 3'd0
gpio1_gpio_ext_porta[29]	I/O	PAD_IO_61	IO_61_sel == 3'd0
gpio1_gpio_ext_porta[30]	I/O	PAD_IO_62	IO_62_sel == 3'd0
gpio1_gpio_ext_porta[31]	I/O	PAD_IO_63	IO_63_sel == 3'd0
gpio1_gpio_ext_portb[0]	I/O	PAD_IO_64	IO_64_sel == 3'd1
gpio1_gpio_ext_portb[1]	I/O	PAD_IO_65	IO_65_sel == 3'd1
gpio1_gpio_ext_portb[2]	I/O	PAD_IO_66	IO_66_sel == 3'd1
gpio1_gpio_ext_portb[3]	I/O	PAD_IO_67	IO_67_sel == 3'd1
gpio1_gpio_ext_portb[4]	I/O	PAD_IO_68	IO_68_sel == 3'd1
gpio1_gpio_ext_portb[5]	I/O	PAD_IO_69	IO_69_sel == 3'd1
gpio1_gpio_ext_portb[6]	I/O	PAD_IO_70	IO_70_sel == 3'd1
gpio1_gpio_ext_portb[7]	I/O	PAD_IO_71	IO_71_sel == 3'd1

12.5.7 Application Notes

- Reading from an unused location or unused bits in a particular register always returns zeros. There is no error mechanism in the APB.
- Programming the GPIO registers for interrupt capability, edge-sensitive or level-sensitive interrupts, and interrupt polarity should be completed prior to enabling the interrupts on Port A in order to prevent spurious glitches on the interrupt lines to the interrupt controller.
- Writing to the interrupt clear register clears an edge-detected interrupt and has no effect on a level-sensitive interrupt. If, for example, the GPIO_PORTA_INTR configuration parameter is equal to 0, then the following all read back 0:
 - gpio_inten
 - gpio_intmask
 - gpio_int_level
 - gpio_int_polarity
 - gpio_int_status
 - gpio_raw_intstatus
 - gpio_debounce
 - gpio_ls_sync
 - gpio_porta_eoi
- When reading back registers that are no longer present due to configuration parameters settings, then 0 is read back. For example, if APB_DATA_WIDTH = 32 bits and GPIO_PWIDTH_A = 8, then the top 24 bits read back 0.

12.6 USB DRD

12.6.1 Overview

The system supports two USB 2.0 OTG module. The USB 2.0 part supports the transmission rate of 480Mbit/s specified by the USB 2.0 protocol and the UTMI interface protocol for high-speed transmission.

12.6.2 Function Description

12.6.2.1 Logical Block Diagram

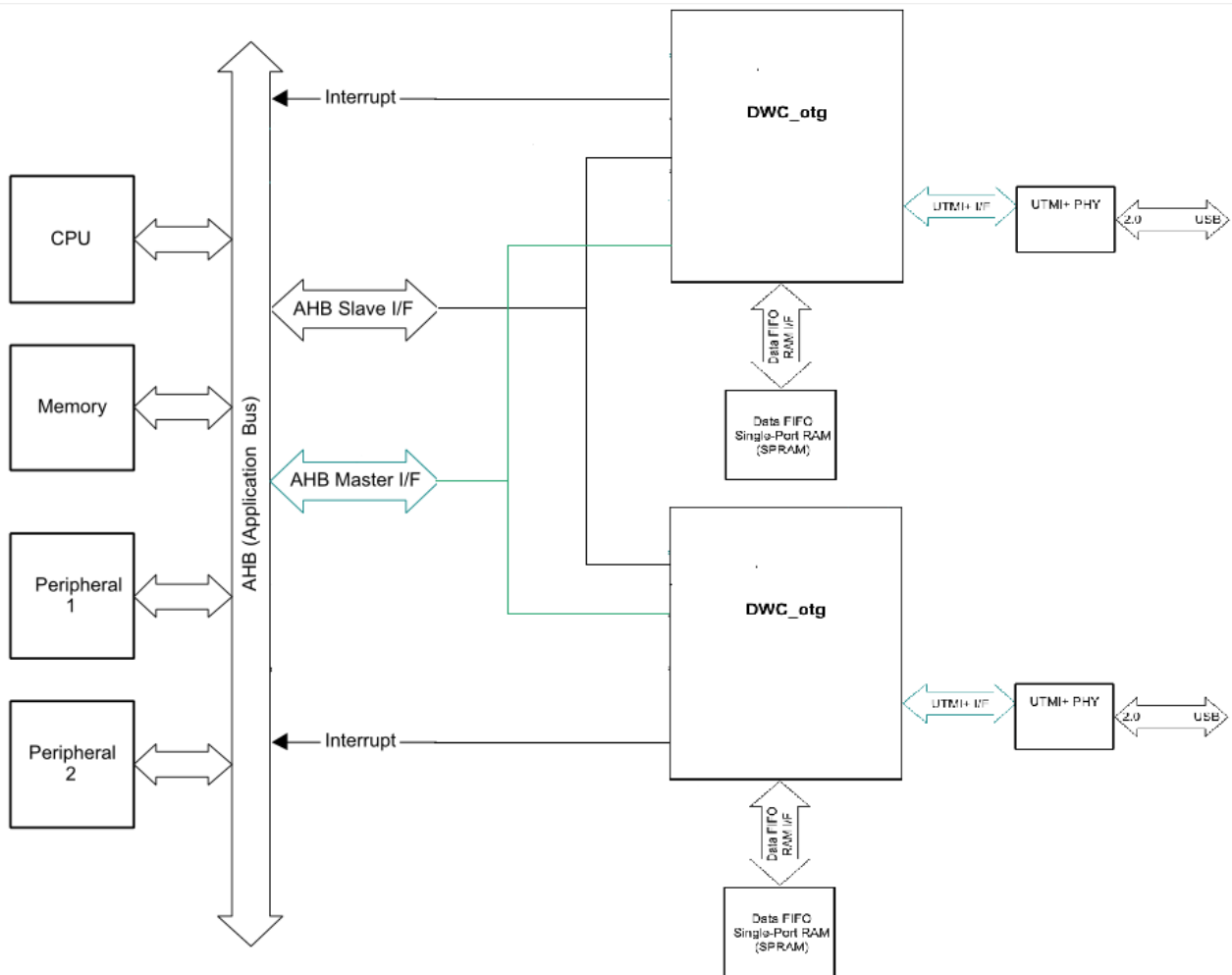


Figure 12- Logical block diagram of the USB 2.0 module

12.6.2.2 Features

The USB2.0 modules have the following features: 1.Support for the following speeds:High-Speed (HS, 480-Mbps),Full-Speed (FS, 12-Mbps) and Low-Speed (LS, 1.5-Mbps) modes 2.Multiple options available for low power operations 3.DMA mode access support on the application side 4.Supports up to 6 bidirectional endpoints, including control endpoint 0. 5.Supports the Keep-Alive in Low-Speed mode and SOFs in High/Full-Speed modes 6.The USB 2.0 modules supports the following transfer types:Control transfer,Bulk transfer,Interrupt transfer,isochronous transfer.

12.6.3 Operating Mode

12.6.3.1 Host/Device Mode Switchover

The host/device working mode can only be statically switched and cannot be dynamically switched. When a detected ID change event or VBUS change event after disconnect, it means a USB2 OTG A device or B device may connect, then check the status of ID. If ID==0, it will work as A device, then follow host programming flow; if ID==1, it will work as a B device, then it follows device programming flow.

12.6.4 Register Summary

12.6.4.1 OTG IP Register

Table describes USB registers in OTG IP.

Offset Address	Register	Description
0x0	GOTGCTL	Control and Status Register
0x4	GOTGINT	Interrupt Register
0x8	GAHBCFG	AHB Configuration Register
0xc	GUSBCFG	USB Configuration Register
0x10	GRSTCTL	Reset Register
0x14	GINTSTS	Interrupt Register
0x18	GINTMSK	Interrupt Mask Register
0x1c	GRXSTSR	Receive Status Debug Read Register
0x20	GRXSTSP	Receive Status Read/Pop Register
0x24	GRXFSIZ	Receive FIFO Size Register
0x28	GNPTXFSIZ	Non-periodic Transmit FIFO Size Register
0x2c	GNPTXSTS	Non-periodic Transmit FIFO/Queue Status Register
0x30	GI2CCTL	I2C Access Register
0x34	GPVNDCTL	PHY Vendor Control Register
0x38	GGPIO	General Purpose Input/Output Register
0x3c	GUID	User ID Register
0x40	GSNPSID	Synopsys ID Register
0x44	GHWCFG1	User Hardware Configuration 1 Register
0x48	GHWCFG2	User Hardware Configuration 2 Register
0x4c	GHWCFG3	User Hardware Configuration 3 Register
0x50	GHWCFG4	User Hardware Configuration 4 Register
0x54	GLPMCFG	LPM Config Register
0x58	GPWRDN	Global Power Down Register
0x5c	GDFIFOCFG	Global DFIFO Configuration Register
0x60	GADPCTL	ADP Timer, Control and Status Register
0x64	GREFCLK	ref_clk Control Register
0x68	GINTMSK2	Interrupt Mask Register 2
0x6c	GINTSTS2	Interrupt Register 2
0x100	HPTXFSIZ	Host Periodic Transmit FIFO Size Register
0x104 + (i-1)*004	DPTXFSIZi (for i = 1; i <= OTG_NUM_PERIO_EPS)	Device Periodic Transmit FIFO-i Size Register
0x104 + (i-1)*004	DIEPTXFi (for i = 1; i <= OTG_NUM_IN_EPS-1)	Device IN Endpoint Transmit FIFO Size Register

Offset Address	Register	Description
0x400	HCFG	Host Configuration Register
0x404	HFIR	Host Frame Interval Register
0x408	HFNUM	Host Frame Number/Frame Time Remaining Register
0x410	HPTXSTS	Host Periodic Transmit FIFO/Queue Status Register
0x414	HAINT	Host All Channels Interrupt Register
0x418	HAINTMSK	Host All Channels Interrupt Mask Register
0x41c	HFLBAddr	Host Frame List Base Address Register
0x440	HPRT	Host Port Control and Status Register
0x500 +i*20	HCCHARi (for i = 0; i <= OTG_NUM_HOST_CHAN-1)	Host Channel Characteristics Register
0x504 +i*20	HCSPLTi (for i = 0; i <= OTG_NUM_HOST_CHAN-1)	Host Channel Split Control Register
0x508 +i*20	HCINTi (for i = 0; i <= OTG_NUM_HOST_CHAN-1)	Host Channel Interrupt Register
0x50C +i*20	HCINTMSKi (for i = 0; i <= OTG_NUM_HOST_CHAN-1)	Host Channel Interrupt Mask Register
0x510 +i*20	HCTSIZi (for i = 0; i <= OTG_NUM_HOST_CHAN-1)	Host Channel Transfer Size Register
0x514 +i*20	HCDMAi (for i = 0; i <= OTG_NUM_HOST_CHAN-1)	Host Channel DMA Address Register
0x51C +i*20	HCDMABi (for i = 0; i <= OTG_NUM_HOST_CHAN-1)	Host Channel DMA Buffer Address Register
0x800	DCFG	Device Configuration Register
0x804	DCTL	Device Control Register
0x808	DSTS	Device Status Register

Offset Address	Register	Description
0x810	DIEPMSK	Device IN Endpoint Common Interrupt Mask Register
0x814	DOEPMSK	Device OUT Endpoint Common Interrupt Mask Register
0x818	DAINT	Device All Endpoints Interrupt Register
0x81c	DAINTMSK	Device All Endpoints Interrupt Mask Register
0x820	DTKNQR1	Device IN Token Sequence Learning Queue Read Register 1
0x824	DTKNQR2	Device IN Token Sequence Learning Queue Read Register 2
0x828	DVBUSDIS	Device VBUS Discharge Time Register
0x82c	DVBUSPULSE	Device VBUS Pulsing Time Register
0x830	DTKNQR3	Device IN Token Sequence Learning Queue Read Register 3
0x830	DTHRCTL	Device Threshold Control Register
0x834	DTKNQR4	Device IN Token Sequence Learning Queue Read Register 4
0x834	DIEPEMPMSK	Device IN Endpoint FIFO Empty Interrupt Mask Register
0x838	DEACHINT	Device Each Endpoints Interrupt Register
0x83c	DEACHINTMSK	Device Each Endpoints Interrupt Mask Register
0x840	DIEPEACHMSK0	Device Each IN Endpoint 0 Interrupt Register
0x840 +i*004	DIEPEACHMSKi (for i = 1; i <= OTG_NUM_EPS)	Device Each IN Endpoint Interrupt Register
0x880	DOEPEACHMSK0	Device Each OUT Endpoint 0 Interrupt Register
0x880 +i*004	DOEPEACHMSKi (for i = 1; i <= OTG_NUM_EPS)	Device Each OUT Endpoint Interrupt Register
0x900	DIEPCTL0	Device Control IN Endpoint 0 Control Register
0x908	DIEPINT0	Device IN Endpoint 0 Interrupt Register
0x910	DIEPTSIZE0	Device IN Endpoint 0 Transfer Size Register
0x914	DIEPDMA0	Device IN Endpoint 0 DMA Address Register
0x918	DTXFSTS0	Device IN Endpoint Transmit FIFO Status Register 0
0x91c	DIEPDMA0	Device IN Endpoint 0 Buffer Address Register
0x900 +i*20	DIEPCTLi (for i = 1; i <= OTG_NUM_EPS)	Device Control IN Endpoint Control Register
0x908 +i*20	DIEPINTi (for i = 1; i <= OTG_NUM_EPS)	Device IN Endpoint Interrupt Register
0x910 +i*20	DIEPTSIZEi (for i = 1; i <= OTG_NUM_EPS)	Device IN Endpoint Transfer Size Register

Offset Address	Register	Description
0x914 +i*20	DIEPDMAi (for i = 1; i <= OTG_NUM_EPS)	Device IN Endpoint DMA Address Register
0x918 +i*20	DTXFSTSi (for i = 1; i <= OTG_NUM_EPS)	Device IN Endpoint Transmit FIFO Status Register
0x91C +i*20	DIEPDMABi (for i = 1; i <= OTG_NUM_EPS)	Device IN Endpoint Buffer Address Register
0xb00	DOEPCTL0	Device Control OUT Endpoint 0 Control Register
0xb08	DOEPINT0	Device OUT Endpoint 0 Interrupt Register
0xb10	DOEPTSIZE0	Device OUT Endpoint 0 Transfer Size Register
0xb14	DOEPDMA0	Device OUT Endpoint 0 DMA Address Register
0xb1c	DOEPDMAB0	Device OUT Endpoint 0 DMA Buffer Address Register
0xB00 +i*20	DOEPCTLi (for i = 1; i <= OTG_NUM_EPS)	Device Control OUT Endpoint Control Register
0xB08 +i*20	DOEPINTi (for i = 1; i <= OTG_NUM_EPS)	Device OUT Endpoint Interrupt Register
0xB10 +i*20	DOEPTSIZEi (for i = 1; i <= OTG_NUM_EPS)	Device OUT Endpoint Transfer Size Register
0xB14 +i*20	DOEPDMAi (for i = 1; i <= OTG_NUM_EPS)	Device OUT Endpoint DMA Address Register
0xB1C +i*20	DOEPDMABi (for i = 1; i <= OTG_NUM_EPS)	Device OUT Endpoint Buffer Address Register
0xe00	PCGCCTL	Power and Clock Gating Control Register
0xe04	PCGCCTL1	Power and Clock Gating Control Register1

12.6.4.2 USB Ctrl Register in SoC

Table describes USB Ctrl registers in SoC.

Offset Address	Register	Description
0x10	OTG0_PROBE0	otg0_internal_probes[31:0]
0x14	OTG0_PROBE1	otg0_internal_probes[61:32]
0x18	OTG0_PROBEP0	otg0_internal_probes_p[31:0]
0x1c	OTG0_PROBEP1	otg0_internal_probes_p[63:32]
0x20	OTG0_PROBEP2	otg0_internal_probes_p[95:64]
0x24	OTG0_PROBEP3	otg0_internal_probes_p[127:96]
0x28	OTG0_PROBEP4	otg0_internal_probes_p[159:128]
0x2c	OTG0_PROBEP5	otg0_internal_probes_p[191:160]
0x30	OTG0_PROBEP6	otg0_internal_probes_p[218:192]

Offset Address	Register	Description
0x34	OTG0_STRAP	select CLKCORE as ref clk
0x40	OTG1_PROBE0	otg0_internal_probes[31:0]
0x44	OTG1_PROBE1	otg0_internal_probes[61:32]
0x48	OTG1_PROBEP0	otg0_internal_probes_p[31:0]
0x4c	OTG1_PROBEP1	otg0_internal_probes_p[63:32]
0x50	OTG1_PROBEP2	otg0_internal_probes_p[95:64]
0x54	OTG1_PROBEP3	otg0_internal_probes_p[127:96]
0x58	OTG1_PROBEP4	otg0_internal_probes_p[159:128]
0x5c	OTG1_PROBEP5	otg0_internal_probes_p[191:160]
0x60	OTG1_PROBEP6	otg0_internal_probes_p[218:192]
0x64	OTG1_STRAP	select CLKCORE as ref clk
0x78	USB0_TEST_CTL2	otg0 phy port ctrl
0x7c	USB0_TEST_CTL3	otg0 phy port ctrl
0x80	USB0_TEST_CTL4	otg0 phy port ctrl
0x84	USB0_TEST_CTL5	otg0 phy port ctrl
0x88	USB0_TEST_CTL6	otg0 phy port ctrl
0x98	USB1_TEST_CTL2	otg1 phy port ctrl
0xa0	USB1_TEST_CTL4	otg1 phy port ctrl
0xa4	USB1_TEST_CTL5	otg1 phy port ctrl
0xa8	USB1_TEST_CTL6	otg1 phy port ctrl
0xb0	USB0_CTL0	otg0 phy port ctrl
0xb4	USB0_CTL1	otg1 phy port ctrl
0xb8	USB1_CTL0	otg0 phy port ctrl
0xbc	USB1_CTL1	otg1 phy port ctrl

12.6.5 USB Register Description

12.6.5.1 OTG IP Register

GOTGCTL

GOTGCTL is OTG Control and Status register controls the behavior and reflects the status of the OTG function of the controller.

Offset:0x0 Total Reset Value:0x000d_0000

Bits	Access	Name	Description	Reset

Bits	Access	Name	Description	Reset
31	R/W	Testmode_corr_eUSB2	UTMI IF correction for eUSB2 PHY during Test mode This bit is used to modify the behavior of UTMI 8-bit interface signals during test J and test K sequences when eUSB2 PHY is used. When this bit is set to 1'b1, the controller asserts utmi_txvalid and utmi_opmode in the same cycle during test J or test K sequence execution. 0x0 (eUSB2_corr_disable): The controller asserts utmi_txvalid one cycle later than utmi_opmode. 0x1 (eUSB2_corr_enable): The controller asserts utmi_txvalid and utmi_opmode in the same cycle.	0x0
30:28	R	Reserved_30_28	RESERVED	0x0
27	R/W	ChirpEn	RESERVED	0x0
26:20	R	MultValIdBC	RESERVED	0x0
21	R	CurMod	Current Mode of Operation (CurMod) Mode: Host and Device Indicates the current mode. 1'b0: Device mode 1'b1: Host mode	0x0
20	R/W	OTGVer	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
19	R	BSesVld	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.	0x1
18	R	ASesVld	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	0x1
17	R	DbncTime	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 micro-sec) 1'b1: Short debounce time, used for soft connections (2.5 micro- sec)	0x0

Bits	Access	Name	Description	Reset
16	R	ConIDSts	Mode: Host and Device Connector ID Status (ConIDSts) Indicates the connector ID status on a connect event. 1'b0: The core is in A-Device mode. 1'b1: The core is in B-Device mode.	0x1
15	R/W	DbnceFltrBypass	Mode: Host and Device Debounce Filter Bypass Bypass Debounce filters for avalid, bvalid, vbusvalid, sessend, iddig signals when enabled. 1'b0: Disabled 1'b1: Enabled	0x0
14:13			Reserved Field: Yes	
12	R/W	EHEn	RESERVED	0x0
11	R/W	DevHNPEn	RESERVED	0x0
10	R/W	HstSetHNPEn	RESERVED	0x0
9	R/W	HNPREq	RESERVED	0x0
8	R	HstNegScs	RESERVED	0x0
7	R/W	BvalidOvVal	Mode: Device only B-Peripheral Session Valid OverrideValue (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
6	R/W	BvalidOvEn	Mode: Device only B-Peripheral Session Valid Override Value (BvalidOvEn) This bit is used to enable/disable the software to override the Bvalid signal using the GOTGCTL.BvalidOvVal. 1'b1 : Internally Bvalid received from the PHY is overridden with GOTGCTL.BvalidOvVal. 1'b0 : Override is disabled and bvalid signal from the respective PHY selected is used internally by the force	0x0
5	R/W	AvalidOvVal	Mode: Host only A-Peripheral Session Valid OverrideValue (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 GOTGCTL.AvalidOvEn =1 1'b1 : Avalid value is 1'b1 when GOTGCTL.AvalidOvEn =1	0x0
4	R/W	AvalidOvEn	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 GOTGCTL.AvalidOvVal. 1'b0: Override is disabled and avalid signal from the respective PHY selected is used internally by the core	0x0

Bits	Access	Name	Description	Reset
3	R/W	VbvalidOvVal	Mode: Host only VBUS Valid OverrideValue (VbvalidOvVal) This bit is used to set Override value for vbusvalid signal when GOTGCTL.VbvalidOvEn is set. 1'b0 : vbusvalid value is 1'b0 when GOTGCTL.VbvalidOvEn =1 1'b1 : vbusvalid value is 1'b1 when GOTGCTL.VbvalidOvEn =1	0x0
2	R/W	VbvalidOvEn	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 controller.	0x0
1	R/W	SesReq	RESERVED	0x0
0	R	SesReqScs	RESERVED	0x0

GOTGINT

GOTGINT is application reads this register whenever there is an OTG interrupt and clears the bits in this register to clear the OTG interrupt.

Offset:0x4 Total Reset Value:0x0000_0000 RESERVED

GAHBCFG

GAHBCFG can be used to configure the core after power-on or a change in mode. This register mainly contains AHB system-related configuration parameters. Do not change this register after the initial programming. The application must program this register before starting any transactions on either the AHB or the USB.

Offset:0x8 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:25	R	RESERVED1	RESERVED	0x0
24	R/W	InvDescEndianness	Mode: Host and Device Invert Descriptor Endianness (InvDescEndianness) 1'b0: Descriptor Endianness is same as 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

Bits	Access	Name	Description	Reset
23	R/W	AHBSingle	Mode: Host and Device AHB Single Support (AHBSingle) This bit when programmed supports Single transfers for the remaining data in a transfer when the core is operating in DMA mode. 1'b0: The remaining data in the transfer is sent using INCR burst size. 1'b1: The remaining data in the transfer is sent using Single burst size.	0x0
22	R/W	NotiAllDmaWrit	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. 1: The 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. 0: The 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
21	R/W	RemMemSupp	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. 0x0 (DISABLED): Remote Memory Support Feature disabled 0x1 (ENABLED): Remote Memory Support Feature enabled	0x0
20:9			Reserved Field: Yes	
8	R/W	PTxFEmpLvl	Mode: Host only Periodic Tx FIFO Empty Level (PTxFEmpLvl) Indicates when the Periodic Tx FIFO 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 Tx FIFO is half empty 1'b1: GINTSTS.PTxFEmp interrupt indicates that the Periodic Tx FIFO is completely empty	0x0

Bits	Access	Name	Description	Reset
7	R/W	NPTxFEmpLvl	Mode: Host and device Non-Periodic TxFIFO Empty Level (NPTxFEmpLvl) 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
6	R	RESERVED	RESERVED	0x0
5	R/W	DMAEn	Mode: Host and device DMA Enable (DMAEn) 0x0 (SLAVEMODE): Core operates in Slave mode 0x1 (DMAMODE): Core operates in a DMA mode	0x0
4:1	R/W	HBstLen	Mode: Host and device Burst Length/Type (HBstLen) 4'b0000 Single 4'b0001 INCR 4'b0011 INCR4 4'b0101 INCR8 4'b0111 INCR16 Others: Reserved	0x0
0	R/W	GlbIntrMsk	Mode: Host and device Global Interrupt Mask (GlbIntrMsk) 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 controller. 0x0 (MASK): Mask the interrupt assertion to the application 0x1 (NOMASK): Unmask the interrupt assertion to the application.	0x0

GUSBCFG

GUSBCFG can be used to configure the core after power-on or when changing to Host mode or Device mode. It contains USB and USB-PHY related configuration parameters. The application must program this register before starting any transactions on either the AHB or the USB. If you are using the HSIC interface, HSIC PHY must be in reset while programming this register. Do not make changes to this register after the initial programming.

Offset:0xc Total Reset Value:0x1000_2800

Bits	Access	Name	Description	Reset

Bits	Access	Name	Description	Reset
31	W	CorruptTxPkt	Mode: Host and device Corrupt Tx packet (CorruptTxPkt) This bit is for debug purposes only. Never Set this bit to 1. The application must always write 1'b0 to this bit. 0x0 (NODEBUG): Normal Mode 0x1 (DEBUG): Debug Mode	0x0
30	R/W	ForceDevMode	Mode:Host and device Force Device Mode (ForceDevMode) Writing a 1 to this bit forces the controller to device mode irrespective of utmiotg_iddig input pin. 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 micro sec is sufficient. This bit is valid only when OTG_MODE = 0, 1 or 2. In all other cases, this bit reads 0. 0x0 (DISABLED): Normal Mode 0x1 (ENABLED): Force Device Mode	0x0
29	R/W	ForceHstMode	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. 0x0 (DISABLED): Normal Mode 0x1 (ENABLED): Force Host Mode	0x0
28	R/W	TxEndDelay	Mode: Device only Tx End Delay (TxEndDelay) The default value of this field is 1'b1 and it is not recommended to program it to 1'b0. The option to set it to 1'b0 (Normal Mode) is present only for debug purpose. 0x0 (DISABLED): Normal Mode 0x1 (ENABLED): Tx End delay	0x1
27:23		RESERVED	RESERVED	0x0
22	R/W	TermSelDLPulse	Mode: Device only TermSel DLine Pulsing Selection (TermSelDLPulse) This bit selects utmi_termselect to drive data line pulse during SRP. 0x0 (TXVALID): Data line pulsing using utmi_txvalid 0x1 (TERMSEL): Data line pulsing using utmi_termsel	0x0
21:16		RESERVED	RESERVED	0x0

Bits	Access	Name	Description	Reset
15	R/W	PhyLPwrClkSel	PHY Low-Power Clock Select (PhyLPwrClkSel) Mode: Host and Device 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. 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 (INTPLLCLK): 480-MHz Internal PLL clock 0x1 (EXTCLK): 48-MHz External Clock	0x0
14	R	Reserved_14	RESERVED	0x0
13:10	R/W	USBTrdTim	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 0x9 (TURNTIME8BIT): MAC interface is 8-bit UTMI+. 0x5 (TURNTIME16BIT): MAC interface is 16-bit UTMI+.	0x5
9:7		RESERVED	RESERVED	0x0
6	R	PHYSel	PHYSel 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. 0x0 (USB20): USB 2.0 high-speed UTMI+ or ULPI PHY is selected 0x1 (USB11): USB 1.1 full-speed serial transceiver is selected	0x0
5	R	FSIntf	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. 0x0 (FS6PIN): 6-pin unidirectional full-speed serial interface 0x1 (FS3PIN): 3-pin bidirectional full-speed serial interface	0x0
4	R	ULPI_UTMI_Sel	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. 0x0 (UTMI): UTMI+ Interface 0x1 (ULPI): ULPI Interface	0x0

Bits	Access	Name	Description	Reset
3	R/W	PHYIf	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. 0x0 (BITS8): PHY 8bit Mode 0x1 (BITS16): PHY 16bit Mode	0x0
2:0	R/W	TOutCal	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 full-speed 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 as follows: 0x0 (ZERO): Add 0 PHY clocks 0x1 (ONE): Add 1 PHY clocks 0x2 (TWO): Add 2 PHY clocks 0x3 (THREE): Add 3 PHY clocks 0x4 (FOUR): Add 4 PHY clocks 0x5 (FIVE): Add 5 PHY clocks 0x6 (SIX): Add 6 PHY clocks 0x7 (SEVEN): Add 7 PHY clocks	0x0

GRSTCTL

The application uses this register to reset various hardware features inside the controller.

Offset:0x10 Total Reset Value:0x8000_0000

Bits	Access	Name	Description	Reset
31	R	AHBIdle	Mode: Host and Device AHB Master Idle (AHBIdle) Indicates that the AHB Master State Machine is in the IDLE condition. 0x0 (INACTIVE): Not Idle 0x1 (ACTIVE): AHB Master Idle	0x1
30	R	DMAReq	Mode: Host and Device DMA Request Signal (DMAReq) Indicates that the DMA request is in progress. Used for debug. 0x0 (INACTIVE): No DMA request 0x1 (ACTIVE): DMA request is in progress	0x0

Bits	Access	Name	Description	Reset
29	R/W1C	CSftRstDone	Mode: Host and Device Core Soft Reset Done (CSftRstDone) The core sets this bit when all the necessary logic is reset in the core. This bit is cleared by the application along with GRSTCTL.CSftRst (bit 0) 0x0 (INACTIVE): No reset 0x1 (ACTIVE): Core Soft Reset is done	0x0
28:11	R	RESERVED	RESERVED	0x0
10:6	R/W	TxFNum	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. 0x0 (TXF0): -Periodic TxFIFO flush in host mode -Periodic TxFIFO 0 flush in device mode when in shared FIFO operation - TXFIFO 0 flush in device mode when in dedicated FIFO mode 0x1 (TXF1): -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 0x2 (TXF2): -Periodic TxFIFO 2 flush in device mode when in shared FIFO operation -TXFIFO 2 flush in device mode when in dedicated FIFO mode 0x3 (TXF3): -Periodic TxFIFO 3 flush in device mode when in shared FIFO operation -TXFIFO 3 flush in device mode when in dedicated FIFO mode 0x4 (TXF4): -Periodic TxFIFO 4 flush in device mode when in shared FIFO operation -TXFIFO 4 flush in device mode when in dedicated FIFO mode 0x5 (TXF5): -Periodic TxFIFO 5 flush in device mode when in shared FIFO operation -TXFIFO 5 flush in device mode when in dedicated FIFO mode 0x6 (TXF6): -Periodic TxFIFO 6 flush in device mode when in shared FIFO operation -TXFIFO 6 flush in device mode when in dedicated FIFO mode 0x7 (TXF7): -Periodic TxFIFO 7 flush in device mode when in shared FIFO operation -TXFIFO 7 flush in device mode when in dedicated FIFO mode 0x8 (TXF8): -Periodic TxFIFO 8 flush in device mode when in shared FIFO operation -TXFIFO 8 flush in device mode when in dedicated FIFO mode 0x9 (TXF9): -Periodic TxFIFO 9 flush in device mode when in shared FIFO operation -TXFIFO 9 flush in device mode when in dedicated FIFO mode 0xa (TXF10): -Periodic TxFIFO 10 flush in device mode when in shared FIFO operation -TXFIFO 10 flush in device mode when in dedicated FIFO mode 0xb (TXF11): -Periodic TxFIFO 11 flush in device mode when in shared FIFO operation -TXFIFO 11 flush in device mode when in dedicated	0x0

Bits	Access	Name	Description	Reset
			FIFO mode 0xc (TXF12): -Periodic TxFIFO 12 flush in device mode when in shared FIFO operation -TXFIFO 12 flush in device mode when in dedicated FIFO mode 0xd (TXF13): -Periodic TxFIFO 13 flush in Device mode when in shared FIFO operation -TXFIFO 13 flush in device mode when in dedicated FIFO mode 0xe (TXF14): -Periodic TxFIFO 14 flush in Device mode when in shared FIFO operation -TXFIFO 14 flush in device mode when in dedicated FIFO mode 0xf (TXF15): -Periodic TxFIFO 15 flush in Device mode when in shared FIFO operation - TXFIFO 15 flush in device mode when in dedicated FIFO mode 0x10 (TXF16): Flush all the transmit FIFOs in device or host mode	
5	R/W1S	TxFFlsh	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. 0x0 (INACTIVE): No Flush 0x1 (ACTIVE): Selectively flushes a single or all transmit FIFOs	0x0
4	R/W1S	RxFFlsh	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. 0x0 (INACTIVE): Does not flush the entire RxFIFO 0x1 (ACTIVE): Flushes the entire RxFIFO	0x0
3	R/W1S	RESERVED	RESERVED	0x0
2	R/W1S	FrmCntrRst	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. 0x0 (NOTACTIVE): No reset 0x1 (ACTIVE): Host Frame Counter Reset	0x0
1	R/W1S	PIUFSSftRst	Mode: Host and Device PIU FS Dedicated Controller Soft Reset (PIUFSSftRst) Resets the PIU FS Dedicated Controller 0x0 (RESET_INACTIVE): No Reset 0x1 (RESET_ACTIVE): PIU FS Dedicated Controller Soft Reset	0x0
0	R/W	CSftRst	Mode: Host and Device Core Soft Reset (CSftRst) Resets the hclk and phy_clock domains as follows: 0x0 (NOTACTIVE): No reset 0x1 (ACTIVE): Resets hclk and phy_clock domains	0x0

GINTSTS

This register interrupts the application for system-level events in the current mode (Device mode or Host mode).

Offset:0x14 Total Reset Value:0x0400_0020

Bits	Access	Name	Description	Reset
31	R/W1C	WkUpInt	Mode: Host and Device Resume/Remote Wakeup Detected Interrupt (WkUpInt) Wakeup Interrupt during Suspend(L2) or LPM(L1) state. 0x0 (INACTIVE): Not active 0x1 (ACTIVE): Resume or Remote Wakeup Detected Interrupt	0x0
30	R/W1C	SessReqInt	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 Host mode, this interrupt is asserted 0x0 (INACTIVE): Not active 0x1 (ACTIVE): Session Request New Session Detected Interrupt	0x0
29	R/W1C	DisconnInt	Mode: Host only Disconnect Detected Interrupt (DisconnInt) Asserted when a device disconnect is detected. 0x0 (INACTIVE): Not active 0x1 (ACTIVE): Disconnect Detected Interrupt	0x0
28	R/W1C	ConIDStsChng	Mode: Host and Device Connector ID Status Change (ConIDStsChng) The core sets this bit when there is a change in connector ID status. 0x0 (INACTIVE): Not Active 0x1 (ACTIVE): Connector ID Status Change	0x0
27	R/W1C	LPM_Int	Mode: Host and Device LPM Transaction Received Interrupt (LPM_Int). 0x0 (INACTIVE): Not Active 0x1 (ACTIVE): LPM Transaction Received Interrupt	0x0
26	R	PTxFEmp	Mode: Host only Periodic Tx FIFO Empty (PTxFEmp) 0x0 (INACTIVE): Not active 0x1 (ACTIVE): Periodic Tx FIFO Empty	0x1
25	R	HChInt	Mode: Host only Host Channels Interrupt (HChInt) 0x0 (INACTIVE): Not active 0x1 (ACTIVE): Host Channels Interrupt	0x0
24	R	PrtInt	Mode: Host only Host Port Interrupt (PrtInt) 0x0 (INACTIVE): Not active 0x1 (ACTIVE): Host Port Interrupt	0x0
23	R/W1C	ResetDet	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. 0x0 (INACTIVE): Not active 0x1 (ACTIVE): Reset detected Interrupt	0x0

Bits	Access	Name	Description	Reset
22	R/W1C	FetSusp	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. 0x0 (INACTIVE): Not active 0x1 (ACTIVE): Data Fetch Suspended	0x0
21	R/W1C	incomplP	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. 0x0 (INACTIVE): Not active 0x1 (ACTIVE): Incomplete Periodic Transfer	0x0
20	R/W1C	incomplSOIN	Mode: Device only Incomplete Isochronous IN Transfer (incomplSOIN) 0x0 (INACTIVE): Not active 0x1 (ACTIVE): Incomplete Isochronous IN Transfer	0x0
19	R	OEPInt	Mode: Device only OUT Endpoints Interrupt (OEPInt) 0x0 (INACTIVE): Not active 0x1 (ACTIVE): OUT Endpoints Interrupt	0x0
18	R	IEPInt	Mode: Device only IN Endpoints Interrupt (IEPInt) 0x0 (INACTIVE): Not active 0x1 (ACTIVE): IN Endpoints Interrupt	0x0
17	R/W1C	EPMis	Mode: Device only Endpoint Mismatch Interrupt (EPMis) 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 0x0 (INACTIVE): Not active 0x1 (ACTIVE): Endpoint Mismatch Interrupt	0x0
16		RESERVED	RESERVED	0x0
15	R/W1C	EOPF	Mode: Device only End of Periodic Frame Interrupt (EOPF) 0x0 (INACTIVE): Not active 0x1 (ACTIVE): End of Periodic Frame Interrupt	0x0
14	R/W1C	ISOOutDrop	Mode: Device only Isochronous OUT Packet Dropped Interrupt (ISOOutDrop) 0x0 (INACTIVE): Not active 0x1 (ACTIVE): Isochronous OUT Packet Dropped Interrupt	0x0

Bits	Access	Name	Description	Reset
13	R/W1C	EnumDone	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. 0x0 (INACTIVE): Not active 0x1 (ACTIVE): Enumeration Done	0x0
12	R/W1C	USBRst	Mode: Device only USB Reset (USBRst) The controller sets this bit to indicate that a reset is detected on the USB. 0x0 (INACTIVE): Not active 0x1 (ACTIVE): USB Reset	0x0
11	R/W1C	USBSusp	Mode: Device only USB Suspend (USBSusp) The controller sets this bit to indicate that a suspend was detected on the USB. The controller enters the Suspended state when there is no activity on the linestate signal for an extended period of time. 0x0 (INACTIVE): Not Active 0x1 (ACTIVE): USB Suspend	0x0
10	R/W1C	ErlySusp	Mode: Device only Early Suspend (ErlySusp) The controller sets this bit to indicate that an Idle state has been detected on the USB for 3 ms. 0x0 (INACTIVE): No Idle state detected 0x1 (ACTIVE): 3ms of Idle state detected	0x0
9		RESERVED	RESERVED	0x0
8		RESERVED	RESERVED	0x0
7	R	GOUTNakEff	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). 0x0 (INACTIVE): Not Active 0x1 (ACTIVE): Global OUT NAK Effective	0x0
6	R	GINNakEff	Mode: Device only Global IN Non-periodic NAK Effective (GINNakEff) 0x0 (INACTIVE): Global Non-periodic IN NAK not active 0x1 (ACTIVE): Set Global Non-periodic IN NAK bit	0x0
5	R	NPTxFEmp	Mode: Host and Device Non-periodic Tx FIFO Empty (NPTxFEmp) 0x0 (INACTIVE): Non-periodic Tx FIFO is not empty 0x1 (ACTIVE): Non-periodic Tx FIFO is empty	0x1

Bits	Access	Name	Description	Reset
4	R	RxFLvl	Mode: Host and Device RxFIFO Non-Empty (RxFLvl) Indicates that there is at least one packet pending to be read from the RxFIFO. 0x0 (INACTIVE): Rx Fifo is empty 0x1 (ACTIVE): Rx Fifo is not empty	0x0
3	R/W1C	Sof	Mode: Host and Device Start of (micro)Frame (Sof) 0x0 (INACTIVE): No Start of Frame 0x1 (ACTIVE): Start of Frame	0x0
2	R	OTGInt	Mode: Host and Device OTG Interrupt (OTGInt) The controller 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. 0x0 (INACTIVE): No Interrupt 0x1 (ACTIVE): OTG Interrupt	0x0
1	R/W1C	ModeMis	Mode: Host and Device Mode Mismatch Interrupt (ModeMis) 0x0 (INACTIVE): No Mode Mismatch Interrupt 0x1 (ACTIVE): Mode Mismatch Interrupt	0x0
0	R	CurMod	Mode: Host and Device Current Mode of Operation (CurMod) Indicates the current mode. 0x0 (DEVICE): Device mode 0x1 (HOST): Host mode	0x0

GINTMSK

This register works with the Interrupt Register (GINTSTS) to interrupt the application. When an interrupt bit is masked, the interrupt associated with that bit is not generated. However, the GINTSTS register bit corresponding to that interrupt is still set.

Offset:0x18 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31	R/W	WkUpIntMsk	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 (MASK): Resume or Remote Wakeup Detected Interrupt Mask 0x1 (NOMASK): Unmask Resume Remote Wakeup Detected	0x0
30	R/W	SessReqIntMsk	Mode: Host and Device Session Request/New Session Detected Interrupt Mask (SessReqIntMsk) 0x0 (MASK): Session Request or New Session Detected Interrupt Mask 0x1 (NOMASK): No Session Request or New Session Detected	0x0

Bits	Access	Name	Description	Reset
29	R/W	DisconnIntMsk	Mode: Host and Device Disconnect Detected Interrupt Mask (DisconnIntMsk) 0x0 (MASK): Disconnect Detected Interrupt Mask 0x1 (NOMASK): No Disconnect Detected Interrupt Mask	0x0
28	R/W	ConIDStsChngMsk	Mode: Host and Device Connector ID Status Change Mask (ConIDStsChngMsk) 0x0 (MASK): Connector ID Status Change Mask 0x1 (NOMASK): No Connector ID Status Change Mask	0x0
27	R/W	LPM_IntMsk	Mode: Host and Device LPM Transaction Received Interrupt (LPM_Int) LPM Transaction received interrupt Mask 0x0 (MASK): LPM Transaction received interrupt Mask 0x1 (NOMASK): No LPM Transaction received interrupt Mask	0x0
26	R/W	PTxFEmpMsk	Mode: Host only Periodic Tx FIFO Empty Mask (PTxFEmpMsk) 0x0 (MASK): Periodic Tx FIFO Empty Mask 0x1 (NOMASK): No Periodic Tx FIFO Empty Mask	0x0
25	R/W	HChIntMsk	Mode: Host only Host Channels Interrupt Mask (HChIntMsk) 0x0 (MASK): Host Channels Interrupt Mask 0x1 (NOMASK): No Host Channels Interrupt Mask	0x0
24	R/W	PrtIntMsk	Mode: Host only Host Port Interrupt Mask (PrtIntMsk) 0x0 (MASK): Host Port Interrupt Mask 0x1 (NOMASK): No Host Port Interrupt Mask	0x0
23	R/W	ResetDetMsk	Mode: Device only Reset detected Interrupt Mask (ResetDetMsk) 0x0 (MASK): Reset detected Interrupt Mask 0x1 (NOMASK): No Reset detected Interrupt Mask	0x0
22	R/W	FetSuspMsk	Mode: Device only Data Fetch Suspended Mask (FetSuspMsk) 0x0 (MASK): Data Fetch Suspended Mask 0x1 (NOMASK): No Data Fetch Suspended Mask	0x0
21	R/W	incomplPMsk	Incomplete Periodic Transfer Mask (incomplPMsk) Mode: Host only Incomplete Isochronous OUT Transfer Interrupt Mask (incomplSOOUTMsk) Mode: Device only 0x0 (MASK): Host mode: Incomplete Periodic Transfer Mask Device mode: Incomplete Isochronous OUT Transfer Mask 0x1 (NOMASK): Host mode: No Incomplete Periodic Transfer Mask Device mode: No Incomplete Isochronous OUT Transfer Mask	0x0

Bits	Access	Name	Description	Reset
20	R	RESERVED	RESERVED	0x0
19	R/W	OEPIntMsk	Mode: Device only OUT Endpoints Interrupt Mask (OEPIntMsk) 0x0 (MASK): OUT Endpoints Interrupt Mask 0x1 (NOMASK): No OUT Endpoints Interrupt Mask	0x0
18	R/W	IEPIntMsk	Mode: Device only IN Endpoints Interrupt Mask (IEPIntMsk) 0x0 (MASK): IN Endpoints Interrupt Mask 0x1 (NOMASK): No IN Endpoints Interrupt Mask	0x0
17	R/W	EPMisMsk	Mode: Device only Endpoint Mismatch Interrupt Mask (EPMisMsk) 0x0 (MASK): Endpoint Mismatch Interrupt Mask 0x1 (NOMASK): No Endpoint Mismatch Interrupt Mask	0x0
16	R/W	RESERVED	RESERVED	0x0
15	R/W	EOPFMsk	Mode: Device only End of Periodic Frame Interrupt Mask (EOPFMsk) 0x0 (MASK): End of Periodic Frame Interrupt Mask 0x1 (NOMASK): No End of Periodic Frame Interrupt Mask	0x0
14	R/W	ISOOutDropMsk	Mode: Device only Isochronous OUT Packet Dropped Interrupt Mask (ISOOutDropMsk) 0x0 (MASK): Isochronous OUT Packet Dropped Interrupt Mask 0x1 (NOMASK): No Isochronous OUT Packet Dropped Interrupt Mask	0x0
13	R/W	EnumDoneMsk	Mode: Device only Enumeration Done Mask (EnumDoneMsk) 0x0 (MASK): Enumeration Done Mask 0x1 (NOMASK): No Enumeration Done Mask	0x0
12	R/W	USBRstMsk	Mode: Device only USB Reset Mask (USBRstMsk) 0x0 (MASK): USB Reset Mask 0x1 (NOMASK): No USB Reset Mask	0x0
11	R/W	USBSuspMsk	Mode: Device only USB Suspend Mask (USBSuspMsk) 0x0 (MASK): USB Suspend Mask 0x1 (NOMASK): No USB Suspend Mask	0x0
10	R/W	ErlySuspMsk	Mode: Device only Early Suspend Mask (ErlySuspMsk) 0x0 (MASK): Early Suspend Mask 0x1 (NOMASK): No Early Suspend Mask	0x0
9	R	RESERVED	RESERVED	0x0
8	R	RESERVED	RESERVED	0x0

Bits	Access	Name	Description	Reset
7	R/W	GOUTNakEffMsk	Mode: Device only Global OUT NAK Effective Mask (GOUTNakEffMsk) 0x0 (MASK): Global OUT NAK Effective Mask 0x1 (NOMASK): No Global OUT NAK Effective Mask	0x0
6	R/W	GINNakEffMsk	Mode: Device only, Global Non-periodic IN NAK Effective Mask (GINNakEffMsk) 0x0 (MASK): Global Non-periodic IN NAK Effective Mask 0x1 (NOMASK): No Global Non-periodic IN NAK Effective Mask	0x0
5	R	RESERVED	RESERVED	0x0
4	R/W	RxFLvIMsk	Mode: Host and Device Receive FIFO Non-Empty Mask (RxFLvIMsk) 0x0 (MASK): Receive FIFO Non-Empty Mask 0x1 (NOMASK): No Receive FIFO Non-Empty Mask	0x0
3	R/W	SofMsk	Mode: Host and Device Start of (micro)Frame Mask (SofMsk) 0x0 (MASK): Start of Frame Mask 0x1 (NOMASK): No Start of Frame Mask	0x0
2	R/W	OTGIntMsk	Mode: Host and Device OTG Interrupt Mask (OTGIntMsk) 0x0 (MASK): OTG Interrupt Mask 0x1 (NOMASK): No OTG Interrupt Mask	0x0
1	R/W	ModeMisMsk	Mode: Host and Device Mode Mismatch Interrupt Mask (ModeMisMsk) 0x0 (MASK): Mode Mismatch Interrupt Mask 0x1 (NOMASK): No Mode Mismatch Interrupt Mask	0x0
0	R	RESERVED	RESERVED	0x0

GRXTSR

A read to the Receive Status Debug Read register returns the contents of the top of the Receive FIFO.

Offset:0x1c Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31			Reserved Field: Yes	
30:25	R	RESERVED	RESERVED	0x0
24:21	R	FN	Mode: Device only Frame Number (FN) This is the least significant 4 bits of the (micro)Frame number in which the packet is received on the USB. This field is supported only when isochronous OUT endpoints are supported.	0x0

Bits	Access	Name	Description	Reset
20:17	R	PktSts	Packet Status (PktSts) indicates the status of the received packet. In host mode, 4'b0010: IN data packet received 4'b0011: IN transfer completed (triggers an interrupt) 4'b0101: Data toggle error (triggers an interrupt) 4'b0111: Channel halted (triggers an interrupt) Others: Reserved Reset:4'b0 In device mode, 4'b0001: Global OUT NAK (triggers an interrupt) 4'b0010: OUT data packet received 4'b0011: OUT transfer completed (triggers an interrupt) 4'b0100: SETUP transaction completed (triggers an interrupt) 4'b0110: SETUP data packet received Others: Reserved Reset:4'h0 Values: 0x1 (OUTNAK): Global OUT NAK in device mode (triggers an interrupt) 0x2 (INOUTDPRX): IN data packet received in host mode and OUT data packet received in device mode 0x3 (INOUTTRCOM): IN or OUT transfer completed in both host and device mode (triggers an interrupt) 0x4 (DSETUPCOM): SETUP transaction completed in device mode (triggers an interrupt) 0x5 (DTTOG): Data toggle error (triggers an interrupt) in host mode 0x6 (DSETUPRX): SETUP data packet received in device mode 0x7 (CHHALT): Channel halted in host mode (triggers an interrupt)	0x0
16:15	R	DPID	Data PID (DPID) In host mode, indicates the Data PID of the received packet. In device mode, indicates the Data PID of the received OUT data packet. 2'b00: DATA0 2'b10: DATA1 2'b01: DATA2 2'b11: MDATA Reset: 2'h0 Values: 0x0 (DATA0): DATA0 0x2 (DATA1): DATA1 0x1 (DATA2): DATA2 0x3 (MDATA): MDATA	0x0
14:4	R	BCnt	Byte Count (BCnt) In host mode, indicates the byte count of the received IN data packet. In device mode, indicates the byte count of the received data packet.	0x0

Bits	Access	Name	Description	Reset
3:0	R	ChNum	Channel Number (ChNum) Mode: Host only Indicates the channel number to which the current received packet belongs. Endpoint Number (EPNum) Mode: Device only Indicates the endpoint number to which the current received packet belongs. Values: 0x0 (CHEP0): Channel or EndPoint 0 0x1 (CHEP1): Channel or EndPoint 1 0x2 (CHEP2): Channel or EndPoint 2 0x3 (CHEP3): Channel or EndPoint 3 0x4 (CHEP4): Channel or EndPoint 4 0x5 (CHEP5): Channel or EndPoint 5 0x6 (CHEP6): Channel or EndPoint 6 0x7 (CHEP7): Channel or EndPoint 7 0x8 (CHEP8): Channel or EndPoint 8 0x9 (CHEP9): Channel or EndPoint 9 0xa (CHEP10): Channel or EndPoint 10 0xb (CHEP11): Channel or EndPoint 11 0xc (CHEP12): Channel or EndPoint 12 0xd (CHEP13): Channel or EndPoint 13 0xe (CHEP14): Channel or EndPoint 14 0xf (CHEP15): Channel or EndPoint 15	0x0

GRXTSP

A read to the Receive Status Read and Pop register returns the contents of the top of the Receive FIFO and additionally pops the top data entry out of the RxFIFO.

Offset:0x20 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31		RESERVED	Reserved Field: Yes	
30:25	R	RESERVED	RESERVED	
24:21	R	FN	Mode: Device only Frame Number (FN) This is the least significant 4 bits of the (micro)Frame number in which the packet is received on the USB. This field is supported only when isochronous OUT endpoints are supported.	0x0

Bits	Access	Name	Description	Reset
20:17	R	PktSts	Packet Status (PktSts) indicates the status of the received packet. In host mode, 4'b0010: IN data packet received 4'b0011: IN transfer completed (triggers an interrupt) 4'b0101: Data toggle error (triggers an interrupt) 4'b0111: Channel halted (triggers an interrupt) Others: Reserved In device mode, 4'b0001: Global OUT NAK (triggers an interrupt) 4'b0010: OUT data packet received 4'b0011: OUT transfer completed (triggers an interrupt) 4'b0100: SETUP transaction completed (triggers an interrupt) 4'b0110: SETUP data packet received Others: Reserved	0x0
16:15	R	DPID	Data PID (DPID) In host mode, indicates the Data PID of the received packet. In device mode, indicates the Data PID of the received OUT data packet. 2'b00: DATA0 2'b10: DATA1 2'b01: DATA2 2'b11: MDATA	0x0
14:4	R	BCnt	Byte Count (BCnt) In host mode, indicates the byte count of the received IN data packet. In device mode, indicates the byte count of the received data packet.	0x0

Bits	Access	Name	Description	Reset
3:0	R	ChNum	Channel Number (ChNum) Mode: Host only Indicates the channel number to which the current received packet belongs. Endpoint Number (EPNum) Mode: Device only Indicates the endpoint number to which the current received packet belongs. Values: 0x0 (CHEP0): Channel or EndPoint 0 0x1 (CHEP1): Channel or EndPoint 1 0x2 (CHEP2): Channel or EndPoint 2 0x3 (CHEP3): Channel or EndPoint 3 0x4 (CHEP4): Channel or EndPoint 4 0x5 (CHEP5): Channel or EndPoint 5 0x6 (CHEP6): Channel or EndPoint 6 0x7 (CHEP7): Channel or EndPoint 7 0x8 (CHEP8): Channel or EndPoint 8 0x9 (CHEP9): Channel or EndPoint 9 0xa (CHEP10): Channel or EndPoint 10 0xb (CHEP11): Channel or EndPoint 11 0xc (CHEP12): Channel or EndPoint 12 0xd (CHEP13): Channel or EndPoint 13 0xe (CHEP14): Channel or EndPoint 14 0xf (CHEP15): Channel or EndPoint 15	0x0

GRXFSIZ

The application can program the RAM size that must be allocated to the RxFIFO.

Offset:0x24 Total Reset Value:0x0000_0400

Bits	Access	Name	Description	Reset
31:22			Reserved Field: Yes	
21:11	R	RESERVED	RESERVED	0x0
10:0	R/W	RxFDep	Mode: Host and Device RxFIFO Depth (RxFDep) This value is in terms of 32-bit words. Minimum value is 16 Maximum value is 32,768 The power-on reset value of this register is specified as the Largest Rx Data FIFO Depth during configuration. If Enable Dynamic FIFO Sizing is selected in coreConsultant, these flops are optimized, and reads return the power-on value. If Enable Dynamic FIFO Sizing is selected in coreConsultant, you can write a new value in this field. Programmed values must not exceed the power-on value.	0x400

GNPTXFSIZ

The application can program the RAM size and the memory start address for the Non- periodic TxFIFO

Offset:0x28 Total Reset Value:0x0400_0400

Bits	Access	Name	Description	Reset
31:27			Reserved Field: Yes	
26:14	R/W	NPTxFDep	Non-periodic TxFIFO Depth (NPTxFDep) The NPTxFDep field description is valid only for, host mode Minimum value is 16 Maximum value is 32,768 Host: IN Endpoint TxFIFO 0 Depth (INEPTxF0Dep) The INEPTxF0Dep field description is valid only for device mode when OTG_EN_DED_TX_FIFO=1. This value is in terms of 32-bit words. Minimum value is 16 Maximum value is 32,768	0x400
15:11			Reserved Field: Yes	
10:0	NPTxFStAddr	R/W	Non-periodic Transmit RAM Start Address (NPTxFStAddr) The NPTxFStAddr field description is valid only for host mode. This field contains the memory start address for Non-periodic Transmit FIFO RAM. IN Endpoint FIFO0 Transmit RAM Start Address(INEPTxF0StAddr) The INEPTxF0StAddr field description is valid only for device mode when OTG_EN_DED_TX_FIFO=1. This field contains the memory start address for IN Endpoint Transmit FIFO# 0. Programmed values must not exceed the power-on value.	0x400

GNPTXSTS

In Device mode, this register is valid only in Shared FIFO operation.

Offset:0x2c Total Reset Value:0x0008_0400

Bits	Access	Name	Description	Reset
31	R	RESERVED	RESERVED	0x0
30:24	R	NPTxQTop	Top of the Non-periodic Transmit Request Queue (NPTxQTop) Entry in the Non-periodic Tx Request Queue that is currently being processed by the MAC. Bits [30:27]: Channel/endpoint number Bits [26:25]: 2'b00: IN/OUT token - 2'b01: Zero-length transmit packet (device IN/host OUT) - 2'b10: PING/CSPLIT token - 2'b11: Channel halt command Bit [24]: Terminate (last Entry for selected channel/endpoint)	0x0

Bits	Access	Name	Description	Reset
23:16	R	NPTxQSpAvail	Non-periodic Transmit Request Queue Space Available (NPTxQSpAvail) Indicates the amount of free space available in the Non-periodic Transmit Request Queue. This queue holds both IN and OUT requests in Host mode. Device mode has only IN requests. 0x0 (FULL): Non-periodic Transmit Request Queue is full 0x1 (QUE1): 1 location available 0x2 (QUE2): 2 locations available 0x3 (QUE3): 3 locations available 0x4 (QUE4): 4 locations available 0x5 (QUE5): 5 locations available 0x6 (QUE6): 6 locations available 0x7 (QUE7): 7 locations available 0x8 (QUE8): 8 locations available	0x8
15:0	R	NPTxFSpAvail	Non-periodic TxFIFO Space Avail (NPTxFSpAvail) Indicates the amount of free space available in the Non-periodic TxFIFO. Values are in terms of 32-bit words. 16'h0: Non-periodic TxFIFO is full 16'h1: 1 word available 16'h2: 2 words available 16'hn: n words available (where 0 <= n <= 32,768) 16'h8000: 32,768 words available	0x400

GI2CCTL

In Device mode, this register is valid only in Shared FIFO operation.

Offset:0x30 Total Reset Value:0x0000_0000 RESERVED

GPVNDCTL

The application can use this register to access PHY registers. It is implemented only if Enable PHY Vendor Control Interface was selected during coreConsultant configuration (parameter OTG_VENDOR_CTL_INTERFACE = 1). For a UTMI+ PHY, the DWC_otg core uses the UTMI+ Vendor Control interface for PHY register access. For a ULPI PHY, the core uses the ULPI interface for PHY register access. The application sets Vendor Control register for PHY register access and times the PHY register access. The application polls the VStatus Done bit in this register for the completion of the PHY register access.

Offset:0x34 Total Reset Value:0x0000_0000 RESERVED

GGPIO

The application can use this register for general purpose input/output ports or for debugging.

Offset:0x38 Total Reset Value:0x12345678 RESERVED

GUID

This is a read/write register containing the User ID. It is implemented only if Remove Optional Features? was deselected during coreConsultant configuration (parameter OTG_RM_OPT_FEATURES = 0). The power-on value for this register is specified as the Power-on Value of User ID Register User Identification Register during coreConsultant configuration (parameter OTG_USERID). This register can be used in the following ways:

Offset:0x3c Total Reset Value:0x0000_0000 RESERVED

GSNPSID

This read-only register contains the release number of the core being used.

Offset:0x40 Total Reset Value:0x000b_0000 RESERVED

GHWCFG1

This register contains the logical endpoint direction(s) selected using coreConsultant.

Offset:0x44 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:0	R	EpDir	This 32-bit field uses two bits per endpoint to determine the endpoint direction. Endpoint Bits [31:30]: Endpoint 15 direction Bits [29:28]: Endpoint 14 direction ... Bits [3:2]: Endpoint 1 direction Bits[1:0]: Endpoint 0 direction (always BIDIR) Direction 2'b00: BIDIR (IN and OUT) endpoint 2'b01: IN endpoint 2'b10: OUT endpoint 2'b11: Reserved	0x0

GHWCFG2

This register contains configuration options selected using coreConsultant.

Offset:0x48 Total Reset Value:0x8883_EC2A

Bits	Name	Access	Description	Reset
31	R	RESERVED	RESERVED	
30:26	R	TknQDepth	Device Mode IN Token Sequence Learning Queue Depth (TknQDepth) Range: 0-30	0x8
25:24	R	PTxQDepth	Host Mode Periodic Request Queue Depth (PTxQDepth) 2'b00: 2 2'b01: 4 2'b10: 8 2'b11:16	0x8
23:22	R	NPTxQDepth	Non-periodic Request Queue Depth (NPTxQDepth) 2'b00: 2 2'b01: 4 2'b10: 8 Others: Reserved	0x8
21	R	RESERVED	RESERVED	
20	R	MultiProcIntrpt	Multi Processor Interrupt Enabled (MultiProcIntrpt) 0x0 (DISABLED): No Multi Processor Interrupt Enabled 0x1 (ENABLED): Multi Processor Interrupt Enabled	0x0
19	R	DynFifoSizing	Dynamic FIFO Sizing Enabled (DynFifoSizing) 0x0 (DISABLED): Dynamic FIFO Sizing Disabled 0x1 (ENABLED): Dynamic FIFO Sizing Enabled	0x1

Bits	Name	Access	Description	Reset
18	R	PerioSupport	Periodic OUT Channels Supported in Host Mode (PerioSupport) 0x0 (DISABLED): Periodic OUT Channels is not supported in Host Mode 0x1 (ENABLED): Periodic OUT Channels Supported in Host Mode Supported	0x1
17:14	R	NumHstChnl	Number of Host Channels (NumHstChnl) Indicates the number of host channels supported by the core in Host mode. The range of this field is 0-15: 0 specifies 1 channel, 15 specifies 16 channels. 0x0 (HOSTCH0): Host Channel 1 0x1 (HOSTCH1): Host Channel 2 0x2 (HOSTCH2): Host Channel 3 0x3 (HOSTCH3): Host Channel 4 0x4 (HOSTCH4): Host Channel 5 0x5 (HOSTCH5): Host Channel 6 0x6 (HOSTCH6): Host Channel 7 0x7 (HOSTCH7): Host Channel 8 0x8 (HOSTCH8): Host Channel 9 0x9 (HOSTCH9): Host Channel 10 0xa (HOSTCH10): Host Channel 11 0xb (HOSTCH11): Host Channel 12 0xc (HOSTCH12): Host Channel 13 0xd (HOSTCH13): Host Channel 14 0xe (HOSTCH14): Host Channel 15 0xf (HOSTCH15): Host Channel 16	0xe
13:10	R	NumDevEps	Number of Device Endpoints (NumDevEps) Indicates the number of device endpoints supported by the core in Device mode. The range of this field is 0-15. 0x0 (ENDPT0): End point 0 0x1 (ENDPT1): End point 1 0x2 (ENDPT2): End point 2 0x3 (ENDPT3): End point 3 0x4 (ENDPT4): End point 4 0x5 (ENDPT5): End point 5 0x6 (ENDPT6): End point 6 0x7 (ENDPT7): End point 7 0x8 (ENDPT8): End point 8 0x9 (ENDPT9): End point 9 0xa (ENDPT10): End point 10 0xb (ENDPT11): End point 11 0xc (ENDPT12): End point 12 0xd (ENDPT13): End point 13 0xe (ENDPT14): End point 14 0xf (ENDPT15): End point 15	0x6

Bits	Name	Access	Description	Reset
9:8	R	FSPhyType	Full-Speed PHY Interface Type (FSPhyType) 0x0 (NO_FS): Full-speed interface not supported 0x1 (FS): Dedicated full-speed interface is supported 0x2 (FSPLUSUTMI): FS pins shared with UTMI+ pins is supported 0x3 (FSPLUSULPI): FS pins shared with ULPI pins is supported	0x0
7:6	R	HSPhyType	High-Speed PHY Interface Type (HSPhyType) 0x0 (NOHS): High-Speed interface not supported 0x1 (UTMIPLUS): High Speed Interface UTMI+ is supported 0x2 (ULPI): High Speed Interface ULPI is supported 0x3 (UTMIPUSULPI): High Speed Interfaces UTMI+ and ULPI is supported	0x1
5	R	SingPnt	Point-to-Point (SingPnt) 0x0 (MULTIPOINT): Multi-point application (hub and split support) 0x1 (SINGLEPOINT): Single-point application (no hub and split support)	0x0
4:3	R	OtgArch	Architecture (OtgArch) 0x0 (SLAVEMODE): Slave Mode 0x1 (EXTERNALDMA): External DMA Mode 0x2 (INTERNALDMA): Internal DMA Mode	0x2
2:0	R	OtgMode	Mode of Operation (OtgMode) 0x0 (HNPSRP): HNP- and SRP-Capable OTG (Host and Device) 0x1 (SRPOTG): SRP-Capable OTG (Host and Device) 0x2 (NHNPNSRP): Non-HNP and Non-SRP Capable OTG (Host and Device) 0x3 (SRPCAPD): SRP-Capable Device 0x4 (NONOTGD): Non-OTG Device 0x5 (SRPCAPH): SRP-Capable Host 0x6 (NONOTGH): Non-OTG Host	0x2

GHWCFG3

This register contains configuration options selected using coreConsultant.

Offset:0x4c Total Reset Value:0x0400_00e8s

Bits	Access	Name	Description	Reset
31:16	R	DfifoDepth	DFIFO Depth (DfifoDepth - EP_LOC_CNT) This value is in terms of 32-bit words. Minimum value is 32 Maximum value is 32,768	0x400
15	R	LPMMode	LPM mode specified for Mode of Operation. 0x0 (DISABLED): LPM disabled 0x1 (ENABLED): LPM enabled	0x0
14	R	BCSupport	This bit indicates the controller support for Battery Charger. 0x0 (DISABLED): No Battery Charger Support 0x1 (ENABLED): Battery Charger Support present	0x0

Bits	Access	Name	Description	Reset
13	R	HSICMode	HSIC mode specified for Mode of Operation Value Range: 0 - 1 0x0 (DISABLED): No HSIC capability 0x1 (ENABLED): HSIC-capable with shared UTMI PHY interface	0x0
12	R	ADPSupport	This bit indicates whether ADP logic is present within or external to the controller 0x0 (DISABLED): ADP logic is not present along with the controller 0x1 (ENABLED): ADP logic is present along with the controller	0x0
11	R	RstType	Reset Style for Clocked always Blocks in RTL (RstType) 0x0 (ASYNCRST): Asynchronous reset is used in the core 0x1 (SYNCRST): Synchronous reset is used in the core	0x0
10	R	OptFeature	Optional Features Removed (OptFeature) Indicates whether the User ID register, GPIO interface ports, and SOF toggle and counter ports were removed for gate count optimization by enabling Remove Optional Features. 0x0 (DISABLED): Optional features were not Removed 0x1 (ENABLED): Optional Features have been Removed	0x1
9	R	VndctlSupt	Vendor Control Interface Support (VndctlSupt) 0x0 (DISABLED): Vendor Control Interface is not available. 0x1 (ENABLED): Vendor Control Interface is available.	0x0
8	R	I2CIntSel	I2C Selection (I2CIntSel) 0x0 (DISABLED): I2C Interface is not available 0x1 (ENABLED): I2C Interface is available	0x0
7	R	OtgEn	OTG Function Enabled (OtgEn) The application uses this bit to indicate the OTG capabilities of the controller . 0x0 (DISABLED): Not OTG Capable 0x1 (ENABLED): OTG Capable	0x1
6:4	R	PktSizeWidth	Width of Packet Size Counters (PktSizeWidth) 0x0 (BITS4): Width of Packet Size Counter 4 0x1 (BITS5): Width of Packet Size Counter 5 0x2 (BITS6): Width of Packet Size Counter 6 0x3 (BITS7): Width of Packet Size Counter 7 0x4 (BITS8): Width of Packet Size Counter 8 0x5 (BITS9): Width of Packet Size Counter 9 0x6 (BITS10): Width of Packet Size Counter 10	0x6
3:0	R	XferSizeWidth	Width of Transfer Size Counters (XferSizeWidth) 0x0 (WIDTH11): Width of Transfer Size Counter 11 bits 0x1 (WIDTH12): Width of Transfer Size Counter 12 bits 0x2 (WIDTH13): Width of Transfer Size Counter 13 bits 0x3 (WIDTH14): Width of Transfer Size Counter 14 bits 0x4 (WIDTH15): Width of Transfer Size Counter 15 bits 0x5 (WIDTH16): Width of Transfer Size Counter 16 bits 0x6 (WIDTH17): Width of Transfer Size Counter 17 bits 0x7 (WIDTH18): Width of Transfer Size Counter 18 bits 0x8 (WIDTH19): Width of Transfer Size Counter 19 bits	0x8

GHWCFG4

This register contains configuration options selected using coreConsultant.

Offset:0x50 Total Reset Value:DA11_AA20

Bits	Access	Name	Description	Reset
31	DescDMA	R	Scatter/Gather DMA configuration 0x0 (CONFIG1): Non Dynamic configuration 0x1 (CONFIG2): Dynamic configuration	0x1
30	DescDMAEnabled	R	Scatter/Gather DMA configuration 0x0 (DISABLE): Non-Scatter/Gather DMA configuration 0x1 (ENABLE): Scatter/Gather DMA configuration	0x1
29:26	INEps	R	Number of Device Mode IN Endpoints Including Control Endpoints (INEps) 0x0 (ENDPT1): 1 IN Endpoint 0x1 (ENDPT2): 2 IN Endpoints 0x2 (ENDPT3): 3 IN Endpoints 0x3 (ENDPT4): 4 IN Endpoints 0x4 (ENDPT5): 5 IN Endpoints 0x5 (ENDPT6): 6 IN Endpoints 0x6 (ENDPT7): 7 IN Endpoints 0x7 (ENDPT8): 8 IN Endpoints 0x8 (ENDPT9): 9 IN Endpoints 0x9 (ENDPT10): 10 IN Endpoints 0xa (ENDPT11): 11 IN Endpoints 0xb (ENDPT12): 12 IN Endpoints 0xc (ENDPT13): 13 IN Endpoints 0xd (ENDPT14): 14 IN Endpoints 0xe (ENDPT15): 15 IN Endpoints 0xf (ENDPT16): 16 IN Endpoints	0x6
25	DedFifoMode	R	Enable Dedicated Transmit FIFO for device IN Endpoints (DedFifoMode) 0x0 (DISABLED): Dedicated Transmit FIFO Operation not enabled 0x1 (ENABLED): Dedicated Transmit FIFO Operation enabled	0x1
24	SessEndFltr	R	session_end Filter Enabled (SessEndFltr) Values: 0x0 (DISABLED): No filter 0x1 (ENABLED): Filter	0x0
23	BValidFltr	R	b_valid Filter Enabled (BValidFltr) Values: 0x0 (DISABLED): No Filter 0x1 (ENABLED): Filter	0x0
22	AValidFltr	R	a_valid Filter Enabled (AValidFltr) Values: 0x0 (DISABLED): No filter 0x1 (ENABLED): Filter Value After Reset: 0x0 Exists: Always	0x0

Bits	Access	Name	Description	Reset
21	VBusValidFiltr	R	VBUS Valid Filter Enabled (VBusValidFiltr) Values: 0x0 (DISABLED): Vbus Valid Filter Disabled 0x1 (ENABLED): Vbus Valid Filter Enabled	0x0
20	IddgFiltr	R	IDDIG Filter Enable (IddgFiltr) Values: 0x0 (DISABLED): Iddig Filter Disabled 0x1 (ENABLED): Iddig Filter Enabled	0x1
19:16	NumCtlEps	R	Number of Device Mode Control Endpoints in Addition to Endpoint 0 (NumCtlEps) Range: 0-15 Values: 0x0 (ENDPT0): End point 0 0x1 (ENDPT1): End point 1 0x2 (ENDPT2): End point 2 0x3 (ENDPT3): End point 3 0x4 (ENDPT4): End point 4 0x5 (ENDPT5): End point 5 0x6 (ENDPT6): End point 6 0x7 (ENDPT7): End point 7 0x8 (ENDPT8): End point 8 0x9 (ENDPT9): End point 9 0xa (ENDPT10): End point 10 0xb (ENDPT11): End point 11 0xc (ENDPT12): End point 12 0xd (ENDPT13): End point 13 0xe (ENDPT14): End point 14 0xf (ENDPT15): End point 15	0x1
15:14	PhyDataWidth	R	UTMI+ PHY/ULPI-to-Internal UTMI+ Wrapper Data Width (PhyDataWidth)When a ULPI PHY is used, an internal wrapper converts ULPI to UTMI+. Values: 0x0 (WIDTH1): 8 bits 0x1 (WIDTH2): 16 bits 0x2 (WIDTH3): 8/16 bits, software selectable	0x2
13	EnhancedLPMSupt	R	Enhanced LPM Support (EnhancedLPMSupt) This bit indicates that the controller supports the following behavior: L1 Entry Behavior based on FIFO Status Values: 0x1 (ENABLED): Enhanced LPM Support is enabled	0x1

Bits	Access	Name	Description	Reset
12	ACGSupt	R	Active Clock Gating Support This bit indicates that the controller supports the Dynamic (Switching) Power Reduction during periods when there is no USB and AHB Traffic. Values: 0x0 (DISABLED) 0x1 (ENABLED): Active Clock Gating Support	0x0
11	ipgisocSupt	R	Interpacket Gap ISOC OUT Worst-case Support (ipgisocSupt) This bit indicates that the controller supports the worst-case scenario of Rx followed by Rx Inter Packet Gap (IPG) (32-bit times) as per the UTMI Specification for any token following an ISOC OUT token. Without this support, when any token follows an ISOC OUT token with the worst-case IPG, the controller does not detect the followed token. The worst-case IPG of the controller without this support depends on the AHB and PHY clock frequency. By default IPG Support is enabled. Values: 0x0 (DISABLED): Interpacket Gap ISOC OUT Worst-case Support is Disabled 0x1 (ENABLED): Interpacket Gap ISOC OUT Worst-case Support is Enabled (Default)	0x1
10	ServIntFlow	R	Service Interval Flow This bit indicates that the controller supports Service-Interval based scheduling flow for ISOC IN EPs. Values: 0x0 (DISABLED): Service Interval Flow not supported 0x1 (ENABLED): Service Interval Flow supported	0x0
9	EnhancedLPMSupt1	R	Enhanced LPM Support1 (EnhancedLPMSupt1) Values: 0x0 (DISABLED): Reject L1 Request even if Non-Periodic (Bulk/Interrupt) Tx FIFO is not empty. 0x1 (ENABLED): Accept L1 Request even if Non-Periodic (Bulk/Interrupt) Tx FIFO is not empty	0x1
8	Reserved_8	R	RESERVED	0x0
7	ExtendedHibernation	R	Enable Hibernation Values: 0x0 (DISABLED): Extended Hibernation feature not enabled 0x1 (ENABLED): Extended Hibernation feature enabled	0x0
6	Hibernation	R	Enable Hibernation (Hibernation) Values: 0x0 (DISABLED): Hibernation feature disabled 0x1 (ENABLED): Hibernation feature enabled	0x0

Bits	Access	Name	Description	Reset
5	AhbFreq	R	Minimum AHB Frequency Less Than 60 MHz (AhbFreq) Values: 0x0 (DISABLED): Minimum AHB Frequency More Than 60 MHz 0x1 (ENABLED): Minimum AHB Frequency Less Than 60 MHz	0x1
4	PartialPwrDn	R	Enable Partial Power Down (PartialPwrDn) Values: 0x0 (DISABLED): Partial Power Down disabled 0x1 (ENABLED): Partial Power Down enabled	0x0
3:0	NumDevPerioEps	R	Number of Device Mode Periodic IN Endpoints (NumDevPerioEps) Range: 0-15 Values: 0x0 (Value0): Number of Periodic IN EPs is 0 0x1 (Value1): Number of Periodic IN EPs is 1 0x2 (Value2): Number of Periodic IN EPs is 2 0x3 (Value3): Number of Periodic IN EPs is 3 0x4 (Value4): Number of Periodic IN EPs is 4 0x5 (Value5): Number of Periodic IN EPs is 5 0x6 (Value6): Number of Periodic IN EPs is 6 0x7 (Value7): Number of Periodic IN EPs is 7 0x8 (Value8): Number of Periodic IN EPs is 8 0x9 (Value9): Number of Periodic IN EPs is 9 0xa (Value10): Number of Periodic IN EPs is 10 0xb (Value11): Number of Periodic IN EPs is 11 0xc (Value12): Number of Periodic IN EPs is 12 0xd (Value13): Number of Periodic IN EPs is 13 0xe (Value14): Number of Periodic IN EPs is 14 0xf (Value15): Number of Periodic IN EPs is 15	0x0

GLPMCFCG

Register to control the LPM functionality of the controller.

Offset:0x54 Total Reset Value:0x000b_0000

Bits	Access	Name	Description	Reset
31	RESERVED	R	RESERVED	0x0
30	RESERVED	R	RESERVED	0x0

Bits	Access	Name	Description	Reset
29	LPM_RestoreSlpSts	R/W	LPM Restore Sleep Status (LPM_RestoreSlpSts) When the application power gates the core (Partial Power Down / Hibernation) the application needs to program this bit to restore the LPM status in the core. The application needs to program this bit, during restore process, based on whether it decides to go into Shallow Sleep (Clock Gating Only) or Deep Sleep (Power Gating) based on the BESL value received from the Host Values: 0x0 (DISABLED): Puts the core in Shallow Sleep mode based on the BESL value from the Host 0x1 (ENABLED): Puts the core in Deep Sleep mode based on the BESL value from the Host	0x0
28	LPM_EnBESL	R/W	LPM Enable BESL (LPM_EnBESL) This bit enables the BESL feature as defined in LPM Errata 0x0 (DISABLED): BESL is disabled 0x1 (ENABLED): BESL is enabled as defined in LPM Errata	0x0
27:25	LPM_RetryCnt_Sts	R	LPM Retry Count Status (LPM_RetryCnt_Sts) Host Mode: Number of LPM Host Retries still remaining to be transmitted for the current LPM sequence. Values: 0x0 (RETRY_REM0): Zero LPM retries remaining 0x1 (RETRY_REM1): One LPM retry remaining 0x2 (RETRY_REM2): Two LPM retries remaining 0x3 (RETRY_REM3): Three LPM retries remaining 0x4 (RETRY_REM4): Four LPM retries remaining 0x5 (RETRY_REM5): Five LPM retries remaining 0x6 (RETRY_REM6): Six LPM retries remaining 0x7 (RETRY_REM7): Seven LPM retries remaining	0x0
24	SndLPM	R/W	Send LPM Transaction (SndLPM) Host Mode: When set by application software, an LPM transaction containing two tokens, EXT & LPM is sent. Cleared by the hardware once a valid response (STALL./NYET/ACK) is received from the Device or the core had finished transmitting the programmed number of LPM retries. Note that this bit must be set only when the host is connected to local port. Values: 0x0 (DISABLED): In host-only mode: Received the response from the device for the LPM transaction 0x1 (ENABLED): In host-only mode: Sending LPM transaction containing EXT and LPM tokens	0x0

Bits	Access	Name	Description	Reset
23:21	LPM_Retry_Cnt	R/W	LPM Retry Count (LPM_Retry_Cnt) Values: 0x0 (RETRY0): Zero LPM retries 0x1 (RETRY1): One LPM retry 0x2 (RETRY2): Two LPM retries 0x3 (RETRY3): Three LPM retries 0x4 (RETRY4): Four LPM retries 0x5 (RETRY5): Five LPM retries 0x6 (RETRY6): Six LPM retries 0x7 (RETRY7): Seven LPM retries	0x0
20:17	LPM_Chnl_Indx	R/W	LPM Channel Index Values: 0x0 (CH0): Channel 0 0x1 (CH1): Channel 1 0x2 (CH2): Channel 2 0x3 (CH3): Channel 3 0x4 (CH4): Channel 4 0x5 (CH5): Channel 5 0x6 (CH6): Channel 6 0x7 (CH7): Channel 7 0x8 (CH8): Channel 8 0x9 (CH9): Channel 9 0xa (CH10): Channel 10 0xb (CH11): Channel 11 0xc (CH12): Channel 12 0xd (CH13): Channel 13 0xe (CH14): Channel 14 0xf (CH15): Channel 15 Value After Reset: 0x0 Exists: (OTG_MODE != 3)	
16	L1ResumeOK	R	Sleep State Resume OK (L1ResumeOK) Values: 0x0 (NOTOK): The application/core cannot start Resume from Sleep state 0x1 (OK): The application/core can start Resume from Sleep state	0x0
15	SlpSts	R	Port Sleep Status (SlpSts) Device Mode: This bit is set as long as a Sleep condition is present on the USB bus. Values: 0x0 (CORE_NOT_IN_L1): In Host or Device mode, this bit indicates core is not in L1 0x1 (CORE_IN_L1): In Host mode, this bit indicates the core transitions to Sleep state as a successful LPM transaction. In Device mode, the core enters the Sleep state when an ACK response is sent to an LPM transaction	0x0

Bits	Access	Name	Description	Reset
14:13	CoreL1Res	R	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 Values: 0x0 (LPMRESP1): ERROR : No handshake response 0x1 (LPMRESP2): STALL response 0x2 (LPMRESP3): NYET response 0x3 (LPMRESP4): ACK response	0x0

Bits	Access	Name	Description	Reset
12:8	HIRD_Thres	R/W	BESL/HIRD Threshold (HIRD_Thres) Device Mode: Note: When a ULPI/UTMI PHY associated with the controller (Host or device) does not support the LPM sleep feature clock gating, it is recommended to keep the HIRD_Thres[12] bit in disabled state (= 1'b0). This ensures the ULPI wrapper is in enabled mode and detects any wakeup events. 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 1b1. 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_Thres[3:0]. The Device sends a NYET response when the received HIRD in LPM token is greater than HIRD threshold. 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 1'b1. 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. 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. When EnBESL = 1'b0, HIRD_Thres must not be programmed with a value greater than 4'b1100 in host mode since this exceeds maximum TL1HubDrvResume2. When EnBESL = 1'b1, HIRD_Thres must not be programmed with a value greater than 4'b0110 in host mode since this exceeds maximum TL1HubDrvResume2. For additional details, see the "Additional Details on GLPMC_CFG.HIRD_Thres Register Field" section in the databook. Value After Reset: 0x0 Exists: OTG_ENABLE_LPM==1	0x0

Bits	Access	Name	Description	Reset
7	EnbISlpM	R/W	Enable utmi_sleep_n (EnbISlpM) Values: 0x0 (DISABLED): utmi_sleep_n assertion from the core is not transferred to the external PHY 0x1 (ENABLED): utmi_sleep_n assertion from the core is transferred to the external PHY when utmi_l1_suspend_n cannot be asserted	0x0
6	bRemoteWake	OTG_MODE!=3 & OTG_MODE!=4 ? R/W : R	RemoteWakeEnable (bRemoteWake) Values: 0x0 (DISABLED): Remote Wakeup is disabled 0x1 (ENABLED): In Host or device mode, this field takes the value of remote wake up	0x0
5:2	HIRD	R/W	Host-Initiated Resume Duration (HIRD) EnBESL = 1'b0 Host Initiated Resume Duration. 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: This field is read only and is updated with the Received LPM Token HIRD bmAttribute when an ACK/NYET/STALL response is sent to an LPM transaction. If HIRD[3:0], 4'b0000, THIRD(us) = 50 4'b0001, THIRD(us) = 125 4'b0010, THIRD(us) = 200 4'b0011, THIRD(us) = 275 4'b0100, THIRD(us) = 350 4'b0101, THIRD(us) = 425 4'b0110, THIRD(us) = 500 4'b0111, THIRD(us) = 575 4'b1000, THIRD(us) = 650 4'b1001, THIRD(us) = 725 4'b1010, THIRD(us) = 800 4'b1011, THIRD(us) = 875 4'b1100, THIRD(us) = 950 4'b1101, THIRD(us) = 1025 4'b1110, THIRD(us) = 1100 4'b1111, THIRD(us) = 1175 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: This field is updated with the Received LPM Token BESL bmAttribute when an ACK/NYET/STALL response is sent to an LPM transaction. If BESL[3:0], 4'b0000, TBESL(us) = 125	0x0

Bits	Access	Name	Description	Reset
			4'b0001, TBESL(us) = 150 4'b0010, TBESL(us) = 200 4'b0011, TBESL(us) = 300 4'b0100, TBESL(us) = 400 4'b0101, TBESL(us) = 500 4'b0110, TBESL(us) = 1000 4'b0111, TBESL(us) = 2000 4'b1000, TBESL(us) = 3000 4'b1001, TBESL(us) = 4000 4'b1010, TBESL(us) = 5000 4'b1011, TBESL(us) = 6000 4'b1100, TBESL(us) = 7000 4'b1101, TBESL(us) = 8000 4'b1110, TBESL(us) = 9000 4'b1111, TBESL(us) = 10000	
0	LPMCap	R/W	LPM-Capable (LPMCap) Values: 0x0 (DISABLED): LPM capability is not enabled 0x1 (ENABLED): LPM capability is enabled	0x0

GPWRDN

This is the Hibernation control register. This register is active only during hibernation and ADP. The application can get the status of the wakeup_logic and control it through this register.

Offset:0x58 Total Reset Value:0x0000_0010

Bits	Name	Access	Description	Reset
31:23		RESERVED	Reserved Field: Yes	0x0
22	R	BSessVld	B Session Valid (BSessVld) Values: 0x0 (NOTVALID): B_Valid is 0 0x1 (VALID): B_Valid is 1	0x0
21	R	IDDIG	This bit indicates the status of the signal IDDIG. The application must read this bit after receiving GPWRDN.StsChngInt and decode based on the previous value stored by the application. This bit is valid only when GPWRDN.PMUActv is 1. Values: 0x0 (DISABLED): Host Mode 0x1 (ENABLED): Device Mode	0x0
20:19	R	LineState	LineState This field indicates the current linestate on USB as seen by the PMU module. This bit is valid only when GPWRDN.PMUActv is 1. Values: 0x0 (LS1): Linestate on USB: DM = 0, DP = 0 0x1 (LS2): Linestate on USB: DM = 0, DP = 1 0x2 (LS3): Linestate on USB: DM = 1, DP = 0 0x3 (LS4): Linestate on USB: Not-defined	0x0

Bits	Name	Access	Description	Reset
18	R/W	StsChngIntMsk	StsChngIntMsk Mask for StsChng Interrupt Values: 0x0 (NOMASK): No Status Change Interrupt Mask 0x1 (MASK): Mask for Status Change Interrupt	0x0
17	R/W1C	StsChngInt	Status Change Interrupt (StsChngInt) This field indicates a status change in either the IDDIG or BSessVld signal. Values: 0x0 (DISABLED): No Status change 0x1 (ENABLED): Status change detected	0x0
16	R/W	SRPDetectMsk	SRPDetectMsk Mask for SRPDetect Interrupt Values: 0x0 (NOMASK): No SRPDetect Interrupt Mask 0x1 (MASK): Mask for SRPDetect Interrupt	0x0
15	R/W1C	SRPDetect	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. Values: 0x0 (DISABLED): SRP not detected 0x1 (ENABLED): SRP detected	0x0
14:7	R	RESERVED	Reserved	0x0
6	R/W	DisableVBUS	DisableVBUS Host Mode: The application must program this bit if HPRT0.PrtPwr was programmed to 0 before entering Hibernation. This is to indicate PMU whether session was ended before entering Hibernation. Values: 0x0 (DISABLED): Host mode:HPRT0.PrtPwr was not programmed to 0, and in Device mode:Session Valid 0x1 (ENABLED): Host mode:HPRT0.PrtPwr was programmed to 0 and in Device mode:Session End	0x0
5	R/W	PwrDnSwch	Power Down Switch (PwrDnSwch) This bit indicates to the controller whether the VDD switch is in ON/OFF state. Values: 0x0 (ON): The controller is in ON state 0x1 (OFF): The controller is in OFF state	0x0

Bits	Name	Access	Description	Reset
4	R/W	PwrDnRst_n	Power Down ResetN (PwrDnRst_n) The application must program this bit to reset the core during the Hibernation exit process or during ADP when powering up the core (in case the core was powered off during ADP process). Values: 0x0 (DISABLE): Reset the controller 0x1 (ENABLE): The controller is in normal operation	0x1
3	R/W	PwrDnClmp	Power Down Clamp (PwrDnClmp) The application must program this bit to enable or disable the clamps to all the outputs of the core module to prevent the corruption of other active logic. Values: 0x0 (DISABLE): Disable PMU power clamp 0x1 (ENABLE): Enable PMU power clamp	0x0
2	R	RESERVED	RESERVED	
1	R/W	PMUActv	PMU Active (PMUActv) This is bit is to enable or disable the PMU logic. Values: 0x0 (DISABLE): Disable PMU module 0x1 (ENABLE): Enable PMU module	0x0
0	R/W	PMUIntSel	PMU Interrupt Select (PMUIntSel) 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. Values: 0x0 (DISABLE): Internal DWC_otg_core interrupt is selected 0x1 (ENABLE): External DWC_otg_pmu interrupt is selected	0x0

GDFIFOCFG

Register to configure the DFIFOs for the controller.

Offset:0x5c Total Reset Value:0x0400_0400

Bits	Access	Name	Description	Reset

Bits	Access	Name	Description	Reset
31:16	EPInfoBaseAddr	R/W	This field provides the start address of the EP info controller. The EP info controller manages the stored values in the last few locations of the SPRAM as listed below. Host Buffer DMA mode: One location per channel is used in SPRAM to store the HCDMA value. Host Scatter/Gather DMA mode: Four locations per channel are used in SPRAM to store the Base Descriptor address, Current Descriptor address, Current Buffer Pointer, and the Status Quadlet. Device Buffer DMA mode: One location per endpoint direction is used in SPRAM to store the DIEPDMA and DOEPDMA value. Device Scatter/Gather DMA mode: Four locations per endpoint direction are used in SPRAM to store the Base Descriptor address, Current Descriptor address, Current Buffer Pointer and the Status Quadlet.	0x400
15:0	GDFIFOCfg	R/W	GDFIFOCfg This field is for dynamic programming of the DFIFO Size. This value takes effect only when the application programs a non zero value to this register. The value programmed must conform to the guidelines described in 'FIFO RAM Allocation'. The core does not have any corrective logic if the FIFO sizes are programmed incorrectly.	0x400

GADPCTL

This register is maintained in the PMU module. These register values are used for deciding the timing values by the ADP controller. This register is available only if the ADP controller logic is present within the HS OTG Controller. If the ADP logic is external to the HS OTG Controller, this register is reserved and the register contents are zero. For more information about ADP controller options, see "ADP Programming Flow when ADP Controller Logic is Supplied with the Core " in the Programming Guide.

Offset:0x60 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:29			Reserved Field: Yes	
28:27	R/W	AR	Access Request (AR) Values: 0x0 (Request1): Read/Write 0x1 (Request2): Read 0x2 (Request3): Write 0x3 (Request4): Reserved	0x0
26	R/W	AdpToutMsk	ADP Timeout Interrupt Mask (AdpToutMsk) When this bit is set, it unmask the interrupt because of AdpTmouInt. This bit is valid only if OTG_Ver = 1'b1(GOTGCTL[20]). Values: 0x0 (MASK): Mask ADP Timeout Interrupt 0x1 (NOMASK): Unmask ADP Timeout Interrupt	0x0

Bits	Access	Name	Description	Reset
25	R/W	AdpSnsIntMsk	ADP Sense Interrupt Mask (AdpSnsIntMsk) Values: 0x0 (MASK): Mask ADP Sense Interrupt 0x1 (NOMASK): Unmask ADP Sense Interrupt	0x0
24	R/W	AdpPrbIntMsk	ADP Probe Interrupt (AdpPrbInt) Values: 0x0 (MASK): Mask ADP Probe Interrupt 0x1 (NOMASK): Unmask ADP Probe Interrupt	0x0
23	R/W1C	AdpToutInt	ADP Timeout Interrupt (AdpToutInt) This bit is relevant only for an ADP probe. When this bit is set, it means that the ramp time has completed (GADPCTL.RTIM has reached its terminal value of 0x7FF). This is a debug feature that allows software to read the ramp time after each cycle. This bit is valid only if OTGVer = 1'b1. Values: 0x0 (NOTACTIVE): Not active 0x1 (ACTIVE): ADP Timeout Interrupt	0x0
22	R/W1C	AdpSnsInt	ADP Sense Interrupt (AdpSnsInt) When this bit is set, it means that the VBUS voltage is greater than VadpSns value or VadpSns is reached. This bit is valid only if OTGVer = 1'b1 (GOTGCTL[20]). Values: 0x0 (NOTACTIVE): Not active 0x1 (ACTIVE): ADP Sense Interrupt	0x0
21	R/W1C	AdpPrbInt	ADP Probe Interrupt (AdpPrbInt) When this bit is set, it means that the VBUS voltage is greater than VADP_PRB or VadpPrb is reached. This bit is valid only if OTGVer = 1'b1 (GOTGCTL[20]). Values: 0x0 (NOTACTIVE): Not active 0x1 (ACTIVE): ADP Probe Interrupt	0x0
20	R/W	ADPEn	ADP Enable (ADPEn) When set, the core performs either ADP probing or sensing based on EnaPrb or EnaSns. This bit is valid only if OTG_Ver = 1'b1 (GOTGCTL[20]). Values: 0x0 (DISABLE): Not active 0x1 (ENABLE): Core does ADP Probing or Sensing	0x0
19	R/W1S	ADPRes	ADP Reset (ADPRes) When set, ADP controller is reset. This bit is auto-cleared after the reset procedure is complete in ADP controller. This bit is valid only if OTG_Ver = 1'b1 (GOTGCTL[20]). Values: 0x0 (DISABLE): Not active 0x1 (ENABLE): ADP controller is reset	0x0

Bits	Access	Name	Description	Reset
18	R/W	EnaSns	Enable Sense (EnaSns) When programmed to 1'b1, the core performs a sense operation. This bit is valid only if OTG_Ver = 1'b1 (GOTGCTL[20]). Values: 0x0 (DISABLE): Not active 0x1 (ENABLE): Core performs sense operation	0x0
17	R/W	EnaPrb	Enable Probe (EnaPrb) When programmed to 1'b1, the core performs a probe operation. This bit is valid only if OTG_Ver = 1'b1 (GOTGCTL[20]). Values: 0x0 (DISABLE): Not active 0x1 (ENABLE): Core performs probe operation	0x0
16:6	R	RTIM	RAMP TIME (RTIM) These bits capture the latest time it took for VBUS to ramp from VADP_SINK to VADP_PRB. The bits are defined in units of 32 kHz clock cycles as follows: Values: 0x0 (RTIM1): 1 cycle 0x1 (RTIM2): 2 cycles 0x2 (RTIM3): 3 cycles 0x7ff (RTIM4): 2048 cycles	0x0
5:4	R/W	PrbPer	Probe Period (PrbPer) These bits sets the TadvPrd as follows: Values: 0x0 (TADP_PRB_0.6_0.9SEC): TadvPrd: 0.625 to 0.925 sec 0x1 (TADP_PRB_1.2_1.8SEC): TadvPrd: 1.25 to 1.85 sec 0x2 (TADP_PRB_1.9_2.6SEC): TadvPrd: 1.9 to 2.6 sec 0x3 (RESERVED): Reserved	0x0
3:2	R/W	PrbDelta	Probe Delta (PrbDelta) These bits set the resolution for RTIM value. The bits are defined in units of 32 kHz clock cycles as follows: For example, if this value is chosen to 2'b01, it means that RTIM increments for every three 32Khz clock cycles. Values: 0x0 (VALUE1): 1 cycle 0x1 (VALUE2): 2 cycles 0x2 (VALUE3): 3 cycles 0x3 (VALUE4): 4 cycles	0x0
1:0	R/W	PrbDschg	Probe Discharge (PrbDschg) These bits set the times for TADP_DSCHG. These bits are defined as follows: Values: 0x0 (VALUE1): Probe Discharge time: 4 msec 0x1 (VALUE2): Probe Discharge time: 8 msec 0x2 (VALUE3): Probe Discharge time: 16 msec 0x3 (VALUE4): Probe Discharge time: 32 msec	0x0

GREFCLK

Register to control the ref_clk properties for the controller.

Offset:0x64 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:15	R/W	REFCLKPER	This bit indicates the period of ref_clk in terms of pico seconds. Note: The default value of this field is 'd0. The period of ref_clk must be an integer multiple of 125us. The minimum frequency supported is 12MHz. Other supported frequencies are 16, 17, 19.2, 20, 24, 30, and 40MHz.	0x0
14	R/W	RefclkMode	This bit is used to enable or disable ref_clk mode of operation. Values: 0x1 (ENABLE): Controller uses ref_clk to run internal micro- frame timers 0x0 (DISABLE): Controller uses phy_clk to run internal micro- frame timers	0x0
13:10	R	RESERVED	RESERVED	0x0
9:0	R/W	SOF_CNT_WKUP_ALERT	This bit indicates the number of SOF's after which the controller must generate an interrupt if the device had been in L1 state until that period. The interrupt is used by software to initiate remote wakeup in the controller in order to sync to the uF number in the host. For more details, refer to the "Auto Remote Wakeup Feature" section of the Programming Guide.	0x0

GINTMSK2

This register works with the Interrupt Register (GINTSTS2) to interrupt the application. When an interrupt bit is masked, the interrupt associated with that bit is not generated. However, the GINTSTS2 register bit corresponding to that interrupt is still set.

Offset:0x68 Total Reset Value:0x0000_0000 RESERVED

GINTSTS2

This register interrupts the application for system-level events in the current mode (Device mode or Host mode).

Offset:0x6c Total Reset Value:0x0000_0000 RESERVED

HPTXFSIZ

This register holds the size and the memory start address of the Periodic TxFIFO.

Offset:0x100 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:16			Reserved Field: Yes	
15:14	PTxFSize	R/W	Host Periodic TxFIFO Depth (PTxFSize) This value is in terms of 32-bit words. Minimum value is 16 Maximum value is 32,768	0x0
13:0			Reserved Field: Yes	

DPTXFSIZ_i (for $i = 1; i \leq \text{OTG\_NUM\_PERIO\_EPS}$)

This register is valid only in shared FIFO operation (OTG_EN_DED_TX_FIFO = 0).

Offset:0x104 + (i-1)*004 Total Reset Value:0x0300_0800

Bits	Access	Name	Description	Reset
31:16	R	DPTxFSize	Device Periodic TxFIFO Size (DPTxFSize) This value is in terms of 32-bit words. Minimum value is 4 Maximum value is 768 The value of this register is the Largest Device Mode Periodic Tx Data FIFO Depth i set during coreConsultant configuration, where i is the FIFO number this register corresponds to.	0x300
15:12			Reserved Field: Yes	
11:0	R/W	DPTxFStAddr	Device Periodic TxFIFO RAM Start Address (DPTxFStAddr) This register field holds the start address in the RAM for the periodic FIFO that this register corresponds to. The power-on reset value of this register is the sum of the Largest Rx Data FIFO Depth, Largest Non-periodic Tx Data FIFO Depth, and all lower numbered Largest Device Mode Periodic Tx Data FIFOs Depth specified during coreConsultant configuration. Only for the first FIFO (DPTXFSIZ ₁), the start address formula is: OTG_RX_DFIFO_DEPTH + OTG_TX_NPERIO_DFIFO_DEPTH. If it is a higher number FIFO (DPTXFSIZ ₂ , ...) the formula is: OTG_RX_DFIFO_DEPTH + OTG_TX_NPERIO_DFIFO_DEPTH + SUM of OTG_TX_DPERIO_DFIFO_DEPTH_ 'n', where $n=1$ to (FIFO number-1). If at POR, the calculated value (C) exceeds 65535, then the Reset value becomes Reset Value(A) = (C-65536). If Enable Dynamic FIFO Sizing is selected in coreConsultant (OTG_DFIFO_DYNAMIC = 1), and you have calculated/programmed new values for the Rx FIFO Non- periodic Tx FIFO, or device Periodic Tx FIFOs, you can program their sum according to the above formula.	0x800

DIEPTXFi (for $i = 1; i \leq \text{OTG\_NUM\_IN\_EPS}-1$)

This register holds the size and the memory start address of the Periodic TxFIFO. Offset: 0x104 + (i-1)*004 Total Reset

Value:0x0000_0200

Bits	Access	Name	Description	Reset
31:25			Reserved Field: Yes	
24:16	R/W	INEPnTxFDep	IN Endpoint TxFIFO Depth (INEPnTxFDep) This value is in terms of 32-bit words. Minimum value is 16 Maximum value is 32,768 The power-on reset value of this register is specified as the Largest IN Endpoint FIFO number Depth (parameter OTG_TX_DINEP_DFIFO_DEPTH_i) set during coreConsultant configuration, where i is the FIFO number this register corresponds to. If "Enable Dynamic FIFO Sizing?" was deselected in coreConsultant (parameter OTG_DFIFO_DYNAMIC = 0), these flops are optimized, and reads return the power-on value. If "Enable Dynamic FIFO Sizing?" was selected in coreConsultant (parameter OTG_DFIFO_DYNAMIC = 1), you can write a new value in this field. Programmed values must not exceed the power-on value.	0x100
15:11			Reserved Field: Yes	
10:0	R/W	INEPnTxFStAddr	IN Endpoint FIFO Transmit RAM Start Address (INEPnTxFStAddr) This field contains the memory start address for the IN endpoint Transmit FIFO that this register corresponds to. The power-on reset value of this register is calculated according to the following formula: OTG_RX_DFIFO_DEPTH + SUM of OTG_TX_DINEP_DFIFO_DEPTH_'n', where n = 0 to (FIFO number-1). For example, Start address of INEP FIFO 1 (DIEPTXF1) is OTG_RX_DFIFO_DEPTH + OTG_TX_DINEP_DFIFO_DEPTH_0, start address of INEP FIFO 2 (DIEPTXF2) is OTG_RX_DFIFO_DEPTH + OTG_TX_DINEP_DFIFO_DEPTH_0 + OTG_TX_DINEP_DFIFO_DEPTH_1, and so on. If at POR the calculated value (C) exceeds 65535, then the Reset value becomes Reset Value(A) = (C-65536).	0x500

HCFCG

This register is valid only in dedicated FIFO mode (OTG_EN_DED_TX_FIFO=1). It holds the size and memory start address of IN endpoint TxFIFOs implemented in Device mode. Each FIFO holds the data for one IN endpoint. This register is repeated for instantiated IN endpoint FIFOs 1 to 15. For IN endpoint FIFO 0, use GNPTXFSIZ register for programming the size and memory start address.

Offset:0x104 + (i-1)*004 Total Reset Value:0x0000_ea60

Bits	Access	Name	Description	Reset

Bits	Access	Name	Description	Reset
31	R/W	ModeChTimEn	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. Values: 0x0 (ENABLED): The Host core waits for either 200 PHY clock cycles or a linestate of SE0 at the end of resume to change the opmode from 0x2 to 0x0 0x1 (DISABLED): The Host core waits only for a linestate of SE0 at the end of resume to change the opmode from 0x2 to 0x0	0x0
30:27	R	RESERVED	RESERVED	0x0
26	R/W	PerSchedEna	Enable Periodic Scheduling (PerSchedEna): Applicable in host DDMA mode only. Enables periodic scheduling within the core. Initially, the bit is reset. 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 again 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. Values: 0x0 (DISABLED): Disables periodic scheduling within the core 0x1 (ENABLED): Enables periodic scheduling within the core	0x0
25:24	R/W	FrListEn	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: 8 Entries 2'b01: 16 Entries 2'b10: 32 Entries 2'b11: 64 Entries	0x0

Bits	Access	Name	Description	Reset
23	R/W	DescDMA	Enable Scatter/gather DMA in Host mode (DescDMA) When the Scatter/Gather DMA option selected during configuration of the RTL, the application can set this bit during initialization to enable the Scatter/Gather DMA operation. Values: 0x0 (DISABLED): Scatter/Gather DMA in Host mode is disabled 0x1 (ENABLED): Scatter/Gather DMA selected	0x0
22:17	R	RESERVED1	RESERVED	0x0
16	R/W	dis_tx_ipgap_dly_check	Disable Linestate check for Token to Token/Data interpacket gap delay calculation (dis_tx_ipgap_dly_check) Values: 0x0 (DISABLE): Enable Linestate check 0x1 (ENABLE): Disable Linestate check	0x0
15:8	R/W	ResValid	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 for 'ResValid' number of clock cycles to detect a valid resume when this is set.	0x2
7	R/W	Ena32KHzS	Enable 32 KHz Suspend mode (Ena32KHzS) This bit can be set only in FS PHY interface is selected. Else, 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 during Suspend is switched from 48 MHz to 32 KHz. Values: 0x0 (DISABLED): 32 KHz Suspend mode disabled 0x1 (ENABLED): 32 KHz Suspend mode enabled	0x0
6:3	R	RESERVED2	RESERVED	0x0
2	R/W	FSLSSupp	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. Values: 0x0 (HSFSLS): HS/FS/LS, based on the maximum speed supported by the connected device 0x1 (FSLS): FS/LS-only, even if the connected device can support HS	0x0

Bits	Access	Name	Description	Reset
1:0	R/W	FSLSPclkSel	FS/LS PHY Clock Select (FSLSPclkSel) When the core is in FS Host mode Values: 0x0 (CLK3060): PHY clock is running at 30/60 MHz 0x1 (CLK48): PHY clock is running at 48 MHz 0x2 (CLK6): PHY clock is running at 6 MHz	0x0

HFIR

This register is used to configure the controller in Host mode.

Offset:0x400 Total Reset Value:0x0000_3fff

Bits	Access	Name	Description	Reset
31:17	R	RESERVED	RESERVED	0x0
16	R/W	HFIRRldCtrl	Reload Control (HFIRRldCtrl) This bit allows dynamic reloading of the HFIR register during run time. Values: 0x0 (DISABLED): The HFIR cannot be reloaded dynamically 0x1 (ENABLED): The HFIR can be dynamically reloaded during runtime	0x0
15:0	R/W	FrInt	Frame Interval (FrInt) 125 us (PHY clock frequency for HS) 1 ms (PHY clock frequency for FS/LS)	0xea60

HFNUM

This register is used to control the interval between two consecutive SOFs.

Offset:0x404 Total Reset Value:0x0000_8400

Bits	Access	Name	Description	Reset
31:16	R	FrRem	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.	0x0
15:0	R	FrNum	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. Values: 0x0 (INACTIVE): No SOF is transmitted 0x1 (ACTIVE): SOF is transmitted	0x3fff

HPTXSTS

This register indicates the current frame number. It also indicates the time remaining (in terms of the number of PHY clocks) in the current (micro)frame.

Offset:0x408 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset

Bits	Access	Name	Description	Reset
31:23	R	PTxQTop	Top of the Periodic Transmit Request Queue (PTxQTop) This indicates the Entry in the Periodic Tx Request Queue that is currently being processed 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]: Last Periodic Entry for the selected periodic channel/endpoint Bit [23]: Terminate (last Entry for the selected channel/endpoint)	0x0
22:16	R	PTxQSpcAvail	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. Values: 0x0 (FULL): Periodic Transmit Request Queue is full 0x1 (FREE1): 1 location available 0x2 (FREE2): 2 locations available 0x3 (FREE3): 3 locations available 0x4 (FREE4): 4 locations available 0x5 (FREE5): 5 locations available 0x6 (FREE6): 6 locations available 0x7 (FREE7): 7 locations available 0x8 (FREE8): 8 locations available 0x9 (FREE9): 9 locations available 0xa (FREE10): 10 locations available 0xb (FREE11): 11 locations available 0xc (FREE12): 12 locations available 0xd (FREE13): 13 locations available 0xe (FREE14): 14 locations available 0xf (FREE15): 15 locations available	0x8
15:0	R	PTxFSpAvail	Periodic Transmit Data FIFO Space Available (PTxFSpAvail) Indicates the number of free locations available to be written to in the Periodic Tx FIFO. Values are in terms of 32-bit words 16'h0 : Periodic Tx FIFO is full 16'h1 : 1 word available 16'h2 : 2 words available 16'h_n : n words available (where 0 ≤ n ≤ 32,768) 16'h8000 : 32,768 words Others : Reserved	0x400

HAINT

This register contains information about the Periodic Transmit Queue in the Host controller.

Offset:0x410 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
x:0	R	HAINT	Channel Interrupt for channel no. Values: 0x0 (INACTIVE): Not active 0x1 (ACTIVE): Host Channel Interrupt	0x0

HAINTMSK

When a significant event occurs on a channel, the Host All Channels Interrupt register interrupts the application using the Host Channels Interrupt bit of the Core Interrupt register (GINTSTS.HChInt). This is shown in the "Interrupt Hierarchy" figure in the databook. There is one interrupt bit per channel, up to a maximum of 16 bits. Bits in this register are set and cleared when the application sets and clears bits in the corresponding Host Channel-n Interrupt register.

Offset:0x414 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
x:0	R/W	HAINTMsk	Channel Interrupt Mask (HAINTMsk) One bit per channel: Bit 0 for channel 0, bit 15 for channel 15 Values: 0x0 (UNMASK): Unmask Channel interrupt 0x1 (MASK): Mask Channel interrupt	0x0

HFLBAddr

The Host All Channel Interrupt Mask register works with the Host All Channel Interrupt register to interrupt the application when an event occurs on a channel. There is one interrupt mask bit per channel, up to a maximum of 16 bits.

Offset:0x418 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:0	R/W	HFLBAddr	The starting address of the Frame list. This register is used only for Isochronous and Interrupt Channels.	0x0

HPRT

This register is present only in case of Scatter/Gather DMA. It is implemented as flops. This register holds the starting address of the Frame list information.

Offset:0x41c Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:19			Reserved Field: Yes	
18:17	R	PrtSpd	Port Speed (PrtSpd) Indicates the speed of the device attached to this port. Values: 0x0 (HIGHSPD): High speed 0x1 (FULLSPD): Full speed 0x2 (LOWSPD): Low speed 0x3 (RESERVED): Reserved	0x0

Bits	Access	Name	Description	Reset
16:13	R/W	PrtTstCtl	Port Test Control (PrtTstCtl) The application writes a nonzero value to this field to put the port into a Test mode, and the corresponding pattern is signaled on the port. Values: 0x0 (DISABLED): Test mode disabled 0x1 (TESTJ): Test_J mode 0x2 (TESTK): Test_K mode 0x3 (TESTSN): Test_SE0_NAK mode 0x4 (TESTPM): Test_Packet mode 0x5 (TESTFENB): Test_force_Enable	0x0
12	R/W	PrtPwr	Port Power (PrtPwr) Values: 0x0 (OFF): Power off 0x1 (ON): Power on	0x0
11:10	R	PrtLnSts	Port Line Status (PrtLnSts) Indicates the current logic level USB data lines Values: 0x1 (PLUSD): Logic level of D+ 0x2 (MINUSD): Logic level of D-	0x0
9	R	RESERVED	RESERVED	0x0
8	R/W	PrtRst	Port Reset (PrtRst) When the application sets this bit, a reset sequence is started on this port. The application must time the reset period and clear this bit after the reset sequence is complete. Values: 0x0 (DISABLED): Port not in reset 0x1 (ENABLED): Port in reset	0x0
7	R/W1S	PrtSusp	Port Suspend (PrtSusp) The application sets this bit to put this port in Suspend mode. The core only stops sending SOFs when this is Set. To stop the PHY clock, the application must Set the Port Clock Stop bit, which asserts the suspend input pin of the PHY. The read value of this bit reflects the current suspend status of the port. This bit is cleared by the core after a remote wakeup signal is detected or the application sets the Port Reset bit or Port Resume bit in this register or the Resume/Remote Wakeup Detected Interrupt bit or Disconnect Detected Interrupt bit in the Core Interrupt register (GINTSTS.WkUpInt or GINTSTS.DisconnInt, respectively). This bit is cleared by the core even if there is no device connected to the Host. Values: 0x0 (INACTIVE): Port not in Suspend mode 0x1 (ACTIVE): Port in Suspend mode	0x0

Bits	Access	Name	Description	Reset
6	R/W	PrtRes	Port Resume (PrtRes) The application sets this bit to drive resume signaling on the port. The core continues to drive the resume signal until the application clears this bit. Values: 0x0 (NORESUME): No resume driven 0x1 (RESUME): Resume driven	0x0
5	R/W1C	PrtOvrCurrChng	Port Overcurrent Change (PrtOvrCurrChng) The core sets this bit when the status of the Port Overcurrent Active bit (bit 4) in this register changes. This bit can be set only by the core and the application must write 1 to clear it Values: 0x0 (INACTIVE): Status of port overcurrent status is not changed 0x1 (ACTIVE): Status of port overcurrent changed	0x0
4	R	PrtOvrCurrAct	Port Overcurrent Active (PrtOvrCurrAct) Indicates the overcurrent condition of the port. Values: 0x0 (INACTIVE): No overcurrent condition 0x1 (ACTIVE): Overcurrent condition	0x0
3	R/W1C	PrtEnChng	Port Enable/Disable Change (PrtEnChng) Values: 0x0 (INACTIVE): Port Enable bit 2 has not changed 0x1 (ACTIVE): Port Enable bit 2 changed	0x0
2	R/W1C	PrtEna	Port Enable (PrtEna) Values: 0x0 (DISABLED): Port disabled 0x1 (ENABLED): Port enabled	0x0
1	R/W1C	PrtConnDet	Port Connect Detected (PrtConnDet) Values: 0x1 (ACTIVE): Device connection detected 0x0 (INACTIVE): No device connection detected	0x0
0	R	PrtConnSts	Port Connect Status (PrtConnSts) Values: 0x0 (NOTATTACHED): No device is attached to the port 0x1 (ATTACHED): A device is attached to the port	0x0

HCCHARi(for i = 0; i <= OTG_NUM_HOST_CHAN-1)

This register is available only in Host mode. Currently, the OTG Host supports only one port. A single register holds USB port-related information such as USB reset, enable, suspend, resume, connect status, and test mode for each port. It is shown in the "Interrupt Hierarchy" figure in the databook. The R_SS_WC bits in this register can trigger an interrupt to the application through the Host Port Interrupt bit of the Core Interrupt register (GINTSTS.PrtInt). On a Port Interrupt, the application must read this register and clear the bit that caused the interrupt. For the R_SS_WC bits, the application must write a 1 to the bit to clear the interrupt.

Offset:0x440 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset

Bits	Access	Name	Description	Reset
31	R/W1S	ChEna	Channel Enable (ChEna) Values: 0x0 (DISABLED): If Scatter/Gather mode is enabled, indicates that the descriptor structure is not yet ready. If Scatter/Gather mode is disabled, indicates that the channel is disabled. 0x1 (ENABLED): If Scatter/Gather mode is enabled, indicates that the descriptor structure and data buffer with data is set up and this channel can access the descriptor. If Scatter/Gather mode is disabled, indicates that the channel is enabled.	0x0
30	R/W1S	ChDis	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. Values: 0x0 (INACTIVE): Transmit/Receive normal 0x1 (ACTIVE): Stop transmitting/receiving data on channel	0x0
29	R/W	OddFrm	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. Values: 0x0 (EFRAME): Even Frame Transfer 0x1 (OFRAME): Odd Frame Transfer	0x0
28:22	R/W	DevAddr	Device Address (DevAddr) This field selects the specific device serving as the data source or sink.	0x0
21:20	R/W	EC	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. Values: 0x0 (RESERVED): Reserved. This field yields undefined result 0x1 (TRANSONE): 1 transaction 0x2 (TRANSTWO): 2 transactions to be issued for this endpoint per microframe 0x3 (TRANSTHREE): 3 transactions to be issued for this endpoint per microframe	0x0

Bits	Access	Name	Description	Reset
19:18	R/W	EPTYPE	Endpoint Type (EPTYPE) Indicates the transfer type selected. Values: 0x0 (CTRL): Control 0x1 (ISOC): Isochronous 0x2 (BULK): Bulk 0x3 (INTERR): Interrupt	0x0
17	R/W	LSPDDEV	Low-Speed Device (LSPDDEV) This field is Set by the application to indicate that this channel is communicating to a low-speed device. Values: 0x0 (DISABLED): Not Communicating with low speed device 0x1 (ENABLED): Communicating with low speed device	0x0
16	R	RESERVED	RESERVED	0x0
15	R/W	EPDIR	Endpoint Direction (EPDIR) Indicates whether the transaction is IN or OUT. Values: 0x0 (OUT): OUT Direction 0x1 (IN): IN Direction	0x0
14:11	R/W	EPNUM	Endpoint Number (EPNUM) Indicates the endpoint number on the device serving as the data source or sink. Values: 0x0 (ENDPT0): End point 0 0x1 (ENDPT1): End point 1 0x2 (ENDPT2): End point 2 0x3 (ENDPT3): End point 3 0x4 (ENDPT4): End point 4 0x5 (ENDPT5): End point 5 0x6 (ENDPT6): End point 6 0x7 (ENDPT7): End point 7 0x8 (ENDPT8): End point 8 0x9 (ENDPT9): End point 9 0xa (ENDPT10): End point 10 0xb (ENDPT11): End point 11 0xc (ENDPT12): End point 12 0xd (ENDPT13): End point 13 0xe (ENDPT14): End point 14 0xf (ENDPT15): End point 15	0x0
10:0	R/W	MPS	Maximum Packet Size (MPS) Indicates the maximum packet size of the associated endpoint.	0x0

HCSPLITi(for i = 0; i <= OTG_NUM_HOST_CHAN-1)

This register contains the Split characteristics of the Host Channel.

Offset: 0x504 + i*20 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
31	R/W	SpltEna	Split Enable (SpltEna) The application sets this field to indicate that this channel is enabled to perform split transactions. Values: 0x0 (DISABLED): Split not enabled 0x1 (ENABLED): Split enabled	0x0
30:17	R	RESERVED	RESERVED	0x0
16	R/W	CompSplt	Do Complete Split (CompSplt) The application sets this field to request the OTG host to perform a complete split transaction. Values: 0x0 (NOSPLIT): No complete split transaction 0x1 (SPLIT): Complete Split transaction	0x0
15:14	R/W	XactPos	Transaction Position (XactPos) This field is used to determine whether to send all, first, middle, or last payloads with each OUT transaction. Values: 0x0 (MIDDLE): Mid. This is the middle payload of this transaction (which is larger than 188 bytes) 0x1 (END): End. This is the last payload of this transaction (which is larger than 188 bytes) 0x2 (BEGIN): Begin. This is the first data payload of this transaction (which is larger than 188 bytes) 0x3 (ALL): All. This is the entire data payload of this transaction (which is less than or equal to 188 bytes)	
13:7	R/W	HubAddr	Hub Address (HubAddr) This field holds the device address of the transaction translator's hub.	0x0
6:0	R/W	PrtAddr	Port Address (PrtAddr) This field is the port number of the recipient transaction translator.	

HCINTi

This register indicates the status of a channel with respect to USB- and AHB-related events. It is shown in the "Interrupt Hierarchy" figure in the databook. The application must read this register when the Host Channels Interrupt bit of the Core Interrupt register (GINTSTS.HChInt) is set. Before the application can read this register, it must first read the Host All Channels Interrupt (HAINT) register to get the exact channel number for the Host Channel-n Interrupt register. The application must clear the appropriate bit in this register to clear the corresponding bits in the HAINT and GINTSTS registers.

Offset: 0x508 + i*20 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
31:14	R	RESERVED	RESERVED	0x0

Bits	Access	Name	Description	Reset
13	R/W1C	DESC_LST_ROLLIntr	Descriptor rollover interrupt (DESC_LST_ROLLIntr) This bit is valid only when Scatter/Gather DMA mode is enabled. The core sets this bit when the corresponding channel's descriptor list rolls over. For non Scatter/Gather DMA mode, this bit is reserved. Values: 0x0 (INACTIVE): No Descriptor rollover interrupt 0x1 (ACTIVE): Descriptor rollover interrupt	0x0
12	R/W1C	XCS_XACT_ERR	Excessive Transaction Error (XCS_XACT_ERR) This bit is valid only when Scatter/Gather DMA mode is enabled. The core sets this bit when 3 consecutive transaction errors occurred on the USB bus. XCS_XACT_ERR is not generated for Isochronous channels. For non Scatter/Gather DMA mode, this bit is reserved. Values: 0x0 (INACTIVE): No Excessive Transaction Error 0x1 (ACTIVE): Excessive Transaction Error	0x0
11	R/W1C	BNAIntr	BNA (Buffer Not Available) Interrupt (BNAIntr) This bit is valid only when Scatter/Gather DMA mode is enabled. The core generates this interrupt when the descriptor accessed is not ready for the Core to process. BNA is not generated for Isochronous channels. For non Scatter/Gather DMA mode, this bit is reserved. Values: 0x0 (INACTIVE): No BNA Interrupt 0x1 (ACTIVE): BNA Interrupt	0x0
10	R/W1C	DataTglErr	Data Toggle Error (DataTglErr). This bit can be set only by the core and the application must write 1 to clear it. In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core. Values: 0x0 (INACTIVE): No Data Toggle Error 0x1 (ACTIVE): Data Toggle Error	0x0
9	R/W1C	FrmOvrn	Frame Overrun (FrmOvrn). In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core. This bit can be set only by the core and the application must write 1 to clear it. Values: 0x0 (INACTIVE): No Frame Overrun 0x1 (ACTIVE): Frame Overrun	0x0
8	R/W1C	BblErr	Babble Error (BblErr)	0x0

Bits	Access	Name	Description	Reset
7	R/W1C	XactErr	Transaction Error (XactErr) Indicates one of the following errors occurred on the USB. In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core. This bit can be set only by the core and the application must write 1 to clear it. Values: 0x0 (INACTIVE): No Transaction Error 0x1 (ACTIVE): Transaction Error	0x0
6	R/W1C	NYET	NYET Response Received Interrupt (NYET) In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core. This bit can be set only by the core and the application must write 1 to clear it. Values: 0x0 (INACTIVE): No NYET Response Received Interrupt 0x1 (ACTIVE): NYET Response Received Interrupt	0x0
5	R/W1C	ACK	ACK Response Received/Transmitted Interrupt (ACK) In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core. This bit can be set only by the core and the application must write 1 to clear it. Values: 0x0 (INACTIVE): No ACK Response Received or Transmitted Interrupt 0x1 (ACTIVE): ACK Response Received or Transmitted Interrupt	0x0
4	R/W1C	NAK	NAK Response Received Interrupt (NAK) In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core. This bit can be set only by the core and the application must write 1 to clear it. Values: 0x0 (INACTIVE): No NAK Response Received Interrupt 0x1 (ACTIVE): NAK Response Received Interrupt	0x0
3	R/W1C	STALL	STALL Response Received Interrupt (STALL) In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core. This bit can be set only by the core and the application must write 1 to clear it. Values: 0x0 (INACTIVE): No Stall Response Received Interrupt 0x1 (ACTIVE): Stall Response Received Interrupt	0x0

Bits	Access	Name	Description	Reset
2	R/W1C	AHBErr	AHB Error (AHBErr) This is generated only in Internal DMA mode when there is an AHB error during AHB read/write. The application can read the corresponding channel's DMA address register to get the error address. For details, see "AHB Error Handling" in the Programming Guide. Values: 0x0 (INACTIVE): No AHB error 0x1 (ACTIVE): AHB error during AHB read/write	0x0
1	R/W1C	ChHltd	Channel Halted (ChHltd) In non Scatter/Gather DMA mode, it indicates the transfer completed abnormally either because of any USB transaction error or in response to disable request by the application or because of a completed transfer. Values: 0x0 (INACTIVE): Channel not halted 0x1 (ACTIVE): Channel Halted	0x0
0	R/W1C	XferCompl	Transfer Completed (XferCompl) Transfer completed normally without any errors. This bit can be set only by the core and the application must write 1 to clear it. Values: 0x0 (INACTIVE): Transfer in progress or No Active Transfer 0x1 (ACTIVE): Transfer completed normally without any errors	0x0

HCINTMSK_i

This register reflects the mask for each channel status described in the previous section.

Offset: 0x50C + i*20 Total Reset Value: 0x0000_0000 Bits Access Name Description Reset

Bits	Access	Name	Description	Reset
31:14	R	RESERVED	RESERVED	0x0
13	R/W	DESC_LST_ROLLIntrMsk	Descriptor List rollover interrupt Mask register (DESC_LST_ROLLIntrMsk) This bit is valid only when Scatter/Gather DMA mode is enabled. Values: 0x0 (MASK): Descriptor Rollover Interrupt Mask 0x1 (NOMASK): Descriptor Rollover Interrupt not masked	0x0
12	R	RESERVED1	RESERVED	0x0
11	R/W	BNAIntrMsk	BNA (Buffer Not Available) Interrupt mask register (BNAIntrMsk) This bit is valid only when Scatter/Gather DMA mode is enabled. Values: 0x0 (MASK): BNA Interrupt Masked 0x1 (NOMASK): BNA Interrupt not masked	0x0
10:3	R	RESERVED	RESERVED	0x0

Bits	Access	Name	Description	Reset
2	R/W	AHBErrMsk	AHB Error Mask (AHBErrMsk) In scatter/gather DMA mode for host, interrupts are not generated due to the corresponding bits set in HCINTn. Values: 0x0 (MASK): AHB Error Mask 0x1 (NOMASK): No AHB Error Mask	0x0
1	R/W	ChHltdMsk	Channel Halted Mask (ChHltdMsk) Values: 0x0 (MASK): Channel Halted Mask 0x1 (NOMASK): No Channel Halted Mask	0x0
0	R/W	XferComplMsk	Transfer Completed Mask (XferComplMsk) Values: 0x0 (MASK): Transfer Completed Mask 0x1 (NOMASK): No Transfer Completed Mask	0x0

HCTSIZEi

This register reflects the transfer size for the Host Channel.

Offset:0x510 + i*20 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31	R/W	DoPng	Do Ping (DoPng) This bit is used only for OUT transfers. Setting this field to 1 directs the host to do PING protocol. Note: Do not set this bit for IN transfers. If this bit is set for IN transfers it disables the channel. Values: 0x0 (NOPING): No ping protocol 0x1 (PING): Ping protocol	0x0
30:29	R/W	Pid	PID (Pid) The application programs this field with the type of PID to use for the initial transaction. The host maintains this field for the rest of the transfer. Values: 0x0 (DATA0): DATA0 0x1 (DATA2): DATA2 0x2 (DATA1): DATA1 0x3 (MDATA): MDATA (non-control)/SETUP (control)	0x0
28:19	R/W	PktCnt	Non-Scatter/Gather DMA Mode: Packet Count (PktCnt) This field is programmed by the application with the expected number of packets to be transmitted (OUT) or received (IN). The host decrements this count on every successful transmission or reception of an OUT/IN packet. Once this count reaches zero, the application is interrupted to indicate normal completion. Scatter/Gather DMA Mode: [28:19]: Reserved	0x0

Bits	Access	Name	Description	Reset
18:0	R/W	XferSize	Non-Scatter/Gather DMA Mode: Transfer Size (XferSize) For an OUT, this field is the number of data bytes the host sends during the transfer. For an IN, this field is the buffer size that the application has Reserved for the transfer. The application is expected to program this field as an integer multiple of the maximum packet size for IN transactions (periodic and non-periodic). This field indicates the number of descriptors present in that list microframe. The possible values for FS are 1: 2 descriptors 3: 4 descriptors 7: 8 descriptors 15: 16 descriptors 31: 32 descriptors 63: 64 descriptors The possible values for HS are 7: 8 descriptors 15: 16 descriptors 31: 32 descriptors 63: 64 descriptors 127: 128 descriptors 255: 256 descriptors [7:0]: SCHED_INFO (Schedule information)	0x0

HCDMAi

This register is used by the OTG host in the internal DMA mode to maintain the current buffer pointer for IN/OUT transactions. The starting DMA address must be DWORD-aligned.

Offset:0x514 + i*20 Total Reset Value:0x000b_0000

Bits	Access	Name	Description	Reset

Bits	Access	Name	Description	Reset
31:0	DMAAddr	R/W	In Scatter-Gather DMA (DescDMA) Mode for Non-Isochronous: [31:9]: DMA Address (DMAAddr) The start address must be 512-bytes aligned. This field holds the start address of the 512 bytes page. The first descriptor in the list must be located in this address. The first descriptor may be or may not be ready. The core starts processing the list from the CTD value. [8:3]: Current Transfer Desc(CTD) This value is in terms of number of descriptors. The values can be from 0 to 63. 0 - 1 descriptor. 63 - 64 descriptors. This field indicates the current descriptor processed in the list. This field is updated both by application and the core. For example, if the application enables the channel after programming CTD=5, then the core starts processing the sixth descriptor. The address is obtained by adding a value of (8bytes5=) 40(decimal) to DMAAddr. Reset: 6'h0 [2:0]: Reserved In Scatter-Gather DMA (DescDMA) Mode for Isochronous: [31:N]: DMA Address (DMAAddr) The start address must be 512-bytes aligned. This field holds the address of the 2(nTD+1) bytes of locations in which the isochronous descriptors are present where N is based on nTD as follows: [31:N]: Base Address [N-1:3]: Offset [2:0]: 000 For HS ISOC, if nTD is, 7, N=6 15, N=7 31, N=8 63, N=9 127, N=10 255, N=11 For FS ISOC, if nTD is, 1, N=4 3, N=5 7, N=6 15, N=7 31, N=8 63, N=9 [N-1:3]: Current Transfer Desc(CTD) CTD for isochronous is based on the current frame/(micro)frame value. Need to be set to zero by application.	0x0

HCDMABi

This register is present only in case of Scatter/Gather DMA. It is implemented in RAM instead of flop-based implementation. This register holds the current buffer address. Offset: 0x51C + i*20 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:0	R	HCDMAB	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.	0x0

DCFG

This register configures the core in Device mode after power-on or after certain control commands or enumeration. Do not make changes to this register after initial programming.

Offset:0x800 Total Reset Value:0x0422_0000

Bits	Access	Name	Description	Reset
31:26	R/W	ResValid	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
25:24	R/W	PerSchIntvl	Periodic Scheduling Interval (PerSchIntvl) PerSchIntvl must be programmed 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. Values: 0x0 (MF25): 25% of (micro)Frame 0x1 (MF50): 50% of (micro)Frame 0x2 (MF75): 75% of (micro)Frame 0x3 (RESERVED): Reserved	0x0
23	R/W	DescDMA	Enable Scatter/gather DMA in device mode (DescDMA). When the Scatter/Gather DMA option 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: Values: 0x0 (DISABLED): Disable Scatter/Gather DMA 0x1 (ENABLED): Enable Scatter/Gather DMA	0x0
22:18	R/W	RESERVED	RESERVED	0x8

Bits	Access	Name	Description	Reset
17	R/W	ipgisocSupt	Worst-Case Inter-Packet Gap ISOC OUT Support (ipgisocSupt) This bit indicates that the controller supports the worst-case scenario of Rx followed by Rx Inter-Packet Gap (IPG) (32 bit times) as per the UTMI Specification for any token following an ISOC OUT token. Without this support, when any token follows an ISOC OUT token with the worst-case IPG, the controller does not detect the followed token. The worst-case IPG of the controller without this support depends on the AHB and PHY clock frequency. Values: 0x0 (DISABLED): Worst-Case Inter-Packet Gap ISOC OUT Support is disabled 0x1 (ENABLED): Worst-Case Inter-Packet Gap ISOC OUT Support is enabled	0x1
16	R	RESERVED	RESERVED	0x0
15	R/W	ErraticIntMsk	Erratic Error Interrupt Mask Values: 0x0 (NOMASK): Early suspend interrupt is generated on erratic error 0x1 (MASK): Mask early suspend interrupt on erratic error	0x0
14	R/W	XCVRDLY	XCVRDLY Enables or disables delay between xcvr_sel and txvalid during device chirp Values: 0x0 (DISABLE): No delay between xcvr_sel and txvalid during Device chirp 0x1 (ENABLE): Enable delay between xcvr_sel and txvalid during Device chirp	0x0
13	R/W	EnDevOutNak	Enable Device OUT NAK (EnDevOutNak) This bit enables setting NAK for Bulk OUT endpoints after the transfer is completed for Device mode Descriptor DMA Values: 0x0 (DISABLED): The core does not set NAK after Bulk OUT transfer complete 0x1 (ENABLED): The core sets NAK after Bulk OUT transfer complete	0x0
12:11	R/W	PerFrInt	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. Values: 0x0 (EOPF80): 80% of the (micro)Frame interval 0x1 (EOPF85): 85% of the (micro)Frame interval 0x2 (EOPF90): 90% of the (micro)Frame interval 0x3 (EOPF95): 95% of the (micro)Frame interval	0x0

Bits	Access	Name	Description	Reset
10:4	R/W	DevAddr	Device Address (DevAddr) The application must program this field after every SetAddress control command.	0x0
3	R/W	Ena32KHzSusp	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. Values: 0x0 (DISABLED): USB 1.1 Full-Speed Serial Transceiver not selected 0x1 (ENABLED): USB 1.1 Full-Speed Serial Transceiver Interface selected	0x0
2	R/W	NZStsOUTHShk	Non-Zero-Length Status OUT Handshake (NZStsOUTHShk) The application can use this field to select the handshake the core sends on receiving a nonzero-length data packet during the OUT transaction of a control transfer's Status stage. Values: 0x0 (SENDOUT): Send the received OUT packet to the application (zero-length or non-zero length) and send a handshake based on NAK and STALL bits for the endpoint in the Device Endpoint Control Register 0x1 (SENDSTALL): Send a STALL handshake on a nonzero-length status OUT transaction and do not send the received OUT packet to the application	0x0
1:0	R/W	DevSpd	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 connect sequence is completed, and is based on the speed of the USB host to which the core is connected. Values: 0x0 (USBHS20): High speed USB 2.0 PHY clock is 30 MHz or 60 MHz 0x1 (USBFS20): Full speed USB 2.0 PHY clock is 30 MHz or 60 MHz 0x2 (USBLS116): Low speed USB 1.1 transceiver clock is 6 MHz 0x3 (USBFS1148): Full speed USB 1.1 transceiver clock is 48 MHz	0x0

DCTL

This register is used to control the characteristics of the Device controller.

Offset:0x804 Total Reset Value:0x0000_0002

Bits	Access	Name	Description	Reset
31:30			Reserved Field: Yes	

Bits	Access	Name	Description	Reset
29:20	R	RESERVED	RESERVED	0x0
19	R/W	ServInt	Service Interval based scheduling for Isochronous IN Endpoints This field is used to enable service interval based scheduling feature for ISOC IN EPs. Note: This bit is applicable only in device mode and when Scatter/Gather DMA mode is used. This feature must not be enabled along with DCTL.IgnrFrmNum. Scatter/Gather DMA Mode (GAHB_CFG.DMAEn=1,DCFG.DescDMA=1): When this bit is enabled, the frame number field in the ISOC IN descriptor structure is interpreted as the last frame of the service interval. In Scatter/Gather DMA mode, if this bit is enabled, the pending packets are flushed by the controller at the last frame of the service interval. Values: 0x0 (DISABLED): The controller behavior depends on DCTL.IgnrFrmNum field. 0x1 (ENABLED): Scatter/Gather DMA Mode: The controller can transmit the packets in any frame of the service interval.	0x0
18	R/W	DeepSleepBESLReject	DeepSleepBESLReject Values: 0x0 (DISABLED): Deep Sleep BESL Reject feature is disabled 0x1 (ENABLED): Deep Sleep BESL Reject feature is enabled	0x0
17	R/W	EnContOnBNA	Enable Continue on BNA (EnContOnBNA) This bit enables the core to continue on BNA for Bulk 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. Values: 0x0 (DISABLED): Core disables the endpoint after receiving BNA interrupt. When application re-enables the endpoint, core starts processing from the DOEPDMA descriptor 0x1 (ENABLED): Core disables the endpoint after receiving BNA interrupt. When application re-enables the endpoint, core starts processing from the descriptor that received the BNA interrupt.	0x0
16	R/W	NakOnBble	NAK on Babble Error (NakOnBble) Set NAK automatically on babble (NakOnBble). The core sets NAK automatically for the endpoint on which babble is received. Values: 0x0 (DISABLED): Disable NAK on Babble Error 0x1 (ENABLED): NAK on Babble Error	0x0

Bits	Access	Name	Description	Reset
15	R/W	IgnrFrmNum	Ignore Frame Number Feature for Isochronous Endpoints (IgnrFrmNum) This field is also used to control the Periodic Transfer Interrupt (PTI) feature. Slave Mode (GAHBCFG.DMAEn=0): This bit is not valid in Slave mode and must not be programmed to 1. Scatter/Gather DMA Mode (GAHBCFG.DMAEn=1,DCFG.DescDMA=1): Note: When Scatter/Gather DMA mode is enabled this feature is not applicable to High Speed, High bandwidth transfers. When this bit is enabled, there must be only one packet per descriptor. 0: The core transmits the packets only in the frame number in which they are intended to be transmitted. 1: The core ignores the frame number, sending packets immediately as the packets are ready. In Scatter/Gather DMA mode, if this bit is enabled, the packets are not flushed when a ISOC IN token is received for an elapsed frame. Non-Scatter/Gather DMA Mode, that is, Buffer DMA Mode (GAHBCFG.DMAEn=1,DCFG.DescDMA=0): When Scatter/Gather DMA mode is disabled, this field is used by the application to enable Periodic Transfer Interrupt (PTI) Mode. The application can program Periodic Endpoint transfers for multiple (micro)Frames. 0: Periodic Transfer Interrupt feature is disabled, application needs to program transfers for periodic endpoints every (micro)Frame 1: Periodic Transfer Interrupt feature is enabled, application can program transfers for multiple (micro)Frames for periodic endpoints. In the PTI mode, the application receives Transfer Complete Interrupt after transfers for multiple (micro)Frames are completed. Values: 0x0 (DISABLED): Scatter/Gather DMA Mode: The core transmits the packets only in the frame number in which they are intended to be transmitted. Non-Scatter/Gather DMA Mode: Periodic Transfer Interrupt feature is disabled. 0x1 (ENABLED): Scatter/Gather DMA Mode: The core ignores the frame number, sending packets immediately as the packets are ready. Non-Scatter/Gather DMA Mode: Periodic Transfer Interrupt feature is enabled.	0x0

Bits	Access	Name	Description	Reset
14:13	R/W	GMC	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 endpoints. Values: 0x0 (NOTVALID): Invalid 0x1 (ONEPACKET): 1 packet 0x2 (TWOPACKET): 2 packets 0x3 (THREEPACKET): 3 packets	0x0
12		RESERVED	RESERVED	
11	R/W	PWROnPrgDone	Power-On Programming Done (PWROnPrgDone) The application uses this bit to indicate that register programming is completed after a wake-up from Power Down mode. Values: 0x0 (NOTDONE): Power-On Programming not done 0x1 (DONE): Power-On Programming Done	0x0
10	W	CGOUTNak	Clear Global OUT NAK (CGOUTNak) A write to this field clears the Global OUT NAK. Values: 0x0 (DISABLED): Disable Clear Global OUT NAK 0x1 (ENABLED): Clear Global OUT NAK	0x0
9	W	SGOUTNak	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. Values: 0x0 (DISABLED): Disable Global OUT NAK 0x1 (ENABLED): Set Global OUT NAK	0x0
8	W	CGNPInNak	Clear Global Non-periodic IN NAK (CGNPInNak) A write to this field clears the Global Non-periodic IN NAK. Values: 0x0 (DISABLE): Disable Global Non-periodic IN NAK 0x1 (ENABLE): Clear Global Non-periodic IN NAK	0x0

Bits	Access	Name	Description	Reset
7	W	SGNPInNak	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 non- periodic 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 Values: 0x0 (DISABLE): Disable Global Non-periodic IN NAK 0x1 (ENABLE): Set Global Non-periodic IN NAK	0x0
6:4	R/W	TstCtl	Test Control (TstCtl) Values: 0x0 (DISABLED): Test mode disabled 0x1 (TESTJ): Test_J mode 0x2 (TESTK): Test_K mode 0x3 (TESTSN): Test_SE0_NAK mode 0x4 (TESTPM): Test_Packet mode 0x5 (TESTFE): Test_force_Enable	0x0
3	R	GOUTNakSts	Global OUT NAK Status (GOUTNakSts) Values: 0x0 (INACTIVE): A handshake is sent based on the FIFO Status and the NAK and STALL bit settings. 0x1 (ACTIVE): 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.	0x0
2	R	GNPINNakSts	Global Non-periodic IN NAK Status (GNPINNakSts) Values: 0x0 (INACTIVE): A handshake is sent out based on the data availability in the transmit FIFO 0x1 (ACTIVE): A NAK handshake is sent out on all non-periodic IN endpoints, irrespective of the data availability in the transmit FIFO.	0x0
1	R/W	SftDiscon	Soft Disconnect (SftDiscon) The application uses this bit to signal the controller 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. Values:	0x1

Bits	Access	Name	Description	Reset
0	R/W	RmtWkUpSig	Remote Wakeup Signaling (RmtWkUpSig) When the application sets this bit, the core initiates remote signaling to wake up 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 microseconds (TL1DevDrvResume) after being set by the application. The application must not set this bit when GLPMCFG bRemoteWake from the previous LPM transaction is zero. Values: 0x0 (DISABLEDRMWKUP): Core does not send Remote Wakeup Signaling 0x1 (ENABLERMWKUP): Core sends Remote Wakeup Signaling	0x0

DSTS

This register indicates the status of the core with respect to USB-related events. It must be read on interrupts from Device All Interrupts (DAINT) register.

Offset:0x808 Total Reset Value:0x0000_0002

Bits	Access	Name	Description	Reset
31:24	R	RESERVED	RESERVED	0x0
23:22	R	DevLnSts	Device Line Status (DevLnSts) Indicates the current logic level USB data lines DevLnSts[1]: Logic level of D+ DevLnSts[0]: Logic level of D-	0x0
21:8	R	SOFFN	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.	0x0
7:4	R	RESERVED1	RESERVED	0x0
3	R	ErrticErr	Erratic Error (ErrticErr) Values: 0x0 (INACTIVE): No Erratic Error 0x1 (ACTIVE): Erratic Error	0x0

Bits	Access	Name	Description	Reset
2:1	R	EnumSpd	Enumerated Speed (EnumSpd) Indicates the speed at which the controller has come up after speed detection through a connect or reset sequence. Values: 0x0 (HS3060): High speed (PHY clock is running at 30 or 60 MHz) 0x1 (FS3060): Full speed (PHY clock is running at 30 or 60 MHz) 0x2 (LS6): Low speed (PHY clock is running at 6 MHz) 0x3 (FS48): Full speed (PHY clock is running at 48 MHz)	0x1
0	R	SuspSts	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 : Values: 0x0 (INACTIVE): No suspend state 0x1 (ACTIVE): Suspend state	0x0

DIEPMSK

This register works with each of the Device IN Endpoint Interrupt (DIEPINTn) registers for all endpoints to generate an interrupt per IN endpoint. The IN endpoint interrupt for a specific status in the DIEPINTn register can be masked by writing to the corresponding bit in this register. Status bits are masked by default. Offset: 0x810 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:14	R	RESERVED	RESERVED	0x0
13	R/W	NAKMsK	NAK interrupt Mask (NAKMsK) Values: 0x0 (MASK): Mask NAK Interrupt 0x1 (NOMASK): No Mask NAK Interrupt	0x0
12:10			Reserved Field: Yes	
9	R/W	BNAIntrMsK	BNA interrupt Mask (BNAIntrMsK)	0x0
8	R/W	TxfifoUndrnMsK	Fifo Underrun Mask (TxfifoUndrnMsK) Values: 0x0 (MASK): Mask Fifo Underrun Interrupt 0x1 (NOMASK): No Fifo Underrun Interrupt Mask	0x0
7	R	RESERVED1	RESERVED	0x0
6	R/W	INEPNakEffMsK	IN Endpoint NAK Effective Mask (INEPNakEffMsK) Values: 0x0 (MASK): Mask IN Endpoint NAK Effective Interrupt 0x1 (NOMASK): No IN Endpoint NAK Effective Interrupt Mask	0x0

Bits	Access	Name	Description	Reset
5	R/W	INTknEPMisMsk	IN Token received with EP Mismatch Mask (INTknEPMisMsk) Values: 0x0 (MASK): Mask IN Token received with EP Mismatch Interrupt 0x1 (NOMASK): No Mask IN Token received with EP Mismatch Interrupt	0x0
4	R/W	INTknTXFEmpMsk	IN Token Received When TxFIFO Empty Mask (INTknTXFEmpMsk) Values: 0x0 (MASK): Mask IN Token Received When TxFIFO Empty Interrupt 0x1 (NOMASK): No IN Token Received When TxFIFO Empty Interrupt	0x0
3	R/W	TimeOUTMsk	Timeout Condition Mask (TimeOUTMsk) (Non-isochronous endpoints) Values: 0x0 (MASK): Mask Timeout Condition Interrupt 0x1 (NOMASK): No Timeout Condition Interrupt Mask	0x0
2	R/W	AHBErrMsk	AHB Error Mask (AHBErrMsk) Values: 0x0 (MASK): Mask AHB Error Interrupt 0x1 (NOMASK): No AHB Error Interrupt Mask	0x0
1	R/W	EPDisbldMsk	Endpoint Disabled Interrupt Mask (EPDisbldMsk) Values: 0x0 (MASK): Mask Endpoint Disabled Interrupt 0x1 (NOMASK): No Endpoint Disabled Interrupt Mask	0x0
0	R/W	XferComplMsk	Transfer Completed Interrupt Mask (XferComplMsk) Values: 0x0 (MASK): Mask Transfer Completed Interrupt 0x1 (NOMASK): No Transfer Completed Interrupt Mask	0x0

DOEPMASK

This register works with each of the Device OUT Endpoint Interrupt (DOEPINTn) registers for all endpoints to generate an interrupt per OUT endpoint. The OUT endpoint interrupt for a specific status in the DOEPINTn register can be masked by writing into the corresponding bit in this register. Status bits are masked by default.

Offset:0x814 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:15	R	RESERVED	RESERVED	0x0
14	R/W	NYETMsk	NYET interrupt Mask (NYETMsk) Values: 0x0 (MASK): Mask NYET Interrupt 0x1 (NOMASK): No NYET Interrupt Mask	0x0

Bits	Access	Name	Description	Reset
13	R/W	NAKMsK	NAK interrupt Mask (NAKMsK) Values: 0x0 (MASK): Mask NAK Interrupt 0x1 (NOMASK): No NAK Interrupt Mask	0x0
12	R/W	BbleErrMsK	Babble Error interrupt Mask (BbleErrMsK) Values: 0x0 (MASK): Mask Babble Error Interrupt 0x1 (NOMASK): No Babble Error Interrupt Mask	0x0
11:10			Reserved Field: Yes	
9	R/W	BnaOutIntrMsK	BNA interrupt Mask (BnaOutIntrMsK) Values: 0x0 (MASK): Mask BNA Interrupt 0x1 (NOMASK): No BNA Interrupt Mask	0x0
8	R/W	OutPktErrMsK	OUT Packet Error Mask (OutPktErrMsK) Values: 0x0 (MASK): Mask OUT Packet Error Interrupt 0x1 (NOMASK): No OUT Packet Error Interrupt Mask	0x0
7	R	RESERVED1	RESERVED	0x0
6	R/W	Back2BackSETup	Back-to-Back SETUP Packets Received Mask (Back2BackSETup) Applies to control OUT endpoints only. Values: 0x0 (MASK): Mask Back-to-Back SETUP Packets Received Interrupt 0x1 (NOMASK): No Back-to-Back SETUP Packets Received Interrupt Mask	0x0
5	R/W	StsPhseRcvdMsK	Status Phase Received Mask (StsPhseRcvdMsK) Applies to control OUT endpoints only. Values: 0x0 (MASK): Status Phase Received Mask 0x1 (NOMASK): No Status Phase Received Mask	0x0
4	R/W	OUTTknEPdisMsK	OUT Token Received when Endpoint Disabled Mask (OUTTknEPdisMsK) Applies to control OUT endpoints only. Values: 0x0 (MASK): Mask OUT Token Received when Endpoint Disabled Interrupt 0x1 (NOMASK): No OUT Token Received when Endpoint Disabled Interrupt Mask	0x0
3	R/W	SetUPMsK	SETUP Phase Done Mask (SetUPMsK) Applies to control endpoints only. Values: 0x0 (MASK): Mask SETUP Phase Done Interrupt 0x1 (NOMASK): No SETUP Phase Done Interrupt Mask	0x0

Bits	Access	Name	Description	Reset
2	R/W	AHBErrMsk	AHB Error (AHBErrMsk) Values: 0x0 (MASK): Mask AHB Error Interrupt 0x1 (NOMASK): No AHB Error Interrupt Mask	0x0
1	R/W	EPDisbldMsk	Endpoint Disabled Interrupt Mask (EPDisbldMsk) Values: 0x0 (MASK): Mask Endpoint Disabled Interrupt 0x1 (NOMASK): No Endpoint Disabled Interrupt Mask	0x0
0	R/W	XferComplMsk	Transfer Completed Interrupt Mask (XferComplMsk) Values: 0x0 (MASK): Mask Transfer Completed Interrupt 0x1 (NOMASK): No Transfer Completed Interrupt Mask	0x0

DAINT

When a significant event occurs on an endpoint, a Device All Endpoints Interrupt register interrupts the application using the Device OUT Endpoints Interrupt bit or Device IN Endpoints Interrupt bit of the Core Interrupt register (GINTSTS.OEPInt or GINTSTS.IEPInt, respectively). This is shown in Figure 5-2. There is one interrupt bit per endpoint, up to a maximum of 16 bits for OUT endpoints and 16 bits for IN endpoints. For a bidirectional endpoint, the corresponding IN and OUT interrupt bits are used. Bits in this register are set and cleared when the application sets and clears bits in the corresponding Device Endpoint-n Interrupt register (DIEPINTn/DOEPINTn).

Offset:0x818 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:23	RESERVED	R	RESERVED	
22	R	OutEPInt6	OUT Endpoint 6 Interrupt Bit Values: 0x0 (INACTIVE): No Interrupt 0x1 (ACTIVE): Interrupt is active for the OUT EP	0x0
21	R	OutEPInt5	OUT Endpoint 5 Interrupt Bit Values: 0x0 (INACTIVE): No Interrupt 0x1 (ACTIVE): Interrupt is active for the OUT EP	0x0
20	R	OutEPInt4	OUT Endpoint 4 Interrupt Bit Values: 0x0 (INACTIVE): No Interrupt 0x1 (ACTIVE): Interrupt is active for the OUT EP	0x0
19	R	OutEPInt3	OUT Endpoint 3 Interrupt Bit Values: 0x0 (INACTIVE): No Interrupt 0x1 (ACTIVE): Interrupt is active for the OUT EP	0x0
18	R	OutEPInt2	OUT Endpoint 2 Interrupt Bit Values: 0x0 (INACTIVE): No Interrupt 0x1 (ACTIVE): Interrupt is active for the OUT EP	0x0

Bits	Access	Name	Description	Reset
17	R	OutEPInt1	OUT Endpoint 1 Interrupt Bit Values: 0x0 (INACTIVE): No Interrupt 0x1 (ACTIVE): Interrupt is active for the OUT EP	0x0
16	R	OutEPInt0	OUT Endpoint 0 Interrupt Bit Values: 0x0 (INACTIVE): No Interrupt 0x1 (ACTIVE): Interrupt is active for OUT EP0	0x0
15:7	RESERVED	R	RESERVED	
6	R	InEpInt6	IN Endpoint 6 Interrupt Bit Values: 0x0 (INACTIVE): No Interrupt 0x1 (ACTIVE): Interrupt is active for the IN EP	0x0
5	R	InEpInt5	IN Endpoint 5 Interrupt Bit Values: 0x0 (INACTIVE): No Interrupt 0x1 (ACTIVE): Interrupt is active for the IN EP	0x0
4	R	InEpInt4	IN Endpoint 4 Interrupt Bit Values: 0x0 (INACTIVE): No Interrupt 0x1 (ACTIVE): Interrupt is active for the IN EP	0x0
3	R	InEpInt3	IN Endpoint 3 Interrupt Bit Values: 0x0 (INACTIVE): No Interrupt 0x1 (ACTIVE): Interrupt is active for the IN EP	0x0
2	R	InEpInt2	IN Endpoint 2 Interrupt Bit Values: 0x0 (INACTIVE): No Interrupt 0x1 (ACTIVE): Interrupt is active for the IN EP	0x0
1	R	InEpInt1	IN Endpoint 1 Interrupt Bit Values: 0x0 (INACTIVE): No Interrupt 0x1 (ACTIVE): Interrupt is active for the IN EP	0x0
0	R	InEpInt0	IN Endpoint 0 Interrupt Bit Values: 0x0 (INACTIVE): No Interrupt 0x1 (ACTIVE): Interrupt is active for IN EP0	0x0

DAINTMSK

The Device Endpoint Interrupt Mask register works with the Device Endpoint Interrupt register to interrupt the application when an event occurs on a device endpoint. However, the Device All Endpoints Interrupt (DAINT) register bit corresponding to that interrupt is still set.

Offset:0x81c Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset

Bits	Access	Name	Description	Reset
31:23	R	RESERVED	RESERVED	0x0
22	R/W	OutEPMsk6	OUT Endpoint 6 Interrupt mask Bit Values: 0x0 (MASK): Mask OUT Endpoint Interrupt 0x1 (NOMASK): No Interrupt mask	0x0
21	R/W	OutEPMsk5	OUT Endpoint 5 Interrupt mask Bit Values: 0x0 (MASK): Mask OUT Endpoint Interrupt 0x1 (NOMASK): No Interrupt mask	0x0
20	R/W	OutEPMsk4	OUT Endpoint 4 Interrupt mask Bit Values: 0x0 (MASK): Mask OUT Endpoint Interrupt 0x1 (NOMASK): No Interrupt mask	0x0
19	R/W	OutEPMsk3	OUT Endpoint 3 Interrupt mask Bit Values: 0x0 (MASK): Mask OUT Endpoint Interrupt 0x1 (NOMASK): No Interrupt mask	0x0
18	R/W	OutEPMsk2	OUT Endpoint 2 Interrupt mask Bit Values: 0x0 (MASK): Mask OUT Endpoint Interrupt 0x1 (NOMASK): No Interrupt mask	0x0
17	R/W	OutEPMsk1	OUT Endpoint 1 Interrupt mask Bit Values: 0x0 (MASK): Mask OUT Endpoint Interrupt 0x1 (NOMASK): No Interrupt mask	0x0
16	R/W	OutEPMsk0	OUT Endpoint 0 Interrupt mask Bit Values: 0x0 (MASK): Mask OUT Endpoint 0 Interrupt 0x1 (NOMASK): No Interrupt mask	0x0
15:7	R	RESERVED	RESERVED	0x0
6	R/W	InEpMsk6	IN Endpoint 6 Interrupt mask Bit Values: 0x0 (MASK): Mask IN Endpoint Interrupt 0x1 (NOMASK): No Interrupt mask	0x0
5	R/W	InEpMsk5	IN Endpoint 5 Interrupt mask Bit Values: 0x0 (MASK): Mask IN Endpoint Interrupt 0x1 (NOMASK): No Interrupt mask	0x0
4	R/W	InEpMsk4	IN Endpoint 4 Interrupt mask Bit Values: 0x0 (MASK): Mask IN Endpoint Interrupt 0x1 (NOMASK): No Interrupt mask	0x0

Bits	Access	Name	Description	Reset
3	R/W	InEpMsk3	IN Endpoint 3 Interrupt mask Bit Values: 0x0 (MASK): Mask IN Endpoint Interrupt 0x1 (NOMASK): No Interrupt mask	0x0
2	R/W	InEpMsk2	IN Endpoint 2 Interrupt mask Bit Values: 0x0 (MASK): Mask IN Endpoint Interrupt 0x1 (NOMASK): No Interrupt mask	0x0
1	R/W	InEpMsk1	IN Endpoint 1 Interrupt mask Bit Values: 0x0 (MASK): Mask IN Endpoint Interrupt 0x1 (NOMASK): No Interrupt mask	0x0
0	R/W	InEpMsk0	IN Endpoint 0 Interrupt mask Bit Values: 0x0 (MASK): Mask IN Endpoint 0 Interrupt 0x1 (NOMASK): No Interrupt mask	0x0

DTKNQR1

This register is valid only in non-periodic Shared FIFO operation (OTG_EN_DED_TX_FIFO = 0). The depth of the IN Token Sequence Learning Queue is specified for Device Mode IN Token Sequence Learning Queue Depth during coreConsultant configuration (parameter OTG_TOKEN_QUEUE_DEPTH). The queue is 4 bits wide to store the endpoint number. A read from this register returns the first 5 endpoint entries of the IN Token Sequence Learning Queue. When the queue is full, the new token is pushed into the queue and oldest token is discarded.

Offset:0x820 Total Reset Value:0x0000_0000 RESERVED

DTKNQR2

This register is valid only in shared non-periodic Shared FIFO operation (OTG_EN_DED_TX_FIFO = 0). A read from this register returns the next 8 endpoint entries of the learning queue.

Offset:0x824 Total Reset Value:0x0000_0000 RESERVED

DVBUSDIS

This register specifies the VBUS discharge time after VBUS pulsing during SRP.

Offset:0x828 Total Reset Value:0x0000_17d7

Bits	Access	Name	Description	Reset
31:16	R	RESERVED	RESERVED	0x0
15:0	R/W	DVBUSDis	Device VBUS Discharge Time (DVBUSDis) Specifies the VBUS discharge time after VBUS pulsing during SRP. This value equals (VBUS discharge time in PHY clocks) / 1, 024. The value you use depends whether the PHY is operating at 30MHz (16-bit data width) or 60 MHz (8-bit data width). Depending on your VBUS load, this value can need adjustment.	0x17d7

DVBUSPULSE

This register contains the VBUS pulsing time during SRP.

Offset:0x82c Total Reset Value:0x0000_05b8

Bits	Access	Name	Description	Reset
31:12	R	RESERVED	RESERVED	0x0
11:0	R/W	DVBUSPulse	Device VBUS Pulsing Time (DVBUSPulse) Specifies the VBUS pulsing time during SRP. This value equals (VBUS pulsing time in PHY clocks) / 1, 024 The value you use depends whether the PHY is operating at 30MHz (16-bit data width) or 60 MHz (8-bit data width).	0x5b8

DTKNQR3

This register is valid only in shared non-periodic Shared FIFO operation (OTG_EN_DED_TX_FIFO = 0). A read from this register returns the next 8 endpoint entries of the learning queue.

Offset:0x830 Total Reset Value:0x000b_0000 RESERVED

DTHRCTL

This register contains the Receive and Transmit Threshold characteristics of the Device controller.

Offset:0x830 Total Reset Value:0x0000_0002

Bits	Access	Name	Description	Reset
31:28	R	RESERVED	RESERVED	0x0
27	R/W	ArbPrkEn	Arbiter Parking Enable (ArbPrkEn) Values: 0x0 (DISABLED): Disable DMA arbiter parking 0x1 (ENABLED): Enable DMA arbiter parking for IN endpoints	0x0
26	R	RESERVED1	RESERVED	0x0
25:17	R/W	RxThrLen	Receive Threshold Length (RxThrLen) This field specifies Receive thresholding size in DWORDS. This field also specifies the amount of data received on the USB before the core can start transmitting on the AHB. The threshold length has to be at least eight DWORDS. The recommended value for ThrLen is to be the same as the programmed AHB Burst Length (GAHB_CFG.HBstLen).	0x0
16	R/W	RxThrEn	Receive Threshold Enable (RxThrEn) Values: 0x0 (DISABLED): Disable thresholding 0x1 (ENABLED): Enable thresholding in the receive direction	0x0
15:13	R	RESERVED2	RESERVED	0x0
12:11	R/W	AHBThrRatio	AHB Threshold Ratio (AHBThrRatio) Values: 0x0 (THRESZERO): AHB threshold = MAC threshold 0x1 (THRESONE): AHB threshold = MAC threshold /2 0x2 (THRESTWO): AHB threshold = MAC threshold /4 0x3 (THRETHREE): AHB threshold = MAC threshold /8	0x0
10:2	R/W	TxThrLen	Transmit Threshold Length (TxThrLen)	0x8

Bits	Access	Name	Description	Reset
1	R/W	ISOThrEn	ISO IN Endpoints Threshold Enable. (ISOThrEn) When this bit is Set, the core enables thresholding for isochronous IN endpoints. Values: 0x0 (DISABLED): No thresholding 0x1 (ENABLED): Enables thresholding for isochronous IN endpoints	0x0
0	R/W	NonISOThrEn	Non-ISO IN Endpoints Threshold Enable. (NonISOThrEn) When this bit is Set, the core enables thresholding for Non Isochronous IN endpoints. Values: 0x0 (DISABLED): No thresholding 0x1 (ENABLED): Enable thresholding for non-isochronous IN endpoints	0x0

DTKNQR4

This register is valid only in shared non-periodic Shared FIFO operation (OTG_EN_DED_TX_FIFO = 0). A read from this register returns the next 8 endpoint entries of the learning queue.

Offset:0x834 Total Reset Value:0x0000_0000 RESERVED

DIEPEMPMSK

This register is valid only in Dedicated FIFO operation (OTG_EN_DED_TX_FIFO = 1). This register is used to control the IN endpoint FIFO empty interrupt generation (DIEPINTn.TxfEmp).

Offset:0x834 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:16	R	RESERVED	RESERVED	0x0
15:0	R/W	InEpTxfEmpMsk	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 EP 0, bit 15 for IN EP 15 Values: 0x1 (EP0_MASK): Mask IN EP0 Tx FIFO Empty Interrupt 0x2 (EP1_MASK): Mask IN EP1 Tx FIFO Empty Interrupt 0x4 (EP2_MASK): Mask IN EP2 Tx FIFO Empty Interrupt 0x8 (EP3_MASK): Mask IN EP3 Tx FIFO Empty Interrupt 0x10 (EP4_MASK): Mask IN EP4 Tx FIFO Empty Interrupt 0x20 (EP5_MASK): Mask IN EP5 Tx FIFO Empty Interrupt 0x40 (EP6_MASK): Mask IN EP6 Tx FIFO Empty Interrupt 0x80 (EP7_MASK): Mask IN EP7 Tx FIFO Empty Interrupt 0x100 (EP8_MASK): Mask IN EP8 Tx FIFO Empty Interrupt 0x200 (EP9_MASK): Mask IN EP9 Tx FIFO Empty Interrupt 0x400 (EP10_MASK): Mask IN EP10 Tx FIFO Empty Interrupt 0x800 (EP11_MASK): Mask IN EP11 Tx FIFO Empty Interrupt 0x1000 (EP12_MASK): Mask IN EP12 Tx FIFO Empty Interrupt 0x2000 (EP13_MASK): Mask IN EP13 Tx FIFO Empty Interrupt 0x4000 (EP14_MASK): Mask IN EP14 Tx FIFO Empty Interrupt 0x8000 (EP15_MASK): Mask IN EP15 Tx FIFO Empty Interrupt	0x0

DEACHINT

This register is available in device mode and only when parameter "OTG_MULTI_PROC_INTRPT" on page 121=1. There is one interrupt bit per endpoint, up to a maximum of 16 bits for OUT endpoints and 16 bits for IN endpoints. For a bidirectional endpoint, the corresponding IN and OUT interrupt bits are used. Bits in this register are set and cleared when the application sets and clears bits in the corresponding Device Endpoint-n Interrupt register (DIEPINTn/DOEPINTn). The interrupt is automatically cleared once the DOEPINTn/DIEPINTn interrupt is cleared by the application.

Offset:0x838 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:16	R	EchOutEPInt	Each OUT Endpoint Interrupt Bits (EchOutEPInt) One bit per OUT endpoint: Bit 16 for OUT endpoint 0, bit 31 for OUT endpoint 15 Values: 0x1 (OEP0_INT): OUT EP0 Interrupt 0x2 (OEP1_INT): OUT EP1 Interrupt 0x4 (OEP2_INT): OUT EP2 Interrupt 0x8 (OEP3_INT): OUT EP3 Interrupt 0x10 (OEP4_INT): OUT EP4 Interrupt 0x20 (OEP5_INT): OUT EP5 Interrupt 0x40 (OEP6_INT): OUT EP6 Interrupt 0x80 (OEP7_INT): OUT EP7 Interrupt 0x100 (OEP8_INT): OUT EP8 Interrupt 0x200 (OEP9_INT): OUT EP9 Interrupt 0x400 (OEP10_INT): OUT EP10 Interrupt 0x800 (OEP11_INT): OUT EP11 Interrupt 0x1000 (OEP12_INT): OUT EP12 Interrupt 0x2000 (OEP13_INT): OUT EP13 Interrupt 0x4000 (OEP14_INT): OUT EP14 Interrupt 0x8000 (OEP15_INT): OUT EP15 Interrupt	0x0
15:0	R	EchInEPInt	Each IN Endpoint Interrupt Bits (EchInEPInt) One bit per IN Endpoint: Bit 0 for IN endpoint 0, bit 15 for endpoint 15 Values: 0x1 (IEP0_INT): IN EP0 Interrupt 0x2 (IEP1_INT): IN EP1 Interrupt 0x4 (IEP2_INT): IN EP2 Interrupt 0x8 (IEP3_INT): IN EP3 Interrupt 0x10 (IEP4_INT): IN EP4 Interrupt 0x20 (IEP5_INT): IN EP5 Interrupt 0x40 (IEP6_INT): IN EP6 Interrupt 0x80 (IEP7_INT): IN EP7 Interrupt 0x100 (IEP8_INT): IN EP8 Interrupt 0x200 (IEP9_INT): IN EP9 Interrupt 0x400 (IEP10_INT): IN EP10 Interrupt 0x800 (IEP11_INT): IN EP11 Interrupt 0x1000 (IEP12_INT): IN EP12 Interrupt 0x2000 (IEP13_INT): IN EP13 Interrupt 0x4000 (IEP14_INT): IN EP14 Interrupt 0x8000 (IEP15_INT): IN EP15 Interrupt	0x0

DEACHINTMSK

This register is available only when parameter OTG_MULTI_PROC_INTRPT=1. The Device Each Endpoint Interrupt Mask register works with the Device Each Endpoint Interrupt register to interrupt the application when an event occurs on a device endpoint. However, the Device Each Endpoints Interrupt (DEACHINT) register bit corresponding to that interrupt remains set.

Offset:0x83c Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:23	R	RESERVED	RESERVED	0x0
22	* Varies	EchOutEpMsk6	OUT EP6 Interrupt Mask Bit Values: 0x1 (OEP6_INT_MASK)	0x0
21	* Varies	EchOutEpMsk5	OUT EP5 Interrupt Mask Bit Values: 0x1 (OEP5_INT_MASK)	0x0
20	* Varies	EchOutEpMsk4	OUT EP4 Interrupt Mask Bit Values: 0x1 (OEP4_INT_MASK)	0x0
19	* Varies	EchOutEpMsk3	OUT EP3 Interrupt Mask Bit Values: 0x1 (OEP3_INT_MASK)	0x0
18	* Varies	EchOutEpMsk2	OUT EP2 Interrupt Mask Bit Values: 0x1 (OEP2_INT_MASK)	0x0
17	* Varies	EchOutEpMsk1	OUT EP1 Interrupt Mask Bit Values: 0x1 (OEP1_INT_MASK)	0x0
16	R/W	EchOutEpMsk0	OUT EP0 Interrupt Mask Bit Values: 0x1 (OEP0_INT_MASK)	0x0
15:7	R	RESERVED	RESERVED	0x0
6	* Varies	EchInEpMsk6	IN EP6 Interrupt Mask Bit Values: 0x1 (IEP6_INT_MASK)	0x0
5	* Varies	EchInEpMsk5	IN EP5 Interrupt Mask Bit Values: 0x1 (IEP5_INT_MASK)	0x0
4	* Varies	EchInEpMsk4	IN EP4 Interrupt Mask Bit Values: 0x1 (IEP4_INT_MASK)	0x0
3	* Varies	EchInEpMsk3	IN EP3 Interrupt Mask Bit Values: 0x1 (IEP3_INT_MASK)	0x0

Bits	Access	Name	Description	Reset
2	* Varies	EchInEpMsk2	IN EP2 Interrupt Mask Bit Values: 0x1 (IEP2_INT_MASK)	0x0
1	* Varies	EchInEpMsk1	IN EP1 Interrupt Mask Bit Values: 0x1 (IEP1_INT_MASK)	0x0
0	R/W	EchInEpMsk0	IN EP0 Interrupt Mask Bit Values: 0x1 (IEP0_INT_MASK)	0x0

DIEPEACHMSK0

This register is available in device mode and only when parameter "OTG_MULTI_PROC_INTRPT" on page 121=1. These registers are endpoint-specific mask registers for (DIEPINTn). The IN endpoint interrupt for a specific status in the DIEPINTn register can be masked by writing 0 to the corresponding bit in this register. Status bits are masked by default.

Offset:0x840 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:14			Reserved Field: Yes	
13	R/W	NAKMsk	NAK interrupt Mask (NAKMsk) Values: 0x0 (MASK): Mask NAK interrupt 0x1 (NOMASK): No NAK interrupt Mask	0x0
12:10			Reserved Field: Yes	
9	R/W	BNAIntrMsk	BNA interrupt Mask (BNAIntrMsk) Values: 0x0 (MASK): Mask BNA interrupt 0x1 (NOMASK): No BNA interrupt Mask	0x0
8	R/W	TxfifoUndrnMsk	Fifo Underrun Mask (TxfifoUndrnMsk) Values: 0x0 (MASK): Mask Fifo Underrun 0x1 (NOMASK): No Fifo Underrun Mask	0x0
7	R	RESERVED	RESERVED	0x0
6	R/W	INEPNakEffMsk	IN Endpoint NAK Effective Mask (INEPNakEffMsk) Values: 0x0 (MASK): Mask IN Endpoint NAK Effective 0x1 (NOMASK): No IN Endpoint NAK Effective Mask	0x0
5	R/W	INTknEPMisMsk	IN Token received with EP Mismatch Mask (INTknEPMisMsk) Values: 0x0 (MASK): Mask IN Token received with EP Mismatch 0x1 (NOMASK): No IN Token received with EP Mismatch Mask	0x0

Bits	Access	Name	Description	Reset
4	R/W	INTknTXFEmpMsk	IN Token Received When TxFIFO Empty Mask (INTknTXFEmpMsk) Values: 0x0 (MASK): Mask IN Token received When TxFIFO Empty 0x1 (NOMASK): No IN Token received When TxFIFO Empty Mask	0x0
3	R/W	TimeOUTMsk	Timeout Condition Mask (TimeOUTMsk) (Non-isochronous endpoints) Values: 0x0 (MASK): Mask Timeout Condition 0x1 (NOMASK): No Timeout Condition Mask	0x0
2	R/W	AHBErrMsk	AHB Error Mask (AHBErrMsk) Values: 0x0 (MASK): Mask AHB Error 0x1 (NOMASK): No AHB Error Mask	0x0
1	R/W	EPDisbldMsk	Endpoint Disabled Interrupt Mask (EPDisbldMsk) Values: 0x0 (MASK): Mask Endpoint Disabled Interrupt 0x1 (NOMASK): No Endpoint Disabled Interrupt Mask	0x0
0	R/W	XferComplMsk	Transfer Completed Interrupt Mask (XferComplMsk) Values: 0x0 (MASK): Mask Transfer Completed Interrupt 0x1 (NOMASK): No Transfer Completed Interrupt Mask	0x0

DIEPEACHMSKI

This register contains the interrupts for the IN Endpoints of the Device controller. Offset: 0x840 + i*004 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:14			Reserved Field: Yes	
13	R/W	NAKMsk	NAK interrupt Mask (NAKMsk) Values: 0x0 (MASK): Mask NAK Interrupt 0x1 (NOMASK): No NAK Interrupt Mask	0x0
12:10			Reserved Field: Yes	
9	R/W	BNAIntrMsk	BNA interrupt Mask (BNAIntrMsk) Values: 0x0 (MASK): Mask BNA Interrupt 0x1 (NOMASK): No BNA Interrupt Mask	0x0
8	R/W	TxfifoUndrnMsk	Fifo Underrun Mask (TxfifoUndrnMsk) Values: 0x0 (MASK): Mask Fifo Underrun Interrupt 0x1 (NOMASK): No Fifo Underrun Interrupt Mask	0x0
7	R	RESERVED	RESERVED	0x0

Bits	Access	Name	Description	Reset
6	R/W	INEPNakEffMsk	IN Endpoint NAK Effective Mask (INEPNakEffMsk) Values: 0x0 (MASK): Mask IN Endpoint NAK Effective 0x1 (NOMASK): No IN Endpoint NAK Effective Mask	0x0
5	R/W	INTknEPMisMsk	IN Token received with EP Mismatch Mask (INTknEPMisMsk) Values: 0x0 (MASK): Mask IN Token received with EP Mismatch Interrupt 0x1 (NOMASK): No Mask IN Token received with EP Mismatch Interrupt	0x0
4	R/W	INTknTXFEmpMsk	IN Token Received When TxFIFO Empty Mask(INTknTXFEmpMsk) Values: 0x0 (MASK): Mask IN Token Received When TxFIFO Empty Interrupt 0x1 (NOMASK): No IN Token Received When TxFIFO Empty Interrupt	0x0
3	R/W	TimeOUTMsk	Timeout Condition Mask (TimeOUTMsk) (Non-isochronous endpoints) Values: 0x0 (MASK): Mask Timeout Condition Interrupt 0x1 (NOMASK): No Timeout Condition Interrupt Mask	0x0
2	R/W	AHBErrMsk	AHB Error Mask (AHBErrMsk) Values: 0x0 (MASK): Mask AHB Error 0x1 (NOMASK): No AHB Error Mask	0x0
1	R/W	EPDisbldMsk	Endpoint Disabled Interrupt Mask (EPDisbldMsk) Values: 0x0 (MASK): Mask Endpoint Disabled Interrupt 0x1 (NOMASK): No Endpoint Disabled Interrupt Mask	0x0
0	R/W	XferComplMsk	Transfer Completed Interrupt Mask (XferComplMsk) Values: 0x0 (MASK): Mask Transfer Completed Interrupt 0x1 (NOMASK): No Transfer Completed Interrupt Mask	0x0

DOEPEACHMSK0

This register is available in device mode and only when parameter OTG_MULTI_PROC_INTRPT=1. These registers are endpoint specific mask registers for (DOEPINTn). The OUT endpoint interrupt for a specific status in the DOEPINTn register can be masked by writing 0 to the corresponding bit in this register. Status bits are masked by default.

Offset:0x880 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:15			Reserved Field: Yes	

Bits	Access	Name	Description	Reset
14	R/W	NYETMsk	NYET interrupt Mask (NYETMsk) Values: 0x0 (MASK): Mask NYET Interrupt 0x1 (NOMASK): No NYET Interrupt Mask	0x0
13	R/W	NAKMsk	NAK interrupt Mask (NAKMsk) Values: 0x0 (MASK): Mask NAK Interrupt 0x1 (NOMASK): No NAK Interrupt Mask	0x0
12	R/W	BbleErrMsk	Babble Error interrupt Mask (BbleErrMsk) Values: 0x0 (MASK): Mask Babble Error Interrupt 0x1 (NOMASK): No Babble Error Interrupt Mask	0x0
11:10			Reserved Field: Yes	
9	R/W	BnaOutIntrMsk	BNA interrupt Mask (BnaOutIntrMsk) Values: 0x0 (MASK): Mask BNA Interrupt 0x1 (NOMASK): No BNA Interrupt Mask	0x0
8	R/W	OutPktErrMsk	OUT Packet Error Mask (OutPktErrMsk) Values: 0x0 (MASK): Mask OUT Packet Error 0x1 (NOMASK): No OUT Packet Error Mask	0x0
7	R	RESERVED	RESERVED	
6	R/W	Back2BackSETup	Back-to-Back SETUP Packets Received Mask (Back2BackSETup) Applies to control OUT endpoints only. Values: 0x0 (MASK): Mask Back-to-Back SETUP Packets Received 0x1 (NOMASK): No Back-to-Back SETUP Packets Received Mask	0x0
5	R/W	StsPhsRcvdMsk	Status Phase Received Mask (StsPhsRcvdMsk) Applies to control OUT endpoints only. Values: 0x0 (MASK): Mask Status Phase Received 0x1 (NOMASK): No Status Phase Received Mask	0x0
4	R/W	OUTTknEPdisMsk	OUT Token Received when Endpoint Disabled Mask (OUTTknEPdisMsk) Applies to control OUT endpoints only. Values: 0x0 (MASK): Mask OUT Token Received when Endpoint Disabled 0x1 (NOMASK): No OUT Token Received when Endpoint Disabled Mask	0x0

Bits	Access	Name	Description	Reset
3	R/W	SetUPMsk	SETUP Phase Done Mask (SetUPMsk) Applies to control endpoints only. Values: 0x0 (MASK): Mask SETUP Phase Done 0x1 (NOMASK): No SETUP Phase Done Mask	0x0
2	R/W	AHBErrMsk	AHB Error (AHBErrMsk) Values: 0x0 (MASK): Mask AHB Error 0x1 (NOMASK): No AHB Error Mask	0x0
1	R/W	EPDisbldMsk	Endpoint Disabled Interrupt Mask (EPDisbldMsk) Values: 0x0 (MASK): Mask Endpoint Disabled Interrupt 0x1 (NOMASK): No Endpoint Disabled Interrupt Mask	0x0
0	R/W	XferComplMsk	Transfer Completed Interrupt Mask (XferComplMsk) Values: 0x0 (MASK): Mask Transfer Completed Interrupt 0x1 (NOMASK): No Transfer Completed Interrupt Mask	0x0

DOEPEACHMSKi

This register contains the interrupts for the OUT Endpoints of the Device controller.

Offset:0x880 + i*004 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:15			Reserved Field: Yes	
14	R/W	NYETMsk	NYET interrupt Mask (NYETMsk) Values: 0x0 (MASK): Mask NYET Interrupt 0x1 (NOMASK): No NYET Interrupt Mask	0x0
13	R/W	NAKMsk	NAK interrupt Mask (NAKMsk) Values: 0x0 (MASK): Mask NAK Response Received Interrupt 0x1 (NOMASK): No NAK Response Received Interrupt Mask	0x0
12	R/W	BbleErrMsk	Babble Error interrupt Mask (BbleErrMsk) Values: 0x0 (MASK): Mask Babble Error Interrupt 0x1 (NOMASK): No Babble Error Interrupt Mask	0x0
11:10			Reserved Field: Yes	
9	R/W	BnaOutIntrMsk	BNA interrupt Mask (BnaOutIntrMsk) Values: 0x0 (MASK): Mask BNA Interrupt 0x1 (NOMASK): No BNA Interrupt Mask	0x0

Bits	Access	Name	Description	Reset
8	R/W	OutPktErrMsk	OUT Packet Error Mask (OutPktErrMsk) Values: 0x0 (MASK): Mask OUT Packet Error Interrupt 0x1 (NOMASK): No OUT Packet Error Interrupt Mask	0x0
7	R	RESERVED	RESERVED	0x0
6	R/W	Back2BackSETup	Back-to-Back SETUP Packets Received Mask (Back2BackSETup) Applies to control OUT endpoints only. Values: 0x0 (MASK): Mask Back-to-Back SETUP Packets Received 0x1 (NOMASK): No Back-to-Back SETUP Packets Received Mask	0x0
5	R/W	StsPhsRcvdMsk	Status Phase Received Mask (StsPhsRcvdMsk) Applies to control OUT endpoints only. Values: 0x0 (MASK): Mask Status Phase Received 0x1 (NOMASK): No Status Phase Received Mask	0x0
4	R/W	OUTTknEPdisMsk	OUT Token Received when Endpoint Disabled Mask (OUTTknEPdisMsk) Applies to control OUT endpoints only. Values: 0x0 (MASK): Mask OUT Token Received when Endpoint Disabled Interrupt 0x1 (NOMASK): No OUT Token Received when Endpoint Disabled Interrupt Mask	0x0
3	R/W	SetUPMsk	SETUP Phase Done Mask (SetUPMsk) Applies to control endpoints only. Values: 0x0 (MASK): Mask SETUP Phase Done Interrupt 0x1 (NOMASK): No SETUP Phase Done Interrupt Mask	0x0
2	R/W	AHBErrMsk	AHB Error (AHBErrMsk) Values: 0x0 (MASK): AHB Error Mask 0x1 (NOMASK): No AHB Error Mask	0x0
1	R/W	EPDisbldMsk	Endpoint Disabled Interrupt Mask (EPDisbldMsk) Values: 0x0 (MASK): Mask Endpoint Disabled Interrupt 0x1 (NOMASK): No Endpoint Disabled Interrupt Mask	0x0
0	R/W	XferComplMsk	Transfer Completed Interrupt Mask (XferComplMsk) Values: 0x0 (MASK): Transfer Completed Mask 0x1 (NOMASK): No Transfer Completed Mask	0x0

DIEPCTL0

This register is used to control the characteristics of the IN Endpoint 0 of the Device controller.

Offset:0x900 Total Reset Value:0x0000_0400

Bits	Access	Name	Description	Reset
31	R/W1S	EPEna	Endpoint Enable (EPEna) Values: 0x0 (INACTIVE): No action 0x1 (ACTIVE): Enable Endpoint	0x0
30	R/W1S	EPDis	Endpoint Disable (EPDis) Values: 0x0 (INACTIVE): No action 0x1 (ACTIVE): Disabled Endpoint	0x0
29:28	R	RESERVED	RESERVED	0x0
27	W	SNAK	Values: 0x0 (NOSET): No action 0x1 (SET): Set NAK	0x0
26	W	CNAK	Clear NAK (CNAK) A write to this bit clears the NAK bit for the endpoint. Values: 0x0 (NOCLEAR): No action 0x1 (CLEAR): Clear NAK	0x0
25:22	RW	TxFNum	TxFIFO Number (TxFNum) Values: 0x0 (TXFIFO0): Tx FIFO 0 0x1 (TXFIFO1): Tx FIFO 1 0x2 (TXFIFO2): Tx FIFO 2 0x3 (TXFIFO3): Tx FIFO 3 0x4 (TXFIFO4): Tx FIFO 4 0x5 (TXFIFO5): Tx FIFO 5 0x6 (TXFIFO6): Tx FIFO 6 0x7 (TXFIFO7): Tx FIFO 7 0x8 (TXFIFO8): Tx FIFO 8 0x9 (TXFIFO9): Tx FIFO 9 0xa (TXFIFO10): Tx FIFO 10 0xb (TXFIFO11): Tx FIFO 11 0xc (TXFIFO12): Tx FIFO 12 0xd (TXFIFO13): Tx FIFO 13 0xe (TXFIFO14): Tx FIFO 14 0xf (TXFIFO15): Tx FIFO 15	0x0
21	R/W1S	Stall	STALL Handshake (Stall) Values: 0x0 (INACTIVE): No Stall 0x1 (ACTIVE): Stall Handshake	0x0
20	R	RESERVED1	RESERVED	0x0
19:18	R	EPTYPE	Endpoint Type (EPTYPE) Hardcoded to 00 for control. Values: 0x0 (ACTIVE): Endpoint Control 0	0x0

Bits	Access	Name	Description	Reset
17	R	NAKSts	NAK Status (NAKSts) Indicates the following: Values: 0x0 (INACTIVE): The core is transmitting non-NAK handshakes based on the FIFO status 0x1 (ACTIVE): The core is transmitting NAK handshakes on this endpoint	0x0
16	R	RESERVED2	RESERVED	0x0
15	R	USBActEP	USB Active Endpoint (USBActEP) Values: 0x1 (ACTIVE0): Control endpoint is always active	0x1
14:2	R	RESERVED3	RESERVED	0x0
1:0	R/W	MPS	Maximum Packet Size (MPS) Applies to IN and OUT endpoints. Values: 0x0 (BYTES64): 64 bytes 0x1 (BYTES32): 32 bytes 0x2 (BYTES16): 16 bytes 0x3 (BYTES8): 8 bytes	0x0

DIEPINT0

This register indicates the status of an endpoint with respect to USB- and AHB-related events. It is shown in the "Interrupt Hierarchy" figure in the databook. The application must read this register when the OUT Endpoints Interrupt bit or IN Endpoints Interrupt bit of the Core Interrupt register (GINTSTS.OEPInt or GINTSTS.IEPInt, respectively) is set. Before the application can read this register, it must first read the Device All Endpoints Interrupt (DAINT) register to get the exact endpoint number for the Device Endpoint-n Interrupt register. The application must clear the appropriate bit in this register to clear the corresponding bits in the DAINT and GINTSTS registers

Offset:0x908 Total Reset Value:0x0000_0080

Bits	Access	Name	Description	Reset
31:15	R	RESERVED	RESERVED	0x0
14	R/W1C	NYETIntrpt	NYET Interrupt (NYETIntrpt) The core generates this interrupt when a NYET response is transmitted for a non isochronous OUT endpoint. Values: 0x0 (INACTIVE): No interrupt 0x1 (ACTIVE): NYET Interrupt	0x0
13	R/W1C	NAKIntrpt	NAK Interrupt (NAKInterrupt) The core generates this interrupt when a NAK is transmitted or received by the device. In case of isochronous IN endpoints the interrupt gets generated when a zero length packet is transmitted due to un-availability of data in the TXFifo. Values: 0x0 (INACTIVE): No interrupt 0x1 (ACTIVE): NAK Interrupt	0x0

Bits	Access	Name	Description	Reset
12	R/W1C	BbleErr	NAK Interrupt (BbleErr) The core generates this interrupt when babble is received for the endpoint. Values: 0x0 (INACTIVE): No interrupt 0x1 (ACTIVE): BbleErr interrupt	0x0
11	R/W1C	PktDrpSts	Packet Drop Status (PktDrpSts) This bit indicates to the application that an ISOC OUT packet has been dropped. This bit does not have an associated mask bit and does not generate an interrupt. Dependency: This bit is valid in non Scatter/Gather DMA mode when periodic transfer interrupt feature is selected. Values: 0x0 (INACTIVE): No interrupt 0x1 (ACTIVE): Packet Drop Status	0x0
10			Reserved Field: Yes	
9	R/W1C	BNAIntr	BNA (Buffer Not Available) Interrupt (BNAIntr) This bit is valid only when Scatter/Gather DMA mode is enabled. The core generates this interrupt when the descriptor accessed is not ready for the Core to process, such as Host busy or DMA done. Values: 0x0 (INACTIVE): No BNA interrupt 0x1 (ACTIVE): BNA interrupt	0x0
8	R/W1C	TxfifoUndrn	Fifo Underrun (TxfifoUndrn) Applies to IN endpoints only. The core generates this interrupt when it detects a transmit FIFO underrun condition in threshold mode for this endpoint. Values: 0x0 (INACTIVE): No Fifo Underrun interrupt 0x1 (ACTIVE): Fifo Underrun interrupt	0x0
7	R	TxFEmp	Transmit FIFO Empty (TxFEmp) This bit is valid only for IN Endpoints This interrupt is asserted when the TxFIFO for this endpoint is either half or completely empty. The half or completely empty status is determined by the TxFIFO Empty Level bit in the Core AHB Configuration register (GAHBCFG.NPTxFEmpLvl). Values: 0x0 (INACTIVE): No Transmit FIFO Empty interrupt 0x1 (ACTIVE): Transmit FIFO Empty interrupt	0x1

Bits	Access	Name	Description	Reset
6	R/W1C	INEPNakEff	IN Endpoint NAK Effective (INEPNakEff) Applies to periodic IN endpoints only. This bit can be cleared when the application clears the IN endpoint NAK by writing to DIEPCTLn.CNAK. This interrupt indicates that the core has sampled the NAK bit Set (either by the application or by the core). The interrupt indicates that the IN endpoint NAK bit Set by the application has taken effect in the core. This interrupt does not guarantee that a NAK handshake is sent on the USB. A STALL bit takes priority over a NAK bit. Values: 0x0 (INACTIVE): No IN Endpoint NAK Effective interrupt 0x1 (ACTIVE): IN Endpoint NAK Effective interrupt	0x0
5	R/W1C	INTknEPMis	IN Token Received with EP Mismatch (INTknEPMis) Applies to non-periodic IN endpoints only. Indicates that the data in the top of the non-periodic TxFIFO belongs to an endpoint other than the one for which the IN token was received. This interrupt is asserted on the endpoint for which the IN token was received. Values: 0x0 (INACTIVE): No IN Token Received with EP Mismatch interrupt 0x1 (ACTIVE): IN Token Received with EP Mismatch interrupt	0x0
4	R/W1C	INTknTXFEmp	IN Token Received When TxFIFO is Empty (INTknTXFEmp) Applies to non-periodic IN endpoints only. Indicates that an IN token was received when the associated TxFIFO (periodic/non-periodic) was empty. This interrupt is asserted on the endpoint for which the IN token was received. Values: 0x0 (INACTIVE): No IN Token Received when TxFIFO Empty interrupt 0x1 (ACTIVE): IN Token Received when TxFIFO Empty Interrupt	0x0
3	R/W1C	TimeOUT	Timeout Condition (TimeOUT) Values: 0x0 (INACTIVE): No Timeout interrupt 0x1 (ACTIVE): Timeout interrupt	0x0
2	R/W1C	AHBErr	AHB Error (AHBErr) Applies to IN and OUT endpoints. This is generated only in Internal DMA mode when there is an AHB error during an AHB read/write. The application can read the corresponding endpoint DMA address register to get the error address. For details, see "AHB Error Handling" in the Programming Guide. Values: 0x0 (INACTIVE): No AHB Error Interrupt 0x1 (ACTIVE): AHB Error interrupt	0x0

Bits	Access	Name	Description	Reset
1	R/W1C	EPDisbld	Endpoint Disabled Interrupt (EPDisbld) Applies to IN and OUT endpoints. This bit indicates that the endpoint is disabled per the application's request. Values: 0x0 (INACTIVE): No Endpoint Disabled Interrupt 0x1 (ACTIVE): Endpoint Disabled Interrupt	0x0
0	R/W1C	XferCompl	Transfer Completed Interrupt (XferCompl) Applies to IN and OUT endpoints. Values: 0x0 (INACTIVE): No Transfer Complete Interrupt 0x1 (ACTIVE): Transfer Completed Interrupt	0x0

DIEPTSIZE0

The application must modify this register before enabling endpoint 0. Once endpoint 0 is enabled using Endpoint Enable bit of the Device Control Endpoint 0 Control registers (DIEPCTL0.EPEna/DOEPCTL0.EPEna), the core modifies this register. The application can only read this register once the core has cleared the Endpoint Enable bit. Nonzero endpoints use the registers for endpoints 115. When Scatter/Gather DMA mode is enabled, this register must not be programmed by the application. If the application reads this register when Scatter/Gather DMA mode is enabled, the core returns all zeros.

Offset:0x910 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:21	R	RESERVED	RESERVED	0x0
20:19	R/W	PktCnt	Packet Count (PktCnt) Indicates the total number of USB packets that constitute the Transfer Size amount of data for endpoint 0. IN Endpoints : This field is decremented every time a packet (maximum size or short packet) is read from the TxFIFO. OUT Endpoints: This field is decremented every time a packet (maximum size or short packet) is written to the RxFIFO.	0x0
18:7	R	RESERVED1	RESERVED	0x0
6:0	R/W	XferSize	Transfer Size (XferSize) This field contains the transfer size in bytes for the current endpoint. The transfer size (XferSize) = Sum of buffer sizes across all descriptors in the list for the endpoint. In Buffer DMA, the core only interrupts the application after it has exhausted the transfer size amount of data. The transfer size can be set to the maximum packet size of the endpoint, to be interrupted at the end of each packet. IN Endpoints: The core decrements this field every time a packet from the external memory is written to the TxFIFO. OUT Endpoints: The core decrements this field every time a packet is read from the RxFIFO and written to the external memory.	0x0

DIEPDMA0

This register contains the DMA Address for the IN Endpoint 0 of the Device controller.

Offset:0x914 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:0	R/W	DMAAddr	DMAAddr This field holds the start address of the external memory for storing or fetching endpoint data. Note: For control endpoints, this field stores control OUT data packets as well as SETUP transaction data packets. When more than three SETUP packets are received back-to-back, the SETUP data packet in the memory is overwritten. This register is incremented on every AHB transaction. The application can give only a DWORD-aligned address. When Scatter/Gather DMA mode is not enabled, the application programs the start address value in this field. When Scatter/Gather DMA mode is enabled, this field indicates the base pointer for the descriptor list.	0/X depending upon the SPRAM model

DTXFSTS0

This register contains information about the IN Endpoint Transmit FIFO of the Device controller.

Offset:0x918 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:16	RESERVED	R	RESERVED	0x0
15:0	INEPTxFSpAvail	R	IN Endpoint TxFIFO Space Avail (INEPTxFSpAvail) Indicates the amount of free space available in the Endpoint TxFIFO. Values are in terms of 32-bit words. 16'h0: Endpoint TxFIFO is full 16'h1: 1 word available 16'h2: 2 words available 16'hn: n words available (where 0 n 32,768) 16'h8000: 32,768 words available Others: Reserved	0x0

DIEPDMAB0

This register contains the DMA Buffer Address for the IN Endpoint 0 of the Device controller.

Offset:0x91c Total Reset Value:0x000b_0000

Bits	Access	Name	Description	Reset
31:0	DMABufferAddr	R	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.	DMA registers are implemented in SPRAM. The reset values of these registers in simulation can be 0/X depending upon the SPRAM model.

DIEPCTLi

This register is used to control the characteristics of Endpoint i. Offset: 0x900 + i*20 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset

Bits	Access	Name	Description	Reset
31	R/W1S	EPEna	Endpoint Enable (EPEna) Applies to IN and OUT endpoints. Values: 0x0 (INACTIVE): No Action 0x1 (ACTIVE): Enable Endpoint	0x0
30	R/W1S	EPDis	Endpoint Disable (EPDis) Applies to IN and OUT endpoints. Values: 0x0 (INACTIVE): No Action 0x1 (ACTIVE): Disable Endpoint	0x0
29	W	SetD1PID	Set DATA1 PID (SetD1PID) Values: 0x0 (DISABLED): Disables Set DATA1 PID or Do not force Odd (micro)Frame 0x1 (ENABLED): Set Endpoint Data PID to DATA1 or Sets EO_FrNum field to Odd (micro)Frame	0x0
28	W	SetD0PID	Set DATA0 PID (SetD0PID) Values: 0x0 (DISABLED): Disables Set DATA0 PID or Do not force Even (micro)Frame 0x1 (ENABLED): Set Endpoint Data PID to DATA0 or Sets EO_FrNum field to Even (micro)Frame	0x0
27	W	SNAK	Set NAK (SNAK) A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also Set this bit for an endpoint after a SETUP packet is received on that endpoint. Values: 0x0 (INACTIVE): No Set NAK 0x1 (ACTIVE): Set NAK Value After Reset: 0x0 Exists: Always	0x0
26	W	CNAK	Clear NAK (CNAK) A write to this bit clears the NAK bit for the endpoint. Values: 0x0 (INACTIVE): No Clear NAK 0x1 (ACTIVE): Clear NAK	0x0

Bits	Access	Name	Description	Reset
25:22	R/W	TxFNum	TxFIFO Number (TxFNum) Values: 0x0 (TXFIFO0): Tx FIFO 0 0x1 (TXFIFO1): Tx FIFO 1 0x2 (TXFIFO2): Tx FIFO 2 0x3 (TXFIFO3): Tx FIFO 3 0x4 (TXFIFO4): Tx FIFO 4 0x5 (TXFIFO5): Tx FIFO 5 0x6 (TXFIFO6): Tx FIFO 6 0x7 (TXFIFO7): Tx FIFO 7 0x8 (TXFIFO8): Tx FIFO 8 0x9 (TXFIFO9): Tx FIFO 9 0xa (TXFIFO10): Tx FIFO 10 0xb (TXFIFO11): Tx FIFO 11 0xc (TXFIFO12): Tx FIFO 12 0xd (TXFIFO13): Tx FIFO 13 0xe (TXFIFO14): Tx FIFO 14 0xf (TXFIFO15): Tx FIFO 15	0x0
21	R/W1S	Stall	STALL Handshake (Stall) Applies to non-control, non-isochronous IN and OUT endpoints only. Values: 0x0 (INACTIVE): STALL All non-active tokens 0x1 (ACTIVE): STALL All Active Tokens	0x0
20			Reserved Field: Yes	
19:18	R/W	EPTYPE	Endpoint Type (EPTYPE) This is the transfer type supported by this logical endpoint. 2'b00: Control 2'b01: Isochronous 2'b10: Bulk 2'b11: Interrupt Values: 0x0 (CONTROL): Control 0x1 (ISOCHRONOUS): Isochronous 0x2 (BULK): Bulk 0x3 (INTERRUPT): Interrupt	
17	R	NAKSts	NAK Status (NAKSts) Indicates the following: Values: 0x0 (NONNAK): The core is transmitting non-NAK handshakes based on the FIFO status 0x1 (NAK): The core is transmitting NAK handshakes on this endpoint	
16	R	DPID	Endpoint Data PID (DPID) Applies to interrupt/bulk IN and OUT endpoints only. Values: 0x0 (DATA0EVENFRM): DATA0 or Even Frame 0x1 (DATA1ODDFRM): DATA1 or Odd Frame	0x0

Bits	Access	Name	Description	Reset
15	R/W	USBActEP	USB Active Endpoint (USBActEP) Values: 0x0 (DISABLED): Not Active 0x1 (ENABLED): USB Active Endpoint	0x0
14:11	R/W	NextEp	Next Endpoint (NextEp) Applies to non-periodic IN endpoints only. Values: 0x0 (EP0): EP0 0x1 (EP1): EP1 0x2 (EP2): EP2 0x3 (EP3): EP3 0x4 (EP4): EP4 0x5 (EP5): EP5 0x6 (EP6): EP6 0x7 (EP7): EP7 0x8 (EP8): EP8 0x9 (EP9): EP9 0xa (EP10): EP10 0xb (EP11): EP11 0xc (EP12): EP12 0xd (EP13): EP13 0xe (EP14): EP14 0xf (EP15): EP15	0x0
10:0	R/W	MPS	Maximum Packet Size (MPS) The application must program this field with the maximum packet size for the current logical endpoint. This value is in bytes.	0x0

DIEPINTi

This register contains the interrupts for the IN Endpoint i of the Device controller.

Offset:0x824 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:15	R	RESERVED	RESERVED	0x0
14	R/W1C	NYETIntrpt	NYET Interrupt (NYETIntrpt) The core generates this interrupt when a NYET response is transmitted for a non isochronous OUT endpoint. Values: 0x0 (INACTIVE): No NYET interrupt 0x1 (ACTIVE): NYET Interrupt	0x0
13	R/W1C	NAKIntrpt	NAK Interrupt (NAKInterrupt) Values: 0x0 (INACTIVE): No NAK interrupt 0x1 (ACTIVE): NAK Interrupt	0x0

Bits	Access	Name	Description	Reset
12	R/W1C	BbleErr	NAK Interrupt (BbleErr) Values: 0x0 (INACTIVE): No interrupt 0x1 (ACTIVE): BbleErr interrupt	0x0
11	R/W1C	PktDrpSts	Packet Drop Status (PktDrpSts) Values: 0x0 (INACTIVE): No interrupt 0x1 (ACTIVE): Packet Drop Status interrupt	0x0
10			Reserved Field: Yes	
9	R/W1C	BNAIntr	BNA (Buffer Not Available) Interrupt (BNAIntr) Values: 0x0 (INACTIVE): No BNA interrupt 0x1 (ACTIVE): BNA interrupt	0x0
8	R/W1C	TxfifoUndrn	Fifo Underrun (TxfifoUndrn) Applies to IN endpoints Only Values: 0x0 (INACTIVE): No Tx FIFO Underrun interrupt 0x1 (ACTIVE): TxFIFO Underrun interrupt	0x0
7	R	TxFEmp	Transmit FIFO Empty (TxFEmp) This bit is valid only for IN endpoints Values: 0x0 (INACTIVE): No Transmit FIFO Empty interrupt 0x1 (ACTIVE): Transmit FIFO Empty interrupt	0x1
6	R/W1C	INEPNakEff	IN Endpoint NAK Effective (INEPNakEff) Applies to periodic IN endpoints only. Values: 0x0 (INACTIVE): No Endpoint NAK Effective interrupt 0x1 (ACTIVE): IN Endpoint NAK Effective interrupt	0x0
5	R/W1C	INTknEPMis	IN Token Received with EP Mismatch (INTknEPMis) Applies to non-periodic IN endpoints only. Values: 0x0 (INACTIVE): No IN Token Received with EP Mismatch interrupt 0x1 (ACTIVE): IN Token Received with EP Mismatch interrupt	0x0
4	R/W1C	INTknTXFEmp	IN Token Received When TxFIFO is Empty (INTknTXFEmp) Applies to non-periodic IN endpoints only. Values: 0x0 (INACTIVE): No IN Token Received interrupt 0x1 (ACTIVE): IN Token Received Interrupt	0x0
3	R/W1C	TimeOUT	Timeout Condition (TimeOUT) Values: 0x0 (INACTIVE): No Timeout interrupt 0x1 (ACTIVE): Timeout interrupt	0x0

Bits	Access	Name	Description	Reset
2	R/W1C	AHBErr	AHB Error (AHBErr) Applies to IN and OUT endpoints. Values: 0x0 (INACTIVE): No AHB Error Interrupt 0x1 (ACTIVE): AHB Error interrupt	0x0
1	R/W1C	EPDisbld	Endpoint Disabled Interrupt (EPDisbld) Applies to IN and OUT endpoints. Values: 0x0 (INACTIVE): No Endpoint Disabled Interrupt 0x1 (ACTIVE): Endpoint Disabled Interrupt	0x0
0	R/W1C	XferCompl	Transfer Completed Interrupt (XferCompl) Applies to IN and OUT endpoints. Values: 0x0 (INACTIVE): No Transfer Complete Interrupt 0x1 (ACTIVE): Transfer Complete Interrupt	0x0

DIEPTSIZEi

This register reflects the Transfer Size of the IN Endpoint i of the Device controller.

Offset:0x910 + i*20 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31	R	RESERVED	RESERVED	0x0
30:29	R/W	MC	MC Applies to IN endpoints only. Values: 0x1 (PACKETONE): 1 packet 0x2 (PACKETTWO): 2 packets 0x3 (PACKETTHREE): 3 packets	0x0
28:19	R/W	PktCnt	Packet Count (PktCnt) Indicates the total number of USB packets that constitute the Transfer Size amount of data for endpoint 0. This field is decremented every time a packet (maximum size or short packet) is read from the TxFIFO.	0x0
18:0	R/W	XferSize	Transfer Size (XferSize) Indicates the transfer size in bytes for the endpoint. The core interrupts the application only after it has exhausted the transfer size amount of data. The transfer size can be Set to the maximum packet size of the endpoint, to be interrupted at the end of each packet. The core decrements this field every time a packet from the external memory is written to the TxFIFO.	0x0

DIEPDMAi

This register contains the DMA Address for the IN Endpoint i of the Device controller.

Offset:0x914 + i*20 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset

Bits	Access	Name	Description	Reset
31:0	R/W	DMAAddr	"Holds the start address of the external memory for storing or fetching endpoint data.	
This register is incremented on every AHB transaction. The application can give only a DWORD-aligned address.				
When Scatter/Gather DMA mode is enabled, this field indicates the base pointer for the descriptor list."				
DMA registers are implemented in SPRAM. The reset values of these registers in simulation can be 0/X depending upon the SPRAM model.				

DTXFSTSi

This register reflects the status of the IN Endpoint Transmit FIFO Status Register i of the Device controller.

Offset:0x918 + i*20 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:16	R	RESERVED	RESERVED	0x0
15:0	R	INEPTxFSpcAvail	IN Endpoint Tx FIFO Space Avail (INEPTxFSpcAvail) Indicates the amount of free space available in the Endpoint Tx FIFO. Values are in terms of 32-bit words. 16'h0: Endpoint Tx FIFO is full 16'h1: 1 word available 16'h2: 2 words available 16'h_n: n words available (where 0 ≤ n ≤ 32,768) 16'h8000: 32,768 words available Others: Reserved	0x0

DIEPDMAB

This register contains the DMA Buffer Address of the IN Endpoint i of the Device controller.

Offset:0x91C + i*20 Total Reset Value:0x000b_0000

Bits	Access	Name	Description	Reset
31:0	R	DMABufferAddr	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.	DMA registers are implemented in SPRAM. The reset values of these registers in simulation can be 0/X depending upon the SPRAM model.

DOEPCTL0

This register is used to control the characteristics of the OUT Endpoint 0 of the Device controller.

Offset:0xb00 Total Reset Value:0x0000_0004

Bits	Access	Name	Description	Reset
31	R/W1S	EPEna	Endpoint Enable (EPEna) Values: 0x0 (INACTIVE): No action 0x1 (ACTIVE): Enable Endpoint	0x0
30	R	EPDis	Endpoint Disable (EPDis) The application cannot disable control OUT endpoint 0. Values: 0x0 (INACTIVE): No Endpoint disable	0x0
29:28	R	RESERVED	RESERVED	0x0
27	W	SNAK	Set NAK (SNAK) Values: 0x0 (NOSET): No action 0x1 (SET): Set NAK	0x0
26	W	CNAK	Clear NAK (CNAK) A write to this bit clears the NAK bit for the endpoint. Values: 0x0 (NOCLEAR): No action 0x1 (CLEAR): Clear NAK	0x0
25:22	R	RESERVED1	RESERVED	0x0
21	R/W1S	Stall	STALL Handshake (Stall) Values: 0x0 (INACTIVE): No Stall 0x1 (ACTIVE): Stall Handshake	0x0
20	R/W	SnP	RESERVED Values: 0x0 (RESERVED0): Reserved 0 0x1 (RESERVED1): Reserved 1	0x0
19:18	R	EPTYPE	Endpoint Type (EPTYPE) Hardcoded to 2'b00 for control. Values: 0x0 (ACTIVE): Endpoint Control 0	0x0
17	R	NAKSts	NAK Status (NAKSts) Indicates the following: Values: 0x0 (INACTIVE): The core is transmitting non-NAK handshakes based on the FIFO status 0x1 (ACTIVE): The core is transmitting NAK handshakes on this endpoint	0x0
16	R	RESERVED2	RESERVED	0x0
15	R	USBActEP	USB Active Endpoint (USBActEP) This bit is always set to 1, indicating that a control endpoint 0 is always active in all configurations and interfaces. Values: 0x1 (ACTIVE): USB Active Endpoint 0	0x1

Bits	Access	Name	Description	Reset
14:2	R	RESERVED3	RESERVED	0x0
1:0	R	MPS	Maximum Packet Size (MPS) Values: 0x0 (BYTE64): 64 bytes 0x1 (BYTE32): 32 bytes 0x2 (BYTE16): 16 bytes 0x3 (BYTE8): 8 bytes	0x0

DOEPINT0

This register is valid only in shared non-periodic Shared FIFO operation (OTG_EN_DED_TX_FIFO = 0). A read from this register returns the next 8 endpoint entries of the learning queue.

Offset:0xb08 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:16			Reserved Field: Yes	
15	R/W1C	StupPktRcvd	Setup Packet Received Values: 0x0 (NOT_RCVD): No Setup packet received 0x1 (RCVD): Setup packet received	0x0
14	R/W1C	NYETIntrpt	NYET Interrupt (NYETIntrpt) Values: 0x0 (INACTIVE): No NYET interrupt 0x1 (ACTIVE): NYET Interrupt	0x0
13	R/W1C	NAKIntrpt	NAK Interrupt (NAKInterrupt) Values: 0x0 (INACTIVE): No NAK interrupt 0x1 (ACTIVE): NAK Interrupt	0x0
12	R/W1C	BbleErr	NAK Interrupt (BbleErr) Values: 0x0 (INACTIVE): No BbleErr interrupt 0x1 (ACTIVE): BbleErr interrupt	0x0
11	R/W1C	PktDrpSts	Packet Drop Status (PktDrpSts) Values: 0x0 (INACTIVE): No interrupt 0x1 (ACTIVE): Packet Drop Status interrupt	0x0
10			Reserved Field: Yes	
9	R/W1C	BNAIntr	BNA (Buffer Not Available) Interrupt (BNAIntr) Values: 0x0 (INACTIVE): No BNA interrupt 0x1 (ACTIVE): BNA interrupt	0x0
8	R/W1C	OutPktErr	OUT Packet Error (OutPktErr) Applies to OUT endpoints Only Values: 0x0 (INACTIVE): No OUT Packet Error 0x1 (ACTIVE): OUT Packet Error	0x0

Bits	Access	Name	Description	Reset
7			Reserved Field: Yes	
6	R/W1C	Back2BackSETup	Back-to-Back SETUP Packets Received (Back2BackSETup) Applies to Control OUT endpoints only. Values: 0x0 (INACTIVE): No Back-to-Back SETUP Packets Received 0x1 (ACTIVE): Back-to-Back SETUP Packets Received	0x0
5	R/W1C	StsPhseRcvd	Status Phase Received for Control Write (StsPhseRcvd) This interrupt is valid only for Control OUT endpoints. Values: 0x0 (INACTIVE): No Status Phase Received for Control Write 0x1 (ACTIVE): Status Phase Received for Control Write	0x0
4	R/W1C	OUTTknEPdis	OUT Token Received When Endpoint Disabled (OUTTknEPdis) Applies only to control OUT endpoints. Values: 0x0 (INACTIVE): No OUT Token Received When Endpoint Disabled 0x1 (ACTIVE): OUT Token Received When Endpoint Disabled	0x0
3	R/W1C	SetUp	SETUP Phase Done (SetUp) Applies to control OUT endpoints only. Values: 0x0 (INACTIVE): No SETUP Phase Done 0x1 (ACTIVE): SETUP Phase Done	0x0
2	R/W1C	AHBErr	AHB Error (AHBErr) Applies to IN and OUT endpoints. Values: 0x0 (INACTIVE): No AHB Error Interrupt 0x1 (ACTIVE): AHB Error interrupt	0x0
1	R/W1C	EPDisbld	Endpoint Disabled Interrupt (EPDisbld) Applies to IN and OUT endpoints. Values: 0x0 (INACTIVE): No Endpoint Disabled Interrupt 0x1 (ACTIVE): Endpoint Disabled Interrupt	0x0
0	R/W1C	XferCompl	Transfer Completed Interrupt (XferCompl) Applies to IN and OUT endpoints. Values: 0x0 (INACTIVE): No Transfer Complete Interrupt 0x1 (ACTIVE): Transfer Complete Interrupt	0x0

DOEPTSIZE0

This register contains the Transfer Size for the OUT Endpoint 0 of the Device controller.

Offset:0xb10 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31	R	RESERVED	RESERVED	0x0

Bits	Access	Name	Description	Reset
30:29	R/W	SUPCnt	SETUP Packet Count (SUPCnt) This field specifies the number of back-to-back SETUP data packets the endpoint can receive. Values: 0x1 (ONEPACKET): 1 packet 0x2 (TWOPACKET): 2 packets 0x3 (THREEPACKET): 3 packets	0x0
28:20	R	RESERVED1	RESERVED	0x0
19	R/W	PktCnt	Packet Count (PktCnt) This field is decremented to zero after a packet is written into the RxFIFO.	0x0
18:7	R	RESERVED2	RESERVED	0x0
6:0	R/W	XferSize	Transfer Size (XferSize) Indicates the transfer size in bytes for endpoint 0. The core interrupts the application only after it has exhausted the transfer size amount of data. The transfer size can be Set to the maximum packet size of the endpoint, to be interrupted at the end of each packet. The core decrements this field every time a packet is read from the RxFIFO and written to the external memory.	0x0

DOEPDMA0

This register contains the DMA Address for the OUT Endpoint 0 of the Device controller.

Offset:0xb14 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:0	DMAAddr	R/W	Holds the start address of the external memory for storing or fetching endpoint data.	DMA registers are implemented in SPRAM. The reset values of these registers in simulation can be 0/X depending upon the SPRAM model.

DOEPDMAB0

This register contains the DMA Buffer Address for the OUT Endpoint 0 of the Device controller.

Offset:0xb1c Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:0	R	DMABufferAddr	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.	DMA registers are implemented in SPRAM. The reset values of these registers in simulation can be 0/X depending upon the SPRAM model.

DOEPCTLi

This register is used to control the characteristics of OUT Endpoint i of the Device controller.

Offset:0xB00 + i*20 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset

Bits	Access	Name	Description	Reset
31	R/W1S	EPEna	Endpoint Enable (EPEna) Applies to IN and OUT endpoints. Values: 0x0 (INACTIVE): No Action 0x1 (ACTIVE): Enable Endpoint	0x0
30	R/W1S	EPDis	Endpoint Disable (EPDis) Applies to IN and OUT endpoints. Values: 0x0 (INACTIVE): No Action 0x1 (ACTIVE): Disable Endpoint	0x0
29	W	SetD1PID	Set DATA1 PID (SetD1PID) Values: 0x0 (DISABLED): Disables Set DATA1 PID or Do not force Odd Frame 0x1 (ENABLED): Set Endpoint Data PID to DATA1 or Sets EO_FrNum field to odd (micro)Frame	0x0
28	W	SetD0PID	Set DATA0 PID (SetD0PID) Values: 0x0 (DISABLED): Disables Set DATA0 PID or Do not force Even Frame 0x1 (ENABLED): Set Endpoint Data PID to DATA0 or Sets EO_FrNum field to odd (micro)Frame	0x0
27	W	SNAK	Set NAK (SNAK) A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also set this bit for an endpoint after a SETUP packet is received on that endpoint. Values: 0x0 (INACTIVE): No Set NAK 0x1 (ACTIVE): Set NAK	0x0
26	W	CNAK	Clear NAK (CNAK) A write to this bit clears the NAK bit for the endpoint. Values: 0x0 (INACTIVE): No Clear NAK 0x1 (ACTIVE): Clear NAK	0x0
25:22			Reserved Field: Yes	
21	R/W1S	Stall	STALL Handshake (Stall) Applies to non-control, non-isochronous IN and OUT endpoints only. Values: 0x0 (INACTIVE): STALL All non-active tokens 0x1 (ACTIVE): STALL All Active Tokens	0x0
20	R/W	Snp	RESERVED	0x0

Bits	Access	Name	Description	Reset
19:18	R/W	EPTYPE	Endpoint Type (EPTYPE) This is the transfer type supported by this logical endpoint. Values: 0x0 (CONTROL): Control 0x1 (ISOCHRONOUS): Isochronous 0x2 (BULK): Bulk 0x3 (INTERRUPT): Interrupt	0x0
17	R	NAKSTS	NAK Status (NAKSTS) Indicates the following: Values: 0x0 (NONNAK): The core is transmitting non-NAK handshakes based on the FIFO status 0x1 (NAK): The core is transmitting NAK handshakes on this endpoint	0x0
16	R	DPID	Endpoint Data PID (DPID) Applies to interrupt/bulk IN and OUT endpoints only. Values: 0x0 (INACTIVE): Endpoint Data PID not active 0x1 (ACTIVE): Endpoint Data PID active	0x0
15	R/W	USBACTEP	USB Active Endpoint (USBACTEP) Values: 0x0 (DISABLED): Not Active 0x1 (ENABLED): USB Active Endpoint	0x0
14:11			Reserved Field: Yes	
10:0	R/W	MPS	Maximum Packet Size (MPS) The application must program this field with the maximum packet size for the current logical endpoint. This value is in bytes.	0x0

DOEPINTi

This register contains the interrupts for the OUT Endpoint i of the Device controller. Offset: 0xB08 + i*20 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:16			Reserved Field: Yes	
15	R/W1C	StupPktRcvd	Setup Packet Received Applicable for Control OUT Endpoints in only in the Buffer DMA Mode 0x0 (NOT_RCVD): No Setup packet received 0x1 (RCVD): Setup packet received	0x0
14	R/W1C	NYETIntrpt	NYET Interrupt (NYETIntrpt) Values: 0x0 (INACTIVE): No NYET interrupt 0x1 (ACTIVE): NYET Interrupt	0x0

Bits	Access	Name	Description	Reset
13	R/W1C	NAKIntrpt	NAK Interrupt (NAKInterrupt) The core generates this interrupt when a NAK is transmitted or received by the device. Values: 0x0 (INACTIVE): No NAK interrupt 0x1 (ACTIVE): NAK Interrupt	0x0
12	R/W1C	BbleErr	NAK Interrupt (BbleErr) Values: 0x0 (INACTIVE): No BbleErr interrupt 0x1 (ACTIVE): BbleErr interrupt	0x0
11	R/W1C	PktDrpSts	Packet Drop Status (PktDrpSts) This bit indicates to the application that an ISOC OUT packet has been dropped. This bit does not have an associated mask bit and does not generate an interrupt. Values: 0x0 (INACTIVE): No interrupt 0x1 (ACTIVE): Packet Drop Status interrupt	0x0
10			Reserved Field: Yes	
9	R/W1C	BNAIntr	BNA (Buffer Not Available) Interrupt (BNAIntr) Values: 0x0 (INACTIVE): No BNA interrupt 0x1 (ACTIVE): BNA interrupt	0x0
8	R/W1C	OutPktErr	OUT Packet Error (OutPktErr) Applies to OUT endpoints Only Values: 0x0 (INACTIVE): No OUT Packet Error 0x1 (ACTIVE): OUT Packet Error	0x0
7			Reserved Field: Yes	
6	R/W1C	Back2BackSETup	Back-to-Back SETUP Packets Received (Back2BackSETup) Applies to Control OUT endpoints only. Values: 0x0 (INACTIVE): No Back-to-Back SETUP Packets Received 0x1 (ACTIVE): Back-to-Back SETUP Packets Received	0x0
5	R/W1C	StsPhseRcvd	Status Phase Received for Control Write (StsPhseRcvd) This interrupt is valid only for Control OUT endpoints. Values: 0x0 (INACTIVE): No Status Phase Received for Control Write 0x1 (ACTIVE): Status Phase Received for Control Write	0x0
4	R/W1C	OUTTknEPdis	OUT Token Received When Endpoint Disabled (OUTTknEPdis) Applies only to control OUT endpoints. Values: 0x0 (INACTIVE): No OUT Token Received When Endpoint Disabled 0x1 (ACTIVE): OUT Token Received When Endpoint Disabled	0x0

Bits	Access	Name	Description	Reset
3	R/W1C	SetUp	SETUP Phase Done (SetUp) Applies to control OUT endpoints only. Values: 0x0 (INACTIVE): No SETUP Phase Done 0x1 (ACTIVE): SETUP Phase Done	0x0
2	R/W1C	AHBErr	AHB Error (AHBErr) Applies to IN and OUT endpoints. Values: 0x0 (INACTIVE): No AHB Error Interrupt 0x1 (ACTIVE): AHB Error interrupt	0x0
1	R/W1C	EPDisbld	Endpoint Disabled Interrupt (EPDisbld) Applies to IN and OUT endpoints. This bit indicates that the endpoint is disabled per the application's request. Values: 0x0 (INACTIVE): No Endpoint Disabled Interrupt 0x1 (ACTIVE): Endpoint Disabled Interrupt	0x0
0	R/W1C	XferCompl	Transfer Completed Interrupt (XferCompl) Applies to IN and OUT endpoints. Values: 0x0 (INACTIVE): No Transfer Complete Interrupt 0x1 (ACTIVE): Transfer Complete Interrupt	0x0

DOEPTSIZE_i

This register contains the Transfer Size for the OUT Endpoint *i* of the Device controller. Offset: 0xB10 + *i**20 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
31	R	RESERVED	RESERVED	0x0
30:29	R	RxDPID	RxDPID Applies to isochronous OUT endpoints only. This is the data PID received in the last packet for this endpoint. Values: 0x0 (DATA0): DATA0 0x1 (DATA2PACKET1): DATA2 or 1 packet 0x2 (DATA1PACKET2): DATA1 or 2 packets 0x3 (MDATAPACKET3): MDATA or 3 packets	0x0
28:19	R/W	PktCnt	Packet Count (PktCnt) This field is decremented to zero after a packet is written into the Rx FIFO. Value After Reset: 0x0	0x0

Bits	Access	Name	Description	Reset
18:0	R/W	XferSize	Transfer Size (XferSize) Indicates the transfer size in bytes for the endpoint. The core interrupts the application only after it has exhausted the transfer size amount of data. The transfer size can be Set to the maximum packet size of the endpoint, to be interrupted at the end of each packet. The core decrements this field every time a packet is read from the RxFIFO and written to the external memory.	0x0

DOEPDMAi

This register contains the DMA Address for the OUT Endpoint i of the Device controller.

Offset:0xB14 + i*20 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:0	R/W	DMAAddr	Holds the start address of the external memory for storing or fetching endpoint data. Note: For control endpoints, this field stores control OUT data packets as well as SETUP transaction data packets. When more than three SETUP packets are received back-to-back, the SETUP data packet in the memory is overwritten. This register is incremented on every AHB transaction. The application can give only a DWORD-aligned address. When Scatter/Gather DMA mode is enabled, this field indicates the base pointer for the descriptor list.	DMA registers are implemented in SPRAM. The reset values of these registers in simulation can be 0/X depending upon the SPRAM model.

DOEPDMABi

This register contains the DMA Buffer Address for the OUT Endpoint i of the Device controller.

Offset:0xB1C + i*20 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:0	R	DMABufferAddr	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.	DMA registers are implemented in SPRAM. The reset values of these registers in simulation can be 0/X depending upon the SPRAM model.

PCGCCTL

This register is used to control the Power and Clock Gating characteristics of the controller.

Offset:0xe00 Total Reset Value:0x000b_0000

Bits	Access	Name	Description	Reset
31:8	R	RESERVED	RESERVED	

Bits	Access	Name	Description	Reset
7	R	L1Suspended	L1 Deep Sleep Indicates that the PHY is in deep sleep when in L1 state. Values: 0x0 (INACTIVE): Non Deep Sleep 0x1 (ACTIVE): Deep Sleep	0x0
6	R	PhySleep	PHY In Sleep Indicates that the PHY is in Sleep State. Values: 0x0 (INACTIVE): Phy not in Sleep state 0x1 (ACTIVE): Phy in Sleep state	0x0
5	R/W	Enbl_L1Gating	Enable Sleep Clock Gating Values: 0x0 (DISABLED): The PHY clock is not gated in Sleep state 0x1 (ENABLED): The Core internal clock gating is enabled in Sleep state	0x0
4			Reserved Field: Yes	
3	R/W	RstPdownModule	Reset Power-Down Modules (RstPdownModule) This bit is valid only in Partial Power-Down mode. Values: 0x0 (ON): Power is turned on 0x1 (OFF): Power is turned off	0x0
2:1	R	RESERVED	RESERVED	
0	R/W	StopPclk	Stop Pclk (StopPclk) Values: 0x0 (DISABLED): Disable Stop Pclk 0x1 (ENABLED): Enable Stop Pclk	0x0

PCGCCTL1

This register is used to control the Power and Clock Gating characteristics of the controller.

Offset:0xe04 Total Reset Value:0x0000_0000 RESERVED

12.6.5.2 USB Ctrl Register in SoC

OTG0_PROBE0

Debug Probes in AHB Clock Domain.

Offset:0x10 Total Reset Value:0x0000_0000

Table 12-6-2 Debug Probes in AHB Clock Domain

Bit Position	Internal Signal	Valid Modes
internal_probes[61:60]	hclk_state_cs	OTG_EN_ACG
internal_probes[59]	hclk_gate_en	OTG_EN_ACG
internal_probes[58]	dsch_chdis_push	DESC_DMA_EN
internal_probes[57]	clr_last_pkt_xmit	DESC_DMA_EN
internal_probes[56]	dsch_xmit_sel	DESC_DMA_EN

Bit Position		Internal Signal		Valid Modes	
internal_probes[55]		dsch_reqq_push		DESC_DMA_EN	
internal_probes[54:47]		dsch_tkn_req		DESC_DMA_EN	
internal_probes[46]		dsch_lp_done		DESC_DMA_EN	
internal_probes[45:44]		dsch_lp_req_type		DESC_DMA_EN	
internal_probes[43]		dsch_lp_req		DESC_DMA_EN	
internal_probes[42:28]		current_state		DESC_DMA_EN	
internal_probes[27]		tx_trans_start		All	
internal_probes[26]		tx_trans_end		All	
internal_probes[25]		txf_rd_rollover		All	
internal_probes[24]		rx_sts_qual		All	
internal_probes[23]		rx_trans_start		All	
internal_probes[22:18]		rxf_numpkts[4:0]		All	
internal_probes[17]		rx_data		All	
internal_probes[16]		rx_nonpktsts		All	
internal_probes[15:13]		rx_pktsts[2:0]		All	
internal_probes[12]		rx_sts		All	
Bits	Access	Name	Description	Reset	
31:0	RO	OTG0_PROBE0	otg0_internal_probes[31:0]	0x0	

OTG0_PROBE1

Debug Probes in AHB Clock Domain.

Offset:0x14 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
29:0	RO	OTG0_PROBE1	otg0_internal_probes[61:32]	0x0

OTG0_PROBE0

Debug Probes in PHY Clock Domain.

Offset:0x18 Total Reset Value:0x0000_0000

Table 12- Debug Probes in PHY Clock Domain

Bit Position		Internal Signal		Valid Modes	
internal_probes_p[218:217]		phy_state_cs		OTG_EN_ACG	
internal_probes_p[216]		phy_clk_gate_en		OTG_EN_ACG	
internal_probes_p[215]		dbg_iso_tkn_end		All	
internal_probes_p[214:211]		curep[3:0]		All	
internal_probes_p[210]		txf_reqq_pop_dev		All	

Bit Position		Internal Signal		Valid Modes
internal_probes_p[209]		src_pop		All
internal_probes_p[208]		sink_sts_push_dev		All
internal_probes_p[207]		sts_push		All
internal_probes_p[206]		short_pkt		All
internal_probes_p[205]		gout_nak_eff		All
internal_probes_p[204]		rewind_tobe_compl		All
internal_probes_p[203]		data_tobe_compl		All
internal_probes_p[202]		xfer_tobe_compl		All
internal_probes_p[201:198]		pkt_epnum		All
internal_probes_p[197:195]		pkt_sts		All
internal_probes_p[194]		txf_reqq_pop_hst		All
internal_probes_p[193]		sink_sts_push_hst		All
internal_probes_p[192:172]		p2hd_reg_status_hst		All
internal_probes_p[171]		ti_tkn_req		All
internal_probes_p[170:121]		ti_tkn_info		All
internal_probes_p[120:108]		current_state		All
internal_probes_p[107]		upi_rxvld		All
internal_probes_p[106]		upi_rxvldh		All
Bits	Access	Name	Description	Reset
31:0	RO	OTG0_PROBEP0	otg0_internal_probes_p[31:0]	0x0

OTG0_PROBEP1

Debug Probes in PHY Clock Domain.

Offset:0x1c Total Reset Value:0x0000_0000

Bits	Access	Name	Description
31:0	RO	OTG0_PROBEP1	otg0_internal_probes_p[63:32]

OTG0_PROBEP2

Debug Probes in PHY Clock Domain.

Offset:0x20 Total Reset Value:0x0000_0000

Bits	Access	Name	Description
31:0	RO	OTG0_PROBEP2	otg0_internal_probes_p[95:64]

OTG0_PROBEP3

Debug Probes in PHY Clock Domain.

Offset:0x24 Total Reset Value:0x0000_0000

Bits	Access	Name	Description

Bits	Access	Name	Description
31:0	RO	OTG0_PROBEP3	otg0_internal_probes_p[127:96]

OTG0_PROBEP4

Debug Probes in PHY Clock Domain.

Offset:0x28 Total Reset Value:0x0000_0000

Bits	Access	Name	Description
31:0	RO	OTG0_PROBEP4	otg0_internal_probes_p[159:128]

OTG0_PROBEP5

Debug Probes in PHY Clock Domain.

Offset:0x2c Total Reset Value:0x0000_0000

Bits	Access	Name	Description
31:0	RO	OTG0_PROBEP5	otg0_internal_probes_p[191:160]

OTG0_PROBEP6

Debug Probes in PHY Clock Domain.

Offset:0x30 Total Reset Value:0x0000_0000

Bits	Access	Name	Description
26:0	RO	OTG0_PROBEP6	otg0_internal_probes_p[218:192]

OTG0_STRAP

USB Ctrl reg in SoC.

Offset:0x34 Total Reset Value:0x0000_0014

Access	Name	Description	Reset
RW	otg0_portreset0	1: The transmit and receive finite state machines (FSMs) are reset, and the line_state logic combinatorially reflects the state of the single-ended receivers. 0: The transmit and receive FSMs are operational, and the line_state logic becomes sequential after 11 PHYCLOCK0 cycles	0x0
RW	otg0_COMMONONN	1: In Suspend mode, the REFCLK_LOGIC, Bias, and PLL blocks are powered down. In Sleep mode, the Bias and PLL blocks are powered down. 0: In Suspend or Sleep modes, the REFCLK_LOGIC, Bias, and PLL blocks remain powered. With this setting, the input reference clock must remain on and valid during suspend or sleep.	0x0
RO	otg0_sof_sent_rcvd_tgl	In Host mode, this signal toggles every time an SOF is successfully transmitted. In Device mode, this signal toggles only when a valid SOF token is received from the USB host.	0x0
RO	otg0_sof_toggle	In Host mode, this signal toggles every time an SOF is generated. In Device mode, this signal toggles every time an SOF token is received from the USB host or when an SOF token is missed at the start of frame.	0x0

Access	Name	Description	Reset
RO	otg0_chep_number	0000: Channel/Endpoint 0 0001: Channel/Endpoint 1 0010: Channel/Endpoint 2 and so on up to, 1111: Channel/Endpoint 15	0x0
RO	otg0_chep_last_trans	0: Current DMA write is not the last transaction 1: Current DMA write is the last transaction	0x0
RO	otg0_int_dma_done	0: DMA transfer not done 1: Current DMA transfer from the controller is done	0x0
RO	otg0_int_dma_req	0: No DMA transfer 1: Active DMA write transfer from the AHB master.	0x0
RW	otg0_FSEL	select CLKCORE as ref clk	0x2
RW	otg0_REFCLKSEL	24M	0x2
RW	otg0_sys_dma_done	0: Data write not complete. 1: Data Write complete in the system memory for the current DMA write-transfer from the controller	0x0

OTG1_PROBE0

Debug Probes in AHB Clock Domain.

Offset:0x40 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:0	RO	OTG1_PROBE0	otg1_internal_probes[31:0]	0x0

OTG1_PROBE1

Debug Probes in AHB Clock Domain.

Offset:0x44 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
29:0	RO	OTG1_PROBE1	otg1_internal_probes[61:32]	0x0

OTG1_PROBEP0

Debug Probes in PHY Clock Domain.

Offset:0x48 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
31:0	RO	OTG1_PROBEP0	otg1_internal_probes_p[31:0]	0x0

OTG1_PROBEP1

Debug Probes in PHY Clock Domain.

Offset:0x4c Total Reset Value:0x0000_0000

Bits	Access	Name	Description
31:0	RO	OTG1_PROBEP1	otg1_internal_probes_p[63:32]

OTG1_PROBEP2

Debug Probes in PHY Clock Domain.

Offset:0x50 Total Reset Value:0x0000_0000

Bits	Access	Name	Description
31:0	RO	OTG0_PROBEP2	otg0_internal_probes_p[95:64]

OTG1_PROBEP3

Debug Probes in PHY Clock Domain.

Offset:0x54 Total Reset Value:0x0000_0000

Bits	Access	Name	Description
31:0	RO	OTG1_PROBEP3	otg1_internal_probes_p[127:96]

OTG1_PROBEP4

Debug Probes in PHY Clock Domain.

Offset:0x58 Total Reset Value:0x0000_0000

Bits	Access	Name	Description
31:0	RO	OTG1_PROBEP4	otg1_internal_probes_p[159:128]

OTG1_PROBEP5

Debug Probes in PHY Clock Domain.

Offset:0x5c Total Reset Value:0x0000_0000

Bits	Access	Name	Description
31:0	RO	OTG1_PROBEP5	otg1_internal_probes_p[191:160]

OTG1_PROBEP6

Debug Probes in PHY Clock Domain.

Offset:0x60 Total Reset Value:0x0000_0000

Bits	Access	Name	Description
26:0	RO	OTG1_PROBEP6	otg1_internal_probes_p[218:192]

OTG1_STRAP

USB Ctrl reg in SoC.

Offset:0x64 Total Reset Value:0x0000_0014

Access	Name	Description	Reset
RW	otg1_portreset0	1: The transmit and receive finite state machines (FSMs) are reset, and the line_state logic combinatorially reflects the state of the single-ended receivers. 0: The transmit and receive FSMs are operational, and the line_state logic becomes sequential after 11 PHYCLOCK0 cycles	0x0

Access	Name	Description	Reset
RW	otg1_COMMONONN	1: In Suspend mode, the REFCLK_LOGIC, Bias, and PLL blocks are powered down. In Sleep mode, the Bias and PLL blocks are powered down. 0: In Suspend or Sleep modes, the REFCLK_LOGIC, Bias, and PLL blocks remain powered. With this setting, the input reference clock must remain on and valid during suspend or sleep.	0x0
RO	otg1_sof_sent_rcvd_tgl	In Host mode, this signal toggles every time an SOF is successfully transmitted. In Device mode, this signal toggles only when a valid SOF token is received from the USB host.	0x0
RO	otg1_sof_toggle	In Host mode, this signal toggles every time an SOF is generated. In Device mode, this signal toggles every time an SOF token is received from the USB host or when an SOF token is missed at the start of frame.	0x0
RO	otg1_chep_number	0000: Channel/Endpoint 0 0001: Channel/Endpoint 1 0010: Channel/Endpoint 2 and so on up to, 1111: Channel/Endpoint 15	0x0
RO	otg1_chep_last_trans	0: Current DMA write is not the last transaction 1: Current DMA write is the last transaction	0x0
RO	otg1_int_dma_done	0: DMA transfer not done 1: Current DMA transfer from the controller is done	0x0
RO	otg1_int_dma_req	0: No DMA transfer 1: Active DMA write transfer from the AHB master.	0x0
RW	otg1_FSEL	select CLKCORE as ref clk	0x2
RW	otg1_REFCLKSEL	24M	0x2
RW	otg1_sys_dma_done	0: Data write not complete. 1: Data Write complete in the system memory for the current DMA write-transfer from the controller	0x0

USB0_TEST_CTL2

USB Ctrl reg in SoC.

Offset:0x78 Total Reset Value:0x0000_0000

Access	Name	Description	Reset
RW	otg0_SUSPENDM0	1: Normal operating mode 0: Suspend mode	0x0

USB0_TEST_CTL3

USB Ctrl reg in SoC.

Offset:0x7c Total Reset Value:0x0000_0020

Bits	Access	Name	Description	Reset
16	RW	otg0_TESTBURNIN	Burn-in Test Enable	0x0

Bits	Access	Name	Description	Reset
15:14	RW	otg0_OPMODE0	11: Normal operation without SYNC or EOP generation. If the XCVRSEL0[1:0] bus is not set to 2'b00 while OPMODE0[1:0] is set to 2'b11, USB 2.0 femtoPHY behavior is undefined. 10: Disable bit stuffing and NRZI encoding 01: Non-Driving 00: Normal	0x0
13	RW	otg0_TERMSEL0	This controller signal sets the USB 2.0 femtoPHY's terminations to FS or HS. 1: Full-speed terminations are enabled. 0: High-speed terminations are enabled.	0x0
12	RW	otg0_WORDINTERFACE0	UTMI+ Data Bus Width and Clock Select This controller signal selects the data bus width of the UTMI+ data buses. 1: 16-bit data interface (PHYCLOCK0 frequency is 30 MHz) 0: 8-bit data interface (PHYCLOCK0 frequency is 60 MHz)	0x0
11:10	RW	otg0_XCVRSEL0	Transceiver Select This controller bus selects the HS, FS, or LS Transceiver. 11: Sends an LS packet on an FS bus or receives an LS packet. 10: LS Transceiver 01: FS Transceiver 00: HS Transceiver	0x0
9	RW	otg0_DPPULLDOWN0	D+ Pull-Down Resistor Enable This controller signal controls the pull-down resistance on the D+ line. 1: The pull-down resistance on D+ is enabled. 0: The pull-down resistance on D+ is disabled.	0x0
8	RW	otg0_DMPULLDOWN0	D- Pull-Down Resistor Enable This controller signal controls the pull-down resistance on the D- line. 1: The pull-down resistance on D- is enabled. 0: The pull-down resistance on D- is disabled.	0x0
7	RW	otg0_TXBITSTUFFEN0	Low-Byte Transmit Bit-Stuffing Enable This controller signal controls bit stuffing on DATAIN0[7:0] when OPMODE0[1:0] = 2'b11. 1: Bit stuffing is enabled. 0: Bit stuffing is disabled.	0x0

Bits	Access	Name	Description	Reset
6	RW	otg0_TXBITSTUFFENH0	High-Byte Transmit Bit-Stuffing Enable This controller signal controls bit stuffing on DATAINH0[7:0] when OPMODE0[1:0] = 2'b11. 1: Bit stuffing is enabled. 0: Bit stuffing is disabled.	0x0
5	RW	otg0_VBUSVLDEXT0	External VBUS Valid Indicator This signal is valid in Device mode and only when the VBUSVLDEXTSEL0 signal is set to 1'b1. VBUSVLDEXT0 indicates whether the VBUS signal on the USB cable is valid. In addition, VBUSVLDEXT0 enables the pull-up resistor on the D+ line. VBUSVLDEXT0 does not change the OTGSESSVLD0 output. 1: The VBUS signal is valid, and the pull-up resistor on D+ is enabled. 0: The VBUS signal is not valid, and the pull-up resistor on D+ is disabled. In Host mode, this input is not used and can be tied to 1'b0.	0x1
4	RW	otg0_IDPULLUP0	Analog ID Input Sample Enable This controller signal controls ID line sampling. 1: ID pin sampling is enabled, and the IDDIG0 output is valid. 0: ID pin sampling is disabled, and the IDDIG0 output is not valid.	0x0
3	RO	otg0_IDDIG0	Mini A/B Plug Indicator This controller signal indicates whether the connected plug is a mini-A or mini-B plug. This signal must be valid within 20 ms after the IDPULLUP0 signal is set to 1'b1. 1: The connected plug is a mini-B plug. 0: The connected plug is a mini-A plug.	0x0
2	RO	otg0_AVALID0	A-Device Session Valid Indicator	0x0
1	RO	otg0_BVALID0	B-Device Session Valid Indicator	0x0

Bits	Access	Name	Description	Reset
0	RO	otg0_OTGSESSVLD0	OTG Device Session Valid Indicator (Low Voltage) This controller signal is output from the USB 2.0 femtoPHY's Session Valid comparator and indicates whether the voltage on VBUS is below the OTG Device Session Valid threshold. 1: The voltage on VBUS is above the OTG Device Session Valid threshold. 0: The voltage on VBUS is below the OTG Device Session Valid threshold.	0x0

USB0_TEST_CTL4

USB Ctrl reg in SoC.

Offset:0x80 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
23:16	RW	otg0_DATAIN0	UTMI+ Data Input Bus, Low Byte	0x0
15:8	RW	otg0_DATAINH0	UTMI+ Data Input Bus, High Byte	0x0
1	RW	otg0_TXVALID0	UTMI+ Transmit Valid Indicator, Low Byte This controller signal indicates that data on the DATAIN0[7:0] bus is valid.	0x0
0	RW	otg0_TXVALIDH0	UTMI+ Transmit Valid Indicator, High Byte This controller signal indicates that data on the DATAINH0[7:0] bus (high byte) is valid.	0x0

USB0_TEST_CTL5

USB Ctrl reg in SoC.

Offset:0x84 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
0	RW	otg0_FSXCVROWNER0	UTMI+/Serial Interface Select This controller signal enables the UTMI+ or serial interface. 1: The TXENABLEN0, FSDATAEXT0, and FSSE0EXT0 inputs drive USB 2.0 femtoPHY data output onto the D+ and D- lines. Data that the USB 2.0 femtoPHY receives from the D+ and D- lines appears on the FSVMINUS0 and FSVPLUS0 outputs. 0: Data on the D+ and D- lines is transmitted and received through the UTMI+. If FSXCVROWNER0 is not used, set it to 1'b0.	0x0

USB0_TEST_CTL6

USB Ctrl reg in SoC.

Offset:0x88 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
8	RW	otg0DCDENB0	Enhanced Data Contact Detection Enable Enables current sourcing on the D+ line. For special applications, the IDP_SRC control and the RDM_DWN control are separated. For standard DCD operation in accordance with the Battery Charger v1.2 specification, simultaneously set DCDENB0=1'b1 (to enable IDP_SRC) and set DMPULLDOWN0=1'b1 (to enable RDM_DWN). Set both signals to 1'b0 when DCD is completed. 1: IDP_SRC current is sourced onto DP0 0: IDP_SRC current is disabled	0x0
7	RW	otg0VDATDETENB0	Battery Charging Attach/Connect Detection Enable Enables or disables attach/connect detection. 1: Data detect voltage (CHG_DET) is enabled. 0: Data detect voltage (CHG_DET) is disabled.	0x0
6	RW	otg0VDATSRCEB0	Battery Charging Sourcing Select Enables or disables sourcing for battery charging. 1: Data source voltage (VDAT_SRC) is enabled. 0: Data source voltage (VDAT_SRC) is disabled.	0x0
5	RW	otg0CHRGSEL0	Battery Charging Source Select Determines whether current is sourced onto or sunk from DP0 or DM0. The definitions shown below apply when CHGRSRCPUENB0[1:0]=2'b00. See CHGRSRCPUENB0[1:0] for other modes. 1: Data source voltage (VDAT_SRC) is sourced onto DM0 and sunk from DP0. 0: Data source voltage (VDAT_SRC) is sourced onto DP0 and sunk from DM0.	0x0
4	RW	otg0BYPASSSEL0	Transmitter Digital Bypass Select Enables/disables Transmitter Digital Bypass mode. The standard FS UART mode is selected by setting HSXCVREXTCTL0 = 1'b0, and a proprietary HS test mode is selected by setting HSXCVREXTCTL0 = 1'b1. 1: Transmitter Digital Bypass mode is enabled. 0: Transmitter Digital Bypass mode is disabled	0x0

Bits	Access	Name	Description	Reset
3	RW	otg0BYPASSDPEN0	DP0 Transmitter Digital Bypass Enable Enables/disables the DP0 FS/LS driver (if HSXCVREXTCTL0 = 1'b0), or the DP0 and DM0 HS driver (if HSXCVREXTCTL0 = 1'b1) in Transmitter Digital Bypass mode. FS mode is the standard UART mode. HS mode is provided for proprietary HS tests. To use either of these modes, BYPASSSEL0 must be 1'b1. 1: DP0 FS/LS driver (if HSXCVREXTCTL0 = 1'b0), or DP0 and DM0 HS driver (if HSXCVREXTCTL0 = 1'b1) is enabled and driven with the BYPASSDPDATA0 signal. 0: DP0 FS/LS driver (if HSXCVREXTCTL0 = 1'b0), or DP0 and DM0 HS driver (if HSXCVREXTCTL0 = 1'b1) is disabled in Transmitter Digital Bypass mode.	0x0
2	RW	otg0BYPASSDMEN0	DM0 Transmitter Digital Bypass Enable Enables/disables the DM0 FS/LS driver (if HSXCVREXTCTL0 = 1'b0) in Transmitter Digital Bypass mode. FS mode is the standard UART mode. To use this mode, BYPASSSEL0 must be 1'b1. 1: DM0 FS/LS driver (if HSXCVREXTCTL0 = 1'b0) is enabled and driven with the BYPASSDPDATA0 signal. 0: DM0 FS/LS driver (if HSXCVREXTCTL0 = 1'b0) is disabled in Transmitter Digital Bypass mode.	0x0
1	RW	otg0BYPASSDPDATA0	Data for DP0 Transmitter Digital Bypass In Transmitter Digital Bypass mode, this signal provides the data that is transmitted on DP0 (when HSXCVREXTCTL0 = 1'b0), or DP0 and DM0 (when HSXCVREXTCTL0 = 1'b1). To use this signal, BYPASSSEL0 must be 1'b1. When HSXCVREXTCTL0 = 1'b0, the FS/LS driver is used and outputs the following FS/LS states: 1: DP0 FS/LS driver drives to a high state. 0: DP0 FS/LS driver drives to a low state	0x0

Bits	Access	Name	Description	Reset
0	RW	otg0BYPASSDMDATA0	Data for DM0 Transmitter Digital Bypass In Transmitter Digital Bypass mode, this signal provides the data that is transmitted on DM0. To use this signal, BYPASSSELO must be 1'b1. When HSXCVREXTCTL0 = 1'b0, the FS/LS driver is used and outputs the following FS/LS states: 1: DM0 FS/LS driver drives to a high state. 0: DM0 FS/LS driver drives to a low state.	0x0

USB1_TEST_CTL2

USB Ctrl reg in SoC.

Offset:0x98 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
0	RW	otg1_SUSPENDM0	Same as otg0	0x0

USB1_TEST_CTL3

USB Ctrl reg in SoC.

Offset:0x9c Total Reset Value:0x0000_0020

Bits	Access	Name	Description	Reset
16	RW	otg1_TESTBURNIN	Same as otg0	0x0
15:14	RW	otg1_OPMODE0	Same as otg0	0x0
13	RW	otg1_TERMSELO	Same as otg0	0x0
12	RW	otg1_WORDINTERFACE0	Same as otg0	0x0
11:10	RW	otg1_XCVRSELO	Same as otg0	0x0
9	RW	otg1_DPPULLDOWN0	Same as otg0	0x0
8	RW	otg1_DMPULLDOWN0	Same as otg0	0x0
7	RW	otg1_TXBITSTUFFEN0	Same as otg0	0x0
6	RW	otg1_TXBITSTUFFENH0	Same as otg0	0x0
5	RW	otg1_VBUSVLDEXT0	Same as otg0	0x1
4	RW	otg1_IDPULLUP0	Same as otg0	0x0
3	RO	otg1_IDDIG0	Same as otg0	0x0
2	RO	otg1_AVALID0	Same as otg0	0x0
1	RO	otg1_BVALID0	Same as otg0	0x0
0	RO	otg1_OTGSESSVLD0	Same as otg0	0x0

USB1_TEST_CTL4

USB Ctrl reg in SoC.

Offset:0xa0 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
23:16	RW	otg1_DATAIN0	Same as otg0	0x0
15:8	RW	otg1_DATAINH0	Same as otg0	0x0
1	RW	otg1_TXVALID0	Same as otg0	0x0
0	RW	otg1_TXVALIDH0	Same as otg0	0x0

USB1_TEST_CTL5
 USB Ctrl reg in SoC.
 Offset:0xa4 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
0	RW	otg1_FSXCVROWNER0	Same as otg0	0x0

USB1_TEST_CTL6
 USB Ctrl reg in SoC.
 Offset:0xa8 Total Reset Value:0x0000_0000

Bits	Access	Name	Description	Reset
8	RW	otg1_DCDENB0	Same as otg0	0x0
7	RW	otg1_VDATDETENB0	Same as otg0	0x0
6	RW	otg1_VDATSRCENB0	Same as otg0	0x0
5	RW	otg1_CHRGSEL0	Same as otg0	0x0
4	RW	otg1_BYPASSSEL0	Same as otg0	0x0
3	RW	otg1_BYPASSDPEN0	Same as otg0	0x0
2	RW	otg1_BYPASSDMEN0	Same as otg0	0x0
1	RW	otg1_BYPASSDPDATA0	Same as otg0	0x0
0	RW	otg1_BYPASSDMDATA0	Same as otg0	0x0

USB0_CTL0
 USB Ctrl reg in SoC.
 Offset:0xac Total Reset Value:0x0003_8818

Bits	Access	Name	Description	Reset
23:22	RW	otg0_PLLITUNE	PLL Integral Path Tune Reserved This signal should be set to the Design default. 11: 2.0x 10: 1.5x 01: 1.0x 00: 0.5x, Design default	0x0

Bits	Access	Name	Description	Reset
21:18	RW	otg0_PLLPTUNE	PLL Proportional Path Tune Reserved This signal should be set to the Design default. 1111: 5.5x 1110: 5x 1101: 4.5x 1100: 4x, Design default 1011: 4.5x 1010: 4x 1001: 3.5x 1000: 3x 0111: 3.5x 0110: 3x 0101: 2.5x 0100: 2x 0011: 2.5x 0010: 2x 0001: 1.5x 0000: 1x	0x0
17	RW	otg0_PLLBTUNE	PLL Bandwidth adjustment	0x1
16	RW	otg0_VREGBYPASS	VReg18 Bypass Control This signal controls the VReg18 bypass mode. 1: The VReg18 is bypassed. 1.8V power supply needs to be provided on VDDH<#>. 0: The VReg18 is not bypassed. A 3.3V supply needs to be provided on VDDH<#> and the internal 1.8V supply is generated by the VReg18	0x1
15	RW	otg0_RETENABLEN	Retention Mode Enable	0x1
14	RW	otg0_RESREQIN	Reserved	0x0

Bits	Access	Name	Description	Reset
13	RW	otg0_VBUSVLDEXTSEL0	External VBUS Valid Select This signal selects either the VBUSVLDEXT0 input or the internal Session Valid comparator to detect whether the VBUS signal on the USB cable is valid and to assert the DP0 pull-up resistor. The activation of the DP0 pull-up resistor also depends on the state of XCVRSEL0[1:0], OPMODE0[1:0], TERMSEL0, DPPULLDOWN0, and DMPULLDOWN0. 1: The VBUSVLDEXT0 input is used to assert the DP0 pull-up resistor. 0: The internal Session Valid comparator is used to assert the DP0 pull-up resistor.	0x0
12	RW	otg0_OTGDISABLE0	OTG Block Disable This signal powers down the VBUS Valid comparator, but not the Session Valid comparator, nor the ID detection circuitry. To save power, if the application does not use the OTG function, this input can be set high. 1: The OTG block is powered down. 0: The OTG block is powered up.	0x0
11	RW	otg0_DRVVBUS0	Drive VBUS This controller signal controls the VBUS Valid comparator. This signal drives 5 V on VBUS through an external charge pump. When OTGDISABLE0 is set to 1'b0 and DRVVBUS0 is asserted, the Bandgap circuitry and VBUS Valid comparator are powered, even in Suspend or Sleep mode. 1: The VBUS Valid comparator is enabled. 0: The VBUS Valid comparator is disabled.	0x1
10	RW	otg0_AUTORSMENB0	Autoresume Mode Enable	0x0

Bits	Access	Name	Description	Reset
9	RW	otg0_HSXCVR_EXTCTL0	HS Transceiver Asynchronous Control This signal asynchronously selects the HS transceiver Tx/Rx path. This signal can be used with the bypass signals in "UART/ Autoresume Signals" on page 50 to perform proprietary tests on the HS Tx / Rx paths. The PHY must be in an HS state with XCVRSEL0[1:0] = 2'b00 and TERMSEL0 = 1'b0. 1: HS Tx/Rx path is selected 0: FS Tx/Rx path is selected	0x0
8	RW	otg0_FS_DATA_EXT0	USB 1.1 Transmit Data This controller signal sets the USB to either a J or K state. This signal is valid only if FSXCVR_OWNER0 is set to 1'b1, TXENABLEN0 is set to 1'b0, and FSSE0_EXT0 is set to 1'b0. 1: The D+ and D- lines are driven to a Differential 1. 0: The D+ and D- lines are driven to a Differential 0. If FSDATA_EXT0 is not used, set it to 1'b0.	0x0
7	RW	otg0_FS_SE0_EXT0	USB 1.1 SE0 Generation This controller signal sets the USB to an SE0 state. This signal is valid only if FSXCVR_OWNER0 is set to 1'b1 and TXENABLEN0 is set to 1'b0. 1: The D+ and D- lines are driven to an SE0 state. 0: The D+ and D- line states are determined by FSDATA_EXT0. If FSSE0_EXT0 is not used, set it to 1'b0.	0x0
6	RW	otg0_TX_ENABLEN0	USB 1.1 Data Enable This controller signal enables the FSDATA_EXT0 and FSSE0_EXT0 inputs. TXENABLEN0 is valid only when the FSXCVR_OWNER0 signal is set to 1'b1. 1: FSDATA_EXT0 and FSSE0_EXT0 are disabled. 0: FSDATA_EXT0 and FSSE0_EXT0 are enabled. If TXENABLEN0 is not used, set it to 1'b1.	0x0

Bits	Access	Name	Description	Reset
5:3	RW	otg0_COMPDISTUNE0	Disconnect Threshold Adjustment This bus adjusts the voltage level for the threshold used to detect a disconnect event at the host. 111: +12% 110: +10% 101: +7% 100: +3% 011: 0, Design default 010: -4% 001: -6% 000: -9%	0x3
2:0	RW	otg0_SQRXTUNE0	Squelch Threshold Adjustment This bus adjusts the voltage level for the threshold used to detect valid high-speed data. 111: -20% 110: -15% 101: -10% 100: -5% 011: 0, Design default 010: +5% 001: +10% 000: +15%	0x0

USB0_CTL1

USB Ctrl reg in SoC.

Offset:0xb0 Total Reset Value:0x0067_F1A3

Bits	Access	Name	Description	Reset
23:22	RW	otg0_VDATREFTUNE0	Data Detect Voltage Adjustment This bus adjusts the threshold voltage level (Vdat_ref) used to detect data during charger type detection. 11: -20% 10: -10% 01: 0, Design default 00: +10%	0x1

Bits	Access	Name	Description	Reset
21:19	RW	otg0_OTGTUNE0	VBUS Valid Threshold Adjustment This bus adjusts the voltage level for the VBUS Valid threshold. 111: +12% 110: +9% 101: +6% 100: +3% 011: 0, Design default 010: -3% 001: -6% 000: -9%	0x4
18:17	RW	otg0_TXHSXVTUNE0	Transmitter High-Speed Crossover Adjustment This bus adjusts the voltage at which the DP0 and DM0 signals cross while transmitting in HS mode. 11: 0, Default setting 10: +15 mV 01: -15 mV 00: Reserved	0x3
16:13	RW	otg0_TXFSLSTUNE0	FS/LS Source Impedance Adjustment This bus adjusts the low- and full-speed single-ended source impedance while driving high. The following adjustment values are based on nominal process, voltage, and temperature. 1111: -10.7% 0111: -5.6% 0011: 0, Design default 0001: +6.5% 0000: +14%	0xf

Bits	Access	Name	Description	Reset
12:9	RW	otg0_TXVREFTUNE0	HS DC Voltage Level Adjustment This bus adjusts the high-speed DC level voltage. 1111: +24% 1110: +22% 1101: +20% 1100: +18% 1011: +16% 1010: +14% 1001: +12% 1000: +10% 0111: +8% 0110: +6% 0101: +4% 0100: +2% 0011: 0, Design default 0010: -2% 0001: -4% 0000: -6%	0x8
8:7	RW	otg0_TXRISETUNE0	HS Transmitter Rise/Fall Time Adjustment This bus adjusts the rise/fall times of the high-speed waveform. 11: -8.1% 10: -7.2% 01: 0, Design default 00: +5.4%	0x3
6:5	RW	otg0_TXRESTUNE0	USB Source Impedance Adjustment In some applications, there can be significant series resistance on the D+ and D- paths between the transceiver and cable. This bus adjusts the driver source impedance to compensate for added series resistance on the USB. Note: Any setting other than the default can result in source impedance variation across process, voltage, and temperature conditions that does not meet USB 2.0 specification limits. 11: Source impedance is decreased by approximately 5.6 ohm 10: Source impedance is decreased by approximately 3.6 ohm 01: 0, Design default 00: Source impedance is increased by approximately 1.3 ohm	0x1

Bits	Access	Name	Description	Reset
4:3	RW	otg0_TXPREEMPAMPTUNE0	HS Transmitter Pre-Emphasis Current Control This signal controls the amount of current sourced to DP0 and DM0 after a J-to-K or K-to-J transition. The HS Transmitter pre- emphasis current is defined in terms of unit amounts. One unit amount is approximately 600 uA and is defined as 1X pre-emphasis current. 11: HS Transmitter pre-emphasis circuit sources 3x pre-emphasis current. 10: HS Transmitter pre-emphasis circuit sources 2x pre-emphasis current. 01: HS Transmitter pre-emphasis circuit sources 1x pre-emphasis current. 00 (design default): HS Transmitter pre-emphasis is disabled. If these signals are not used, set them to the design default.	0x0
2	RW	otg0_TXPREEMPULSETUNE0	HS Transmitter Pre-Emphasis Duration Control This signal controls the duration for which the HS pre-emphasis current is sourced onto DP0 or DM0. The HS Transmitter pre-emphasis duration is defined in terms of unit amounts. One unit of pre- emphasis duration is approximately 580 ps and is defined as 1X pre- emphasis duration. This signal is valid only if either TXPREEMPAMPTUNE0[1] or TXPREEMPAMPTUNE0[0] is set to 1'b1. 1: 1x, short pre-emphasis current duration 0 (design default): 2x, long pre-emphasis current duration	0x0
1:0	RW	otg0_CHRGSRCPUENB0	Reserved	0x3

USB1_CTL0

USB Ctrl reg in SoC.

Offset:0xb4 Total Reset Value:0x0003_8818

Bits	Access	Name	Description	Reset
23:22	RW	otg1_PLLITUNE	Same as otg0	0x0

Bits	Access	Name	Description	Reset
21:18	RW	otg1_PLLPTUNE	Same as otg0	0x0
17	RW	otg1_PLLBTUNE	Same as otg0	0x1
16	RW	otg1_VREGBYPASS	Same as otg0	0x1
15	RW	otg1_RETENABLEN	Same as otg0	0x1
14	RW	otg1_RESREQIN	Same as otg0	0x0
13	RW	otg1_VBUSVLDEXTSEL0	Same as otg0	0x0
12	RW	otg1_OTGDISABLE0	Same as otg0	0x0
11	RW	otg1_DRVVBUS0	Same as otg0	0x1
10	RW	otg1_AUTORSMENB0	Same as otg0	0x0
9	RW	otg1_HSXCVREXTCTL0	Same as otg0	0x0
8	RW	otg1_FSDATAEXT0	Same as otg0	0x0
7	RW	otg1_FSSE0EXT0	Same as otg0	0x0
6	RW	otg1_TXENABLEN0	Same as otg0	0x0
5:3	RW	otg1_COMPDISTUNE0	Same as otg0	0x3
2:0	RW	otg1_SQRXTUNE0	Same as otg0	0x0

USB1_CTL1

USB Ctrl reg in SoC.

Offset:0xb8 Total Reset Value:0x0067_F1A3

Bits	Access	Name	Description	Reset
23:22	RW	otg1_VDATREFTUNE0	Same as otg0	0x1
21:19	RW	otg1_OTGTUNE0	Same as otg0	0x4
18:17	RW	otg1_TXHSXVTUNE0	Same as otg0	0x3
16:13	RW	otg1_TXFSLSTUNE0	Same as otg0	0xf
12:9	RW	otg1_TXVREFTUNE0	Same as otg0	0x8
8:7	RW	otg1_TXRISETUNE0	Same as otg0	0x3
6:5	RW	otg1_TXRESTUNE0	Same as otg0	0x1
4:3	RW	otg1_TXPREEMPAMPTUNE0	Same as otg0	0x0
2	RW	otg1_TXPREEMPPULSETUNE0	Same as otg0	0x0
1:0	RW	otg1_CHRGSRCPUENB0	Same as otg0	0x3

12.7 LSADC

12.7.1 Overview

The low speed ADC (LSADC) measures analog signals by converting external analog signals into values in proportion, which can be applied in battery level detection and key detection. The chip provides one LSADC and 6

independent channels.

12.7.2 Feature

The LSADC has the following features:

- bus interface: using APB 3.0 interface (including Pready interface is after death as 1)
- data bits wide: 32 bit, through the APB bus configuration
- configuration clock: configuration clock for PCLK clock (maximum 200 MHZ)
- sampling clock: LSADC sampling clock is 16 MHZ, support 1 ~ 32 points frequency
- data conversion time: an analog-to-digital conversion need 16 sampling clock cycle
- conversion rate: in terms of 16M sampling clock, the highest 1M per second, minimum 50K per second; Using 32 frequency division of 16M clock as the sampling clock (500K), the maximum and the minimum are 31.25K and 1.5625K per second
- input channel number: support 6 channel switch input
- resolution: If the full voltage is 1.8V, the theoretical resolution is $1.8/2^{12}=0.44\text{mV}$, while the actual resolution is $1.8/2^{10}=1.76\text{mV}$. For example, when the full voltage is 1.8V, the effective bit of the 12bit digital signal is 10bit. If the analog input is 900mV, the theoretical conversion value is 2048, but the actual conversion value may be any of the following values: 2046, 2047, 2048, 2049, or 2050
- output modes: support four modes. 00 indicates a single sampling of a channel (channel configurable), and the data is stored in a register. 01 refers to high-speed continuous sampling for a channel (the channel can be configured) (the sampling rate can be up to 1M times per second, and the sampling rate can be reduced by reducing the sampling clock). The data is stored in registers or DMA (through register selection). When stored in registers, the data of the last sample will overwrite the previous one, and all data will be stored in DMA. 10 refers to the automatic hardware switching of two channels (channels can be configured), low speed continuous sampling, data DMA, all the sampled data will be stored; 11 is for a certain three channels (channel configurable) hardware automatic switching, low speed continuous sampling, data DMA, all the sampled data will be stored
- support interrupt: 1) support DMA_ERROR interrupt, sending, a total interrupt by register to distinguish the specific is that all the way to the error of the DMA interrupt; 2) In mode 01 mode, one threshold interrupt uplink is supported, regardless of channel. The threshold supports the combination design of upper limit and lower limit (high-pass, low-pass, band-pass and band-stop)

12.7.3 Block Diagram

Figure 12-7-1 shows the following functional groupings of the main interfaces to the LSADC block:

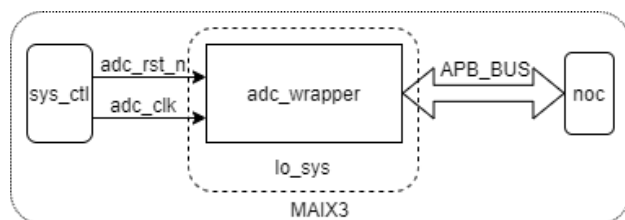


Figure 12-7-1 LSADC Block Diagram

The LSADC in the MAIX3 is the CANAAN IP team's 6-channel, 12bit resolution (10bit significant), one-way LSADC converter that delivers up to 1M per second of continuous single-channel LSADC.

12.7.4 Function Description

The internal structure of LSADC is shown in Figure 12-7-2:

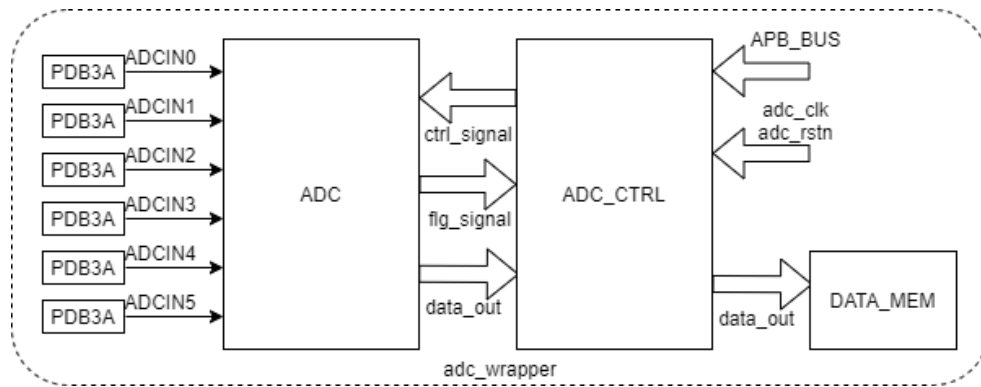


Figure 12-7-2 LSADC Architecture Diagram

LSADC mainly realizes the function of analog-to-digital conversion. It supports six analog inputs to switch inputs and one digital output. It is connected to the noc module via the APB3.0 bus and its sampling clock and reset are provided by the sys_ctl module. LSADC control signal and conversion data output are transmitted through the APB bus.

In MAIX3, we support four output modes of LSADC. Because MAIX3 only has one LSADC converter, and the product has the demand of multi-channel low-speed conversion at the same time, we wrap a layer of circuit around LSADC to realize the time division multiplexing of LSADC conversion and meet the demand of the product.

The four output modes of LSADC are as follows:

- 00 is a single sampling of a channel (channel configurable), and the data is stored in the register;
- 01 refers to high-speed continuous sampling for a channel (the channel can be configured) (the sampling rate can be up to 1M times per second, and the sampling rate can be reduced by reducing the sampling clock). The data is stored in registers or DMA (through register selection). When stored in registers, the data of the last sample will overwrite the previous one, and all data will be stored in DMA.
- 10 refers to the automatic hardware switching of two channels (channels can be configured), low speed continuous sampling, data DMA, all the sampled data will be stored;
- 11 refers to the automatic hardware switching of three channels (channels can be configured), low-speed continuous sampling, data DMA, and all the sampled data will be stored.

In the continuous sampling and continuous output mode, the converted data is temporarily stored in the DATA_MEM module inside adc_wrapper, and 8 data are stored for burst transmission. In order to make LSADC usage as flexible as possible, three 16x12bit FIFOs are built into the DATA_MEM module to realize the temporary storage of the output data of the maximum three-way conversion. At the same time, there are six 12bit data registers to store the data of the single converted digital signal.

As shown in the Figure 12-8-3, it is the conversion sequence diagram when LSADC continuously switches the input port:

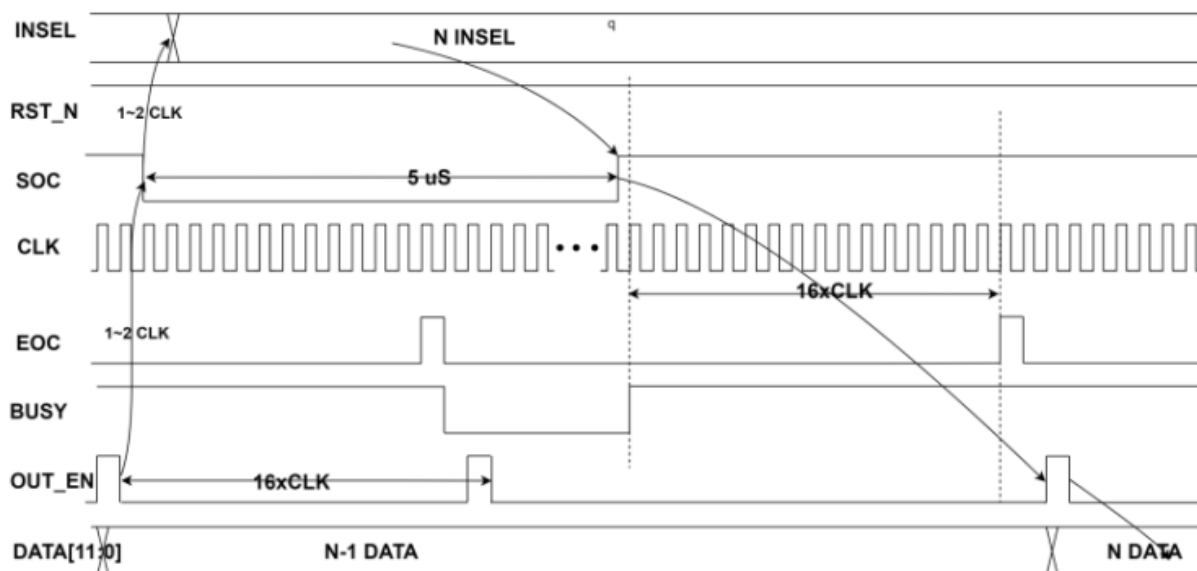


Figure 12-7-3 LSADC Continuously Switches Input Conversion Sequence diagram

When the input port is continuously switched (the switch is automatically controlled by the hardware circuit), $21T_{clk} + 5\mu s = 18.9375\mu s$ ($T_{clk} = 1/16\text{MHz}$) is required for a data conversion. In the slowest case (continuous switching of three inputs), $18.9375\mu s \times 3$ is required for a set of data conversion, that is, the conversion rate is about 50K times per second.

As shown in the Figure 12-7-4, it is the conversion sequence diagram of LSADC single-channel input:

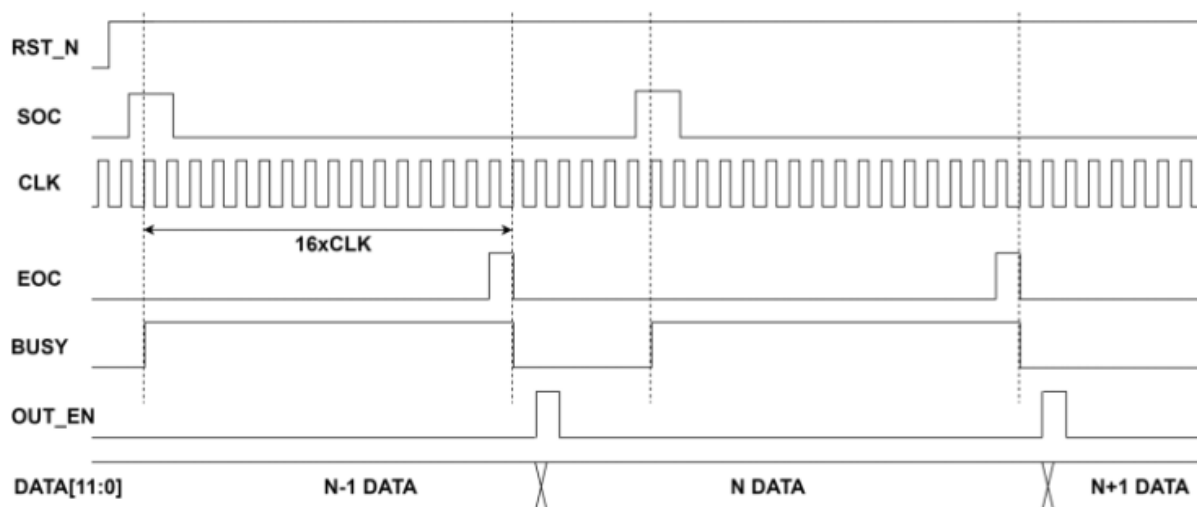


Figure 12-7-4 LSADC Single-channel Input Conversion Sequence Diagram

When a single channel is used for single conversion, it takes $16 \times T_{clk} = 1\mu s$ ($T_{clk} = 1/16\text{MHz}$) to convert one data, so the conversion rate is about 1M times per second.

12.7.5 Register Description

12.7.5.1 Register Summary

The LSADC base addresses is 0x9140_D000.

Table 12-7-1 Summary of LSADC registers

Offset Address	Register	Description	Page

Offset Address	Register	Description	Page
0x00	TRIM_REG	LSADC initializes the self-calibrating control register	12-7-2
0x04	CFG_REG	LSADC Data conversion control register	12-7-3
0x08	MODE_REG	LSADC output mode selection control register	12-7-4
0x0C	THSD_REG	LSADC threshold interrupt control register	12-7-5
0x10	DMA_INTR_REG	LSADC's DMA error interrupt register	12-7-6
0x14+N*4	DATA_REG_N (N=0,1,2,3,4,5)	Input channel N Digital signal output	12-7-7
0x2C+(N-1)*4	DATA_DMA_N (N=1,2,3)	Continuous sampling channel N digital signal output	12-7-8

12.7.5.2 Register Description

12.7.5.2.1 TRIM_REG

- Name: LSADC initializes the self-calibrating control register
- Description: LSADC initializes the self-calibrating control register
- Reset Value: 0x8100
- Offset Address: 0x00

Table [12-7-2](#) Fields for Register: TRIM_REG

Bits	Access	Name	Description	Reset Value
31:25	R	RSV	Reserved	0x0
24	R	OSCAL_DONE	SAR LSADC Offset Calibration End flg.	0x0
23:21	R	RSV	Reserved	0x0
20	R/W	OSCAL_EN	SAR LSADC Offset Calibration Enable. 1: Short VIP and VIN Together Internal for Offset Calibration.	0x0
19:16	R	RSV	Reserved	0x0
15:12	R/W	BGTRIM	Bandgap Trim Code for reference trim in BG. Default Value 4'b1000.	0x8
11:10	R	RSV	Reserved	0x0
9:8	R/W	VRSEL	SAR Inter-Reference Value Selection. 00: 0.85V; 01: 0.9V; 10: 0.95V; 11: 1.0V	0x1
7:5	R	RSV	Reserved	0x0
4	R/W	INTREFSEL	SAR Inter-Reference Selection. 0: External-VREFP/VREFN Select; 1: Inner-Reference Select.	0x0
3:1	R	RSV	Reserved	0x0

Bits	Access	Name	Description	Reset Value
0	R/W	ENADC	SAR Analog Part Power Down Control. 1: Power On; 0: Power Down.	0x0

12.7.5.2.2 CFG_REG

- Name: LSADC Data conversion control register
- Description: LSADC Data conversion control register
- Reset Value: 0x00
- Offset Address: 0x04

Table 12-7-3 Fields for Register: CFG_REG

Bits	Access	Name	Description	Reset Value
31:17	R	RSV	Reserved	0x0
16	R	OUTEN	SAR DATA Output Effective. 0: Not Effective; 1: Effective.	0x0
15:13	R	RSV	Reserved	0x0
12	R	EOC	SAR End of Conversion	0x0
11:9	R	RSV	Reserved	0x0
8	R	BUSY	SAR is In Conversion Mode. 0: SAR is Not Busy; 1: SAR is Busy.	0x0
7:5	R	RSV	Reserved	0x0
4	R/W	SOC	SAR Start of Conversion	0x0
3	R	RSV	Reserved	0x0
2:0	R/W	INSEL	SAR Input Selection. 000: ADCIN0 Select; 001: ADCIN1 Select; 010: ADCIN2 Select; 011: ADCIN3 Select; 100: ADCIN4 Select; 101: ADCIN5 Select;	0x0

12.7.5.2.3 MODE_REG

- Name: LSADC output mode selection control register
- Description: LSADC output mode selection control register
- Reset Value: 0x00
- Offset Address: 0x08

Table 12-7-4 Fields for Register: MODE_REG

Bits	Access	Name	Description	Reset Value
31:28	R	RSV	Reserved	0x0
27	R/W	DMA_4_CLR	1: clears data stored in DMA3	0x0
26:24	R/W	DMA_3_INSEL	Input channel selection for storing data in DMA3: 000: ADCIN0 Select; 001: ADCIN1 Select; 010: ADCIN2 Select; 011: ADCIN3 Select; 100: ADCIN4 Select; 101: ADCIN5 Select;	0x0
23:20	R	RSV	Reserved	0x0
19	R/W	DMA_2_CLR	1: clears data stored in DMA2	0x0
18:16	R/W	DMA_2_INSEL	Input channel selection for storing data in DMA2: 000: ADCIN0 Select; 001: ADCIN1 Select; 010: ADCIN2 Select; 011: ADCIN3 Select; 100: ADCIN4 Select; 101: ADCIN5 Select;	0x0
15:13	R	RSV	Reserved	0x0
12	R/W	DMA_1_CLR	1: clears data stored in DMA1	0x0
11:9	R/W	DMA_1_INSEL	Input channel selection for storing data in DMA1: 000: ADCIN0 Select; 001: ADCIN1 Select; 010: ADCIN2 Select; 011: ADCIN3 Select; 100: ADCIN4Select; 101: ADCIN5 Select;	0x0
8	R/W	DMA_1_EN	This bit is valid when MODE_SEL =01 1: The converted data is stored in DMA 0: The converted data is stored in the register.	0x0
7:5	R	RSV	Reserved	0x0
4	R/W	DMA_PAUSE	1: Pause continuous sampling, switch to single sampling, and continue the previous continuous sampling after completing one sampling.	0x0
3:2	R	RSV	Reserved	0x0

Bits	Access	Name	Description	Reset Value
1:0	R/W	MODE_SEL	Output mode selection : 00: single input, single sample once, save to register; 01: single input, high speed continuous sampling, save to DMA1 or the corresponding channel register 10: two input automatic switching, low speed continuous sampling, save to DMA 11: three input automatic switching, low speed continuous sampling, save to DMA	0x0

12.7.5.2.4 THSD_REG

- Name: LSADC threshold interrupt control register
- Description: LSADC threshold interrupt control register
- Reset Value: 0x00
- Offset Address: 0x0C

Table 12-7-5 Fields for Register: THSD_REG

Bits	Access	Name	Description	Reset Value
31:28	R	RSV	Reserved	0x0
27:16	R/W	THRESHOLD_H	Upper comparison value of a threshold	0x0
15:4	R/W	THRESHOLD_L	Lower limit comparison value of a threshold	0x0
3:2	R	RSV	Reserved	0x0
1:0	R/W	THRESHOLD_SEL	Threshold interrupt mode selection: 00: high pass mode; 01: band pass mode; 10: band resistance mode; 11: low pass mode;	0x0

12.7.5.2.5 DMA_INTR_REG

- Name: LSADC's DMA error interrupt register
- Description: LSADC's DMA error interrupt register
- Reset Value: 0x00
- Offset Address: 0x10

Table 12-7-6 Fields for Register: DMA_INTR_REG

Bits	Access	Name	Description	Reset Value
31:3	R	RSV	Reserved	0x0
2	R	DMA_3_INTR	Error interrupt for DMA3	0x0
1	R	DMA_2_INTR	Error interrupt for DMA2	0x0
0	R	DMA_1_INTR	Error interrupt for DMA1	0x0

12.7.5.2.6 DATA_REG_N (N=0,1,2,3,4,5)

- Name: Input channel N Digital signal output
- Description: Input channel N Digital signal output
- Reset Value: 0x00
- Offset Address: 0x14+N*4

Table 12-7-7 Fields for Register: DATA_REG_N (N=0,1,2,3,4,5)

Bits	Access	Name	Description	Reset Value
31:12	R	RSV	Reserved	0x0
11:0	R	DATA_N	N Input channel conversion data output.	0x0

12.7.5.2.7 DATA_DMA_N (N=1,2,3)

- Name: Continuous sampling channel N digital signal output
- Description: Continuous sampling channel N digital signal output
- Reset Value: 0x00
- Offset Address: 0x2C+(N-1)*4

Table 12-7-8 Fields for Register: DATA_DMA_N (N=1,2,3)

Bits	Access	Name	Description	Reset Value
31:12	R	RSV	Reserved	0x0
11:0	R	DATA_N	Continuous conversion data output of sampling channel N (8 burst transfers).	0x0

12.7.6 Interface Description

Table 12-7-9 LSADC Interface Description

Module pin	Direction	PAD name	IOMUX
adc_ADCIN0	I	ANA_adc_ADCIN0	Analog pin
adc_ADCIN1	I	ANA_adc_ADCIN1	Analog pin
adc_ADCIN2	I	ANA_adc_ADCIN2	Analog pin
adc_ADCIN3	I	ANA_adc_ADCIN3	Analog pin
adc_ADCIN4	I	ANA_adc_ADCIN4	Analog pin
adc_ADCIN5	I	ANA_adc_ADCIN5	Analog pin

12.7.7 Application Notes

The following Figure 12-7-5 shows the timing diagram of LSADC initialization and initial calibration:

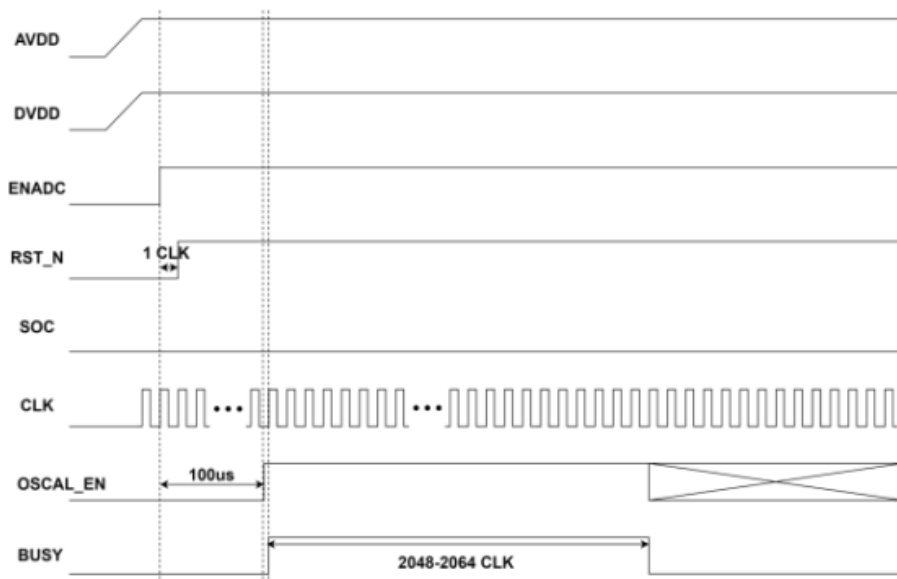


Figure 12-7-5 LSADC Initializes The Sequence Diagram

When the analog and digital VDD are powered:

1. Enable LSADC first (ENADC set to 1);
2. Wait for at least 1 LSADC cycle and then dereset (LSADC is reset by default);
3. Wait at least 100us to enable automatic calibration (OSCAL_EN set to 1);
4. Finally, when the calibration signal is detected (OSCAL_DONE is 1), the automatic calibration will be disabled (OSCAL_EN is set to 0, if not, subsequent data conversion will be affected).

The automatic calibration will last about 2048~2064 LSADC cycles.

12.8 PWM

12.8.1 Overview

There are two independent 3-bit Pulse Width Modulation (PWM)generators, each can generate multiple types of waveforms.

12.8.2 Feature

The PWM has the following features:

- 6 channel
- interface of APB 3.0, width 32 bit data; Slave Error response is not supported
- support arbitrary duty ratio
- support programmable output waveform (support two adjacent channel output waveform and even)
- support produce periodic pulse signal of pulse signal and one-time
- precise interruption generator can be used as a cycle
- support output burr elimination

12.8.3 Block Diagram

Figure 12-8-1 shows the following functional groupings of the main interfaces to the PWM block:

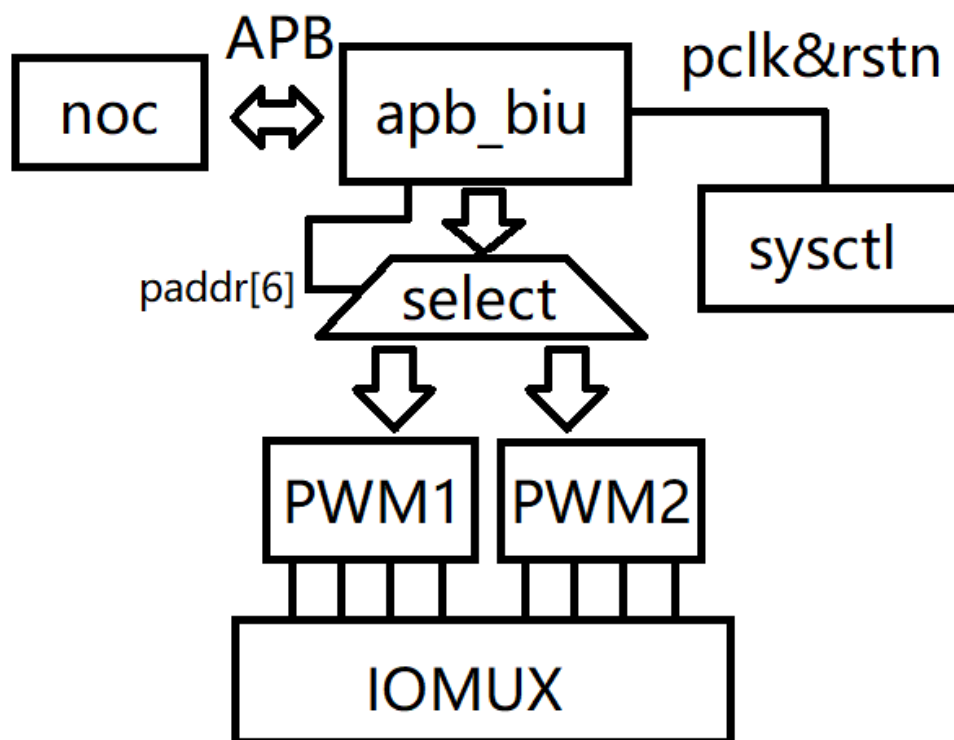


Figure 12-8-1 PWM Block Diagram

The PWM consists of 4 channels: 0,1,2,3, among which 4 channels are bound (the number of clock cycles in which the output pulse repeats is the same), 0 channel outputs 1 narrow pulse of clock cycle width, and the output pulse width of 1, 2, 3 channels is adjustable (adjustable duty ratio). We use channels 1,2,3 channels.

12.8.4 Function Description

The internal structure of PWM is shown in Figure 12-8-2, and its characteristics are as follows:

1. Each PWM supports four programmable comparators (called pwmcmp0~pwmcmp3), each of which can generate a corresponding PWM output (signal name pwmcmp_X_gpio) and interrupt (signal name pwmcmp_X_ip), as shown in Figure 12-8-2.
2. Each PWM output can generate left-aligned or right-aligned pulse signals.
3. Each PWM output can generate a centered pulse signal.
4. Each PWM output can generate any shape pulse signal.
5. Each PWM output can generate periodic pulse signal or one-time pulse signal.
6. Each PWM output can be used as a period accurate interrupt generator.

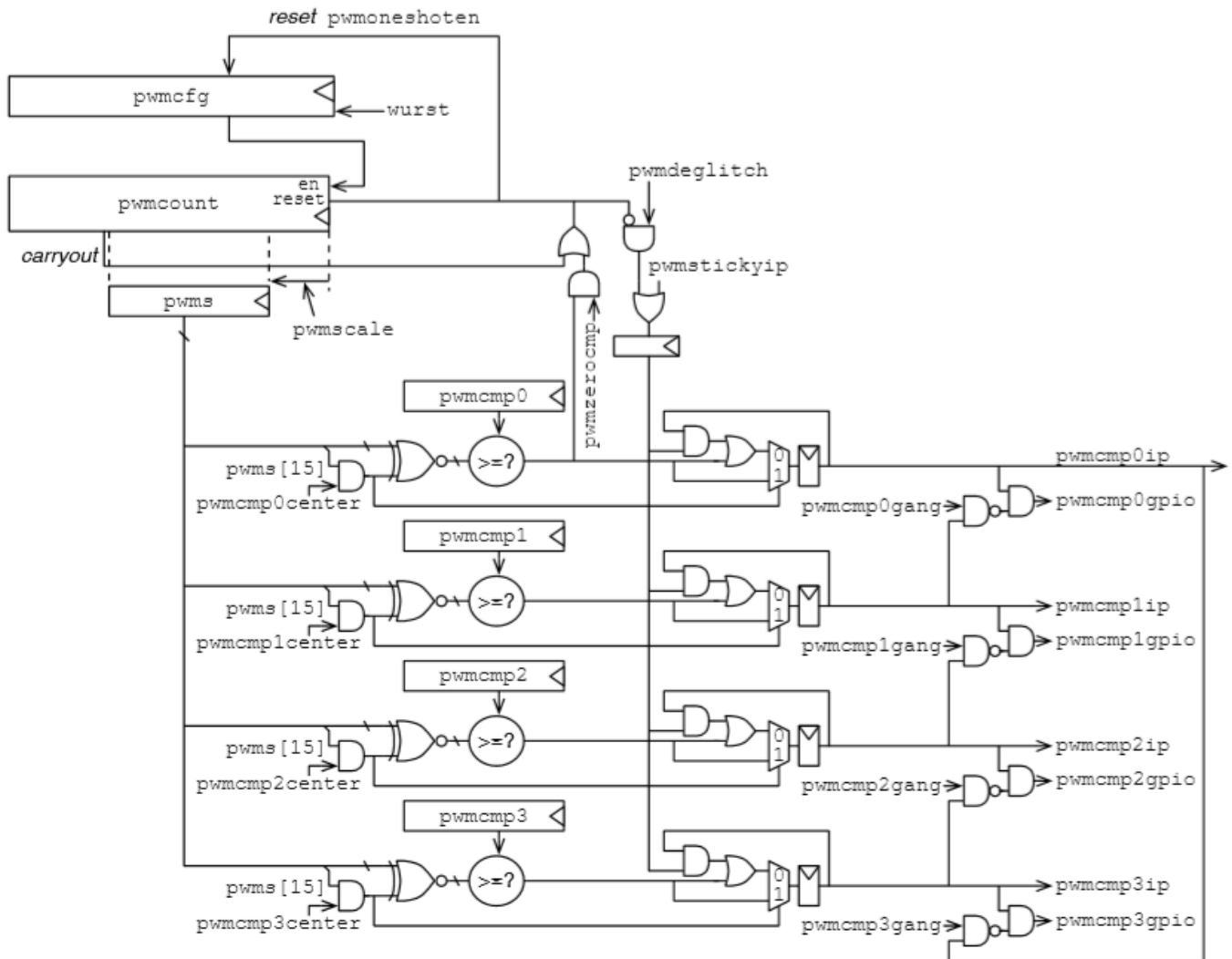


Figure 12-8-2 PWM Architecture Diagram

12.8.5 Register Description

12.8.5.1 Register Summary

The PWM base addresses is 0x9140_A000.

Table 12-8-1 Summary of PWM registers

Offset Address	Register	Description	Page
0x00	PWMCFG	PWM configuration register	12-8-2
0x08	PWMCOUNT	PWM counter count value register	12-8-3
0x10	PWMS	PWM counter is relatively straight register	12-8-4
0x20+N*4	PWMCMPN (N=0,1,2,3)	PWM comparator register N	12-8-5

12.8.5.2 Register Description

12.8.5.2.1 PWMCFG

- Name: PWM configuration register
- Description: PWM configuration register
- Reset Value: 0x0

- Offset Address: 0x00

Table 12-8-2 Fields for Register: PWMCFG

Bits	Access	Name	Description	Reset Value
31	R/W	Pwmcmp3ip	The interrupt pending bits pwmcmp3ip The pwmcmp3ip bits are only allowed to change at the start of the next PWM cycle. The interrupt pending bits pwmcmp3ip can be cleared down using writes to the pwmcfg register. The PWM peripheral can also be used as a regular timer with no counter reset (pwmzerocmp=0), where the comparators are now used to provide timer interrupts.	0x0
30	R/W	Pwmcmp2ip	The interrupt pending bits pwmcmp2ip The pwmcmp2ip bits are only allowed to change at the start of the next PWM cycle. The interrupt pending bits pwmcmp2ip can be cleared down using writes to the pwmcfg register. The PWM peripheral can also be used as a regular timer with no counter reset (pwmzerocmp=0), where the comparators are now used to provide timer interrupts.	0x0
29	R/W	Pwmcmp1ip	The interrupt pending bits pwmcmp1ip The pwmcmp1ip bits are only allowed to change at the start of the next PWM cycle. The interrupt pending bits pwmcmp1ip can be cleared down using writes to the pwmcfg register. The PWM peripheral can also be used as a regular timer with no counter reset (pwmzerocmp=0), where the comparators are now used to provide timer interrupts.	0x0
28	R/W	Pwmcmp0ip	The interrupt pending bits pwmcmp0ip The pwmcmp0ip bits are only allowed to change at the start of the next PWM cycle. Note the pwmcmp0ip bit will only be high for one cycle when pwmdeglitch and pwmzerocmp are set where pwmcmp0 is used to define the PWM cycle, but can be used as a regular PWM edge otherwise. The interrupt pending bits pwmcmp0ip can be cleared down using writes to the pwmcfg register. The PWM peripheral can also be used as a regular timer with no counter reset (pwmzerocmp=0), where the comparators are now used to provide timer interrupts.	0x0
27	R/W	Pwmcmp3gang	A comparator can be ganged together with its next-highest-numbered neighbor to generate arbitrary PWM pulses. When the pwmcmp3gang bit is set, comparator 3 fires and raises its pwm3gpio signal. When comparator 3 + 1 (or pwmcmp0 for pwmcmp3) fires, the pwm3gpio output is reset to zero.	0x0
26	R/W	Pwmcmp2gang	A comparator can be ganged together with its next-highest-numbered neighbor to generate arbitrary PWM pulses. When the pwmcmp2gang bit is set, comparator 2 fires and raises its pwm2gpio signal. When comparator 2 + 1 (or pwmcmp0 for pwmcmp3) fires, the pwm2gpio output is reset to zero.	0x0

Bits	Access	Name	Description	Reset Value
25	R/W	Pwmcmp1gang	A comparator can be ganged together with its next-highest-numbered neighbor to generate arbitrary PWM pulses. When the pwmcmp1gang bit is set, comparator 1 fires and raises its pwm1gpio signal. When comparator 1 + 1 (or pwmcmp0 for pwmcmp3) fires, the pwm1gpio output is reset to zero.	0x0
24	R/W	Pwmcmp0gang	A comparator can be ganged together with its next-highest-numbered neighbor to generate arbitrary PWM pulses. When the pwmcmp0gang bit is set, comparator 0 fires and raises its pwm0gpio signal. When comparator 0 + 1 (or pwmcmp0 for pwmcmp3) fires, the pwm0gpio output is reset to zero.	0x0
23:20	R	RSV	Reserved	0x0
19	R/W	Pwmcmp3center	A per-comparator pwmcmp3center bit in pwmcfg allows a single PWM comparator to generate a center-aligned symmetric duty-cycle	0x0
18	R/W	Pwmcmp2center	A per-comparator pwmcmp2center bit in pwmcfg allows a single PWM comparator to generate a center-aligned symmetric duty-cycle	0x0
17	R/W	Pwmcmp1center	A per-comparator pwmcmp1center bit in pwmcfg allows a single PWM comparator to generate a center-aligned symmetric duty-cycle	0x0
16	R/W	Pwmcmp0center	A per-comparator pwmcmp0center bit in pwmcfg allows a single PWM comparator to generate a center-aligned symmetric duty-cycle	0x0
15:14	R	RSV	Reserved	0x0
13	R/W	pwmenoneshot	The pwmenbits control the conditions under which the PWM counter pwmcount is incremented. The counter increments by one each cycle only if any of the enabled conditions are true. When pwmenoneshot is set, the counter can increment but pwmenoneshot is reset to zero once the counter resets, disabling further counting (unless pwmenalways is set). The pwmenoneshot bit provides a way for software to generate a single PWM cycle then stop. Software can set the pwmenoneshot again at any time to replay the one-shot waveform. The pwmen bits are reset at wakeup reset, which disables the PWM counter and saves power.	0x0
12	R/W	pwmenalways	The pwmen* bits control the conditions under which the PWM counter pwmcount is incremented. The counter increments by one each cycle only if any of the enabled conditions are true. If the pwmenalways bit is set, the PWM counter increments continuously.	0x0
11	R	RSV	Reserved	0x0

Bits	Access	Name	Description	Reset Value
10	R/W	pwmdeglitch	To avoid glitches in the PWM waveforms when changing pwmcmpX register values, the pwmdeglitch bit in pwmcfg can be set to capture any high output of a PWM comparator in a sticky bit(pwmcmpXip for comparator X) and prevent the output falling again within the same PWM cycle. The pwmcmpXip bits are only allowed to change at the start of the next PWM cycle. Note the pwmcmp0ip bit will only be high for one cycle when pwmdeglitch and pwmzerocmp are set where pwmcmp0 is used to define the PWM cycle, but can be used as a regular PWM edge otherwise. If pwmdeglitch is set, but pwmzerocmp is clear, the deglitch circuit is still operational but is now triggered when pwms contains all 1s and will cause a carry out of the high bit of the pwms incrementer just before the counter wraps to zero.	0x0
9	R/W	pwmzerocmp	The pwmzerocmp bit, if set, causes the PWM counter pwmcount to be automatically reset to zero one cycle after the pwms counter value matches the compare value in pwmcmp0. This is normally used to set the period of the PWM cycle. This feature can also be used to implement periodic counter interrupts, where the period is independent of interrupt service time.	0x0
8	R/W	pwmsticky	The pwmsticky bit will disallow the pwmcmpXip registers from clearing if they're already set, and is used to ensure interrupts are seen from the pwmcmpXip bits.	0x0
7:4	R	RSV	Reserved	0x0
3:0	R/W	pwmscale	The 4-bit pwmscale field scales the PWM counter value before feeding it to the PWM comparators. The value in pwmscale is the bit position within the pwmcount register of the start of a cmpwidth-bit pwms field. A value of 0 in pwmscale indicates no scaling, and pwms would then be equal to the low cmpwidth bits of pwmcount. The maximum value of 15 in pwmscale corresponds to dividing the clock rate by 215, so for an input bus clock of 16MHz, the LSB of pwms will increment at 488.3Hz.	0x0

12.8.5.2.2 PWMCOUNT

- Name: PWM counter count value register
- Description: PWM counter count value register
- Reset Value: 0x0
- Offset Address: 0x08

Table 12-8-3 Fields for Register: PWMCOUNT

Bits	Access	Name	Description	Reset Value
31	R	RSV	Reserved	0x0

Bits	Access	Name	Description	Reset Value
30:0	R/W	counter	The PWM unit is based around a counter held in pwmcount. The counter can be read or written over the bus. The pwmcount register is (15 + cmpwidth) bits wide. For example, for cmpwidth of 16 bits, the counter is held in pwmcount[30:0], and bit 31 of pwmcount returns a zero when read.	0x0

12.8.5.2.3 PWMS

- Name: PWM counter is relatively straight register
- Description: PWM counter is relatively straight register
- Reset Value: 0x0
- Offset Address: 0x10

Table 12-8-4 Fields for Register: PWMS

Bits	Access	Name	Description	Reset Value
31:16	R	RSV	Reserved	0x0
15:0	R/W	pwms	The value of pwms is memory-mapped and can be read as a single cmpwidth-bit value over the bus.	0x0

12.8.5.2.4 PWMCPMN (N=0,1,2,3)

- Name: PWM comparator register N
- Description: PWM comparator register N
- Reset Value: 0x0
- Offset Address: 0x20+N*4

Table 12-8-5 Fields for Register: PWMCPMN

Bits	Access	Name	Description	Reset Value
31	R	RSV	Reserved	0x0
30:0	R/W	PWM_CMPN (N=0,1,2,3)	The primary use of the ncmp PWM compare registers is to define the edges of the PWM waveforms within the PWM cycle. Each compare register is a cmpwidth-bit value against which the current pwms value is compared every cycle. The output of each comparator is high whenever the value of pwms is greater than or equal to the corresponding pwmcmpN.	0x0

12.8.6 Interface Description

Table 12-8-6 PWM Interface Description

Module pin	Direction	PAD name	IOMUX

Module pin	Direction	PAD name	IOMUX
pwm_pwm_pins_1_io_pins_pwm_0_o_oval	O	PAD_IO_60;PAD_IO_42;PAD_IO_54	IO_60_sel == 3'd1;IO_42_sel == 3'd2;IO_54_sel == 3'd3
pwm_pwm_pins_1_io_pins_pwm_1_o_oval	O	PAD_IO_61;PAD_IO_43;PAD_IO_55	IO_61_sel == 3'd1;IO_43_sel == 3'd2;IO_55_sel == 3'd3
pwm_pwm_pins_1_io_pins_pwm_2_o_oval	O	PAD_IO_7;PAD_IO_46;PAD_IO_56	IO_7_sel == 3'd1;IO_46_sel == 3'd2;IO_56_sel == 3'd3
pwm_pwm_pins_1_io_pins_pwm_3_o_oval	O	PAD_IO_8;PAD_IO_47;PAD_IO_57	IO_8_sel == 3'd1;IO_47_sel == 3'd2;IO_57_sel == 3'd3
pwm_pwm_pins_1_io_pins_pwm_4_o_oval	O	PAD_IO_9;PAD_IO_52;PAD_IO_58	IO_9_sel == 3'd1;IO_52_sel == 3'd2;IO_58_sel == 3'd3
pwm_pwm_pins_1_io_pins_pwm_5_o_oval	O	PAD_IO_25;PAD_IO_53;PAD_IO_59	IO_25_sel == 3'd1;IO_53_sel == 3'd2;IO_59_sel == 3'd3

12.8.7 Application Notes

PCLK, PWM use APB clock to program registers as well as to generate waveforms. The default frequency is 100MHz. Configure the system control to modify the frequency of PCLK.

12.9 IOMUX

12.9.1 Pinout maps

For detailed description on each Pin, please refer to the [K230 Pinout Spreadsheets](#). If above link is unavailable, please visit the [Canaan Developer Community](#) to search for the latest version.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
A		USB0_DP	USB1_VB_US	USB1_DN	MMC0_CLK	MMC0_ST_N	MMC0_D5	CLK24M_XOUT	TEST_EN	INT2	INT3	CLK32K768_XOUT	VSS	DQ19_DQ83	DQS2N_DQS80N	VSS	DQ20_DQ84	DQ50N_DQ581N	VSS	NC	A
B	ADC_2	USB0_DN	USB1_ID	USB1_DP	MMC0_D2	MMC0_STROBE	MMC0_D6	CLK24M_XIN	RSTN	INT1	OUT0	CLK32K768_XIN	VSS	DQ18_DQ82	DQS2P_DQS80P	DQ21_DQ85	DQ23_DQ87	DQ50P_DQ581P	DQ6_DQ89	VSS	B
C	ADC_4	ADC_0	USB0_VB_US	USB1_TX_RTUNE	MMC0_D1	MMC0_D3	MMC0_D7	GPIO0	GPIO1	INT0	OUT1	INT4	VSS	DQ16_DQ80	DM2_DM180	DQ22_DQ86	DQ0_DQ815	DQ3_DQ812	DQ7_DQ88	NC	C
D	ADC_5	ADC_1	USB0_ID	USB0_TX_RTUNE	MMC0_D0	MMC0_CMD	MMC0_D4	ANA_TS_VTEST	VSS	VSS	VSS	INT5	VSS	DQ17_DQ81	VSS	DQ1_DQ814	DQ2_DQ813	DM0_DM181	NC	NC	D
E	MICPL	MICNL	MIC_BIAS	VSS		VSS	VSS	VSS	AVDD3P3_USB	VDD3P3_SD	VDD1P8	VSS	VSS	VSS	VSS	VSS	DQ5_DQ810	DQ4_DQ811	VSS	NC_CAB0	E
F	MICPR	MICNR	CODEC_VCM	VSS		VSS	AVDD1P8_CODEC	AVDD1P8_ADC	AVDD1P8_USB	VSS	AVSS	VDD0P8_CORE	VDD1P8	AVDD1P8_LDO	AVDD1P8_RTC	VSS	NC_CAB4	NC	NC_CAB5	NC_CLKBN	F
G	HPOUTL	HPOUTR	ADC_3	GPIO48	GPIO49	VSS	AVSS	AVSS	VDD0P8_CORE	VDD0P8_CORE	AVDD0P8_PLL	VDD0P8_CPU	VSS	VSS	VSS	VSS	NC	NC_CAB2	NC_CAB1	NC_CLKBP	G
H	GPIO42	GPIO44	GPIO45	GPIO46	GPIO47	VDDIO_BANK3	VSS	VDD0P8_KPU	VDD0P8_KPU	VDD0P8_KPU	VSS	VDD0P8_CPU	VSS	VDDIO_DR	VSS	VSS	NC_CAB3	NC	NC	VSS	H
J	GPIO41	GPIO40	GPIO39	GPIO38	GPIO14	GPIO43	VSS	VDD0P8_KPU	VDD0P8_KPU	VDD0P8_KPU	VSS	VDD0P8_CPU	VSS	VDDIO_DR	VSS	DDR_RST	NC_CKEB1	NC_CKEB0	NC_CSB1	NC_CSB0	J
K	GPIO15	GPIO16	VSS	GPIO17	GPIO18	VSS	VDDIO_BANK1	VSS	VDD0P8_CORE	VSS	VDD0P8_CORE	VSS	VDD0P8_DDR_CORE	VSS	VAA_DDR	VSS	DDR_VREF	VSS	DDR_ZH	CA0_NC	K
L	GPIO19	GPIO20	GPIO21	GPIO22	GPIO23	GPIO24	VSS	VSS	VDD0P8_CORE	VSS	VDD0P8_CORE	VSS	VDD0P8_DDR_CORE	VDDIO_DR	VSS	CA5_CAA1	CA3_NC	ODT_NC	VSS	CA1_NC	L
M	GPIO25	GPIO54	GPIO52	GPIO51	GPIO50	VSS	VDDIO_BANK4	VSS	VDD0P8_CORE	VSS	VDD0P8_CORE	VSS	VDD0P8_DDR_CORE	VDDIO_DR	VSS	VSS	NC	CA8_CAA4	CA4_CAA0	CA2_IA	M
N	GPIO57	GPIO55	VSS	GPIO53	VSS	AVDD0P8_MIPI	AVDD1P8_MIPI	VSS	VDD0P8_CORE	VSS	VDD0P8_CORE	VSS	VDD0P8_DDR_CORE	VDDIO_DR	VSS	VSS	CKE0_CKEA0	NC	CA6_CAA2	CA7_CAA3	N
P	GPIO58	GPIO59	GPIO56	GPIO61	GPIO60	VSS	VSS	VSS	VDD0P8_CORE	VSS	VDD0P8_CORE	VSS	VDD0P8_CORE	VSS	VSS	DQ14_DQA14	DM1_DM1A1	CKE1_CKEA1	CA9_CAA5	VSS	P
R	MIPI_RX2_CLK_P	MIPI_RX2_CLK_N	VSS			VSS	VSS	AVDD0P8_MIPI	AVDD1P8_MIPI	VSS		VSS	VDDIO_BANK0	VSS	VDDIO_BANK5	VSS	DQ8_DQA8	DQ10_DQA10	CLKP_CLKAP	CLKN_CLKAN	R
T	MIPI_RX2_D4_N	MIPI_RX2_D4_P	VSS				VSS	VSS	VSS	GPIO36	GPIO33	GPIO13	GPIO9	GPIO62	VSS	DQ30_DQA1	DQ15_DQA15	DQ9_DQA9	CS1_CSA1	CS0_CSA0	T
U	MIPI_RX2_D5_P	MIPI_RX2_D5_N	VSS			MIPI_ATB	MIPI_REXT	VSS	GPIO34	GPIO32	GPIO26	GPIO8	GPIO10	GPIO5	GPIO63	DQ31_DQA0	DQ24_DQA7	DQ13_DQA13	VSS	DQ11_DQA11	U
V	MIPI_RX0_D1_N	MIPI_RX0_D1_P	VSS	VSS	VSS	VSS	VSS	VSS	GPIO31	GPIO30	GPIO29	GPIO28	GPIO12	GPIO6	GPIO3	DQ29_DQA2	DM3_DM1A0	DQ26_DQA5	DQS1N_DQS11N	DQS1P_DQS11P	V
W	VSS	MIPI_RX0_CLK_P	MIPI_RX0_D0_N	MIPI_RX1_D2_P	MIPI_RX1_CLK_N	MIPI_RX1_D3_N	MIPI_TX1_D3_N	MIPI_TX1_D2_P	MIPI_TX0_CLK_N	MIPI_TX0_D0_P	MIPI_TX0_D1_N	GPIO35	GPIO27	GPIO7	GPIO2	VSS	DQS3P_DQSA0P	DQ27_DQA4	DQ12_DQA12	VSS	W
Y	NC	MIPI_RX0_CLK_N	MIPI_RX0_D0_P	MIPI_RX1_D2_N	MIPI_RX1_CLK_P	MIPI_RX1_D3_P	MIPI_TX1_D3_P	MIPI_TX1_D2_N	MIPI_TX0_CLK_P	MIPI_TX0_D0_N	MIPI_TX0_D1_P	GPIO37	VSS	GPIO11	GPIO4	DQ28_DQA3	DQS3N_DQSA0N	DQ25_DQA6	VSS	NC	Y
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	

Figure 12-9-1 K230 Pin Map

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16		
A	NC	HPOUTL	MICPR	MIC_BIAS	MICNL	ADC_1	ADC_0	USB0_DN	USB0_ID	MMC0_D1	MMC0_D2	MMC0_D0	MMC0_D3	MMC0_CMD	CLK24M_XIN	NC	A	
B	GPIO44	HPOUTR	MICNR	CODEC_VCM	MICPL	ADC_2	USB0_VBUS	USB0_DP	USB0_TXRTUNE	MMC0_D5	MMC0_D6	MMC0_D7	MMC0_D4	MMC0_CLK	CLK24M_XOUT	VSS	B	
C	GPIO41	GPIO45	GPIO49	VSS	VSS	AVDD1P8_USB	AVDD3P3_USB	VSS	VDD3P3_SD	MMC0_STROBE	MMC0_RST_N	RSTN	TEST_EN	GPIO0	GPIO1	INT0	C	
D	GPIO39	GPIO40	GPIO48	AVDD1P8_CODEC	AVDD1P8_ADC	VSS	VSS	AVDD0P8_PLL	AVSS	VDD1P8	AVDD1P8_LDO	AVDD1P8_RTC	VSS	INT1	CLK32K768_XIN	CLK32K768_XOUT	D	
E	GPIO15	GPIO14	GPIO38	AVSS	AVSS	VDD0P8_CORE	VDD0P8_CORE	VDD1P8	VDD0P8_CORE	VDD0P8_CPU	VSS	VSS	OUT1	INT2	VSS	VSS	E	
F	GPIO17	GPIO16	VSS	VDDIO_BANK3	VDD0P8_KPU	VDD0P8_KPU	VDD0P8_KPU	VSS	VDD0P8_CPU	VSS	VDDIO_DDR	VSS	OUT0	INT4	VDD1P1_DDR_VDD2	VDD1P1_DDR_VDDQ	F	
G	GPIO18	GPIO19	VSS	VSS	VDD0P8_KPU	VDD0P8_KPU	VDD0P8_KPU	VSS	VDD0P8_CPU	VSS	VDDIO_DDR	VSS	VSS	VSS	VSS	VSS	G	
H	GPIO21	GPIO20	VSS	VDDIO_BANK1	VSS	VDD0P8_CORE	VSS	VDD0P8_CORE	VSS	VDD0P8_DDR_CORE	DDR_VREF	VAA_DDR	VSS	VDD1P8_DDR_VDD1	VSS	DDR_ZN	H	
J	GPIO22	GPIO23	GPIO24	VSS	VSS	VDD0P8_CORE	VSS	VDD0P8_CORE	VSS	VDD0P8_DDR_CORE	VDDIO_DDR	VSS	VSS	VSS	VSS	DDR_ZQ	J	
K	MIPLRX2_CLK_N	MIPLRX2_CLK_P	GPIO25	VDDIO_BANK4	VSS	VDD0P8_CORE	VSS	VDD0P8_CORE	VSS	VDD0P8_DDR_CORE	VDDIO_DDR	VSS	VSS	VSS	VSS	VSS	K	
L	MIPLRX2_D4_P	MIPLRX2_D4_N	VSS	AVDD1P8_MIPI	VSS	VDD0P8_CORE	VSS	VDD0P8_CORE	VSS	VDD0P8_DDR_CORE	VDDIO_DDR	VSS	VSS	GPIO62	GPIO2	VSS	VDD1P1_DDR_VDDQ	L
M	MIPLRX2_D5_P	MIPLRX2_D5_N	VSS	AVDD0P8_MIPI	VSS	AVDD1P8_MIPI	VSS	VDD0P8_CORE	VSS	VDD0P8_CORE	VSS	VSS	VSS	GPIO63	GPIO5	VDD1P1_DDR_VDD2	GPIO7	M
N	MIPLRX0_D1_N	MIPLRX0_D1_P	GPIO60	VSS	AVDD0P8_MIPI	VSS	VSS	VDDIO_BANK2	VSS	VDDIO_BANK0	VDDIO_BANK5	VSS	GPIO6	GPIO4	GPIO3	GPIO8	N	
P	MIPLRX0_CLK_N	MIPLRX0_CLK_P	GPIO61	GPIO59	MIPLATB	MIPLRXT	VSS	VSS	VSS	GPIO27	GPIO26	GPIO13	GPIO12	VSS	GPIO10	GPIO11	P	
R	VSS	MIPLRX0_D0_N	MIPLRX1_D2_P	MIPLRX1_CLK_N	MIPLRX1_D3_N	MIPLTX1_D3_N	MIPLTX1_D2_P	MIPLTX0_D0_N	MIPLTX0_CLK_N	MIPLTX0_D1_N	GPIO33	GPIO32	GPIO31	GPIO30	GPIO29	GPIO28	R	
T	NC	MIPLRX0_D0_P	MIPLRX1_D2_N	MIPLRX1_CLK_P	MIPLRX1_D3_P	MIPLTX1_D3_P	MIPLTX1_D2_N	MIPLTX0_D0_P	MIPLTX0_CLK_P	MIPLTX0_D1_P	GPIO37	GPIO36	VSS	GPIO35	GPIO34	NC	T	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16		

Figure 12-9-2 K230D Pin Map

12.9.2 IO MUX and PAD control register

12.9.2.1 Function IO register list

Base Address : 0x9110_5000

PAD Name	Register Name	Offset	Default Value
PAD_IO_0	IO_CFG_0_TMP	0x00	0x944
PAD_IO_1	IO_CFG_1_TMP	0x04	0x944

PAD Name	Register Nmae	Offset	Default Value
PAD_IO_2	IO_CFG_2_TMP	0x08	0x929
PAD_IO_3	IO_CFG_3_TMP	0x0c	0x908
PAD_IO_4	IO_CFG_4_TMP	0x10	0x888
PAD_IO_5	IO_CFG_5_TMP	0x14	0x908
PAD_IO_6	IO_CFG_6_TMP	0x18	0x948
PAD_IO_7	IO_CFG_7_TMP	0x1c	0x890
PAD_IO_8	IO_CFG_8_TMP	0x20	0x890
PAD_IO_9	IO_CFG_9_TMP	0x24	0x890
PAD_IO_10	IO_CFG_10_TMP	0x28	0x910
PAD_IO_11	IO_CFG_11_TMP	0x2c	0x890
PAD_IO_12	IO_CFG_12_TMP	0x30	0x890
PAD_IO_13	IO_CFG_13_TMP	0x34	0x8A9
PAD_IO_14	IO_CFG_14_TMP	0x38	0xA9E
PAD_IO_15	IO_CFG_15_TMP	0x3c	0xABF
PAD_IO_16	IO_CFG_16_TMP	0x40	0xB9E
PAD_IO_17	IO_CFG_17_TMP	0x44	0xB9E
PAD_IO_18	IO_CFG_18_TMP	0x48	0xB9E
PAD_IO_19	IO_CFG_19_TMP	0x4c	0xB9E
PAD_IO_20	IO_CFG_20_TMP	0x50	0xB9E
PAD_IO_21	IO_CFG_21_TMP	0x54	0xB9E
PAD_IO_22	IO_CFG_22_TMP	0x58	0xB9E
PAD_IO_23	IO_CFG_23_TMP	0x5c	0xB9E
PAD_IO_24	IO_CFG_24_TMP	0x60	0xB1E
PAD_IO_25	IO_CFG_25_TMP	0x64	0xA90
PAD_IO_26	IO_CFG_26_TMP	0x68	0xABF
PAD_IO_27	IO_CFG_27_TMP	0x6c	0xB9E
PAD_IO_28	IO_CFG_28_TMP	0x70	0xB9E
PAD_IO_29	IO_CFG_29_TMP	0x74	0xB9E
PAD_IO_30	IO_CFG_30_TMP	0x78	0xB9E
PAD_IO_31	IO_CFG_31_TMP	0x7c	0xB9E
PAD_IO_32	IO_CFG_32_TMP	0x80	0xBD0
PAD_IO_33	IO_CFG_33_TMP	0x84	0xBD0
PAD_IO_34	IO_CFG_34_TMP	0x88	0xBD0
PAD_IO_35	IO_CFG_35_TMP	0x8c	0xBD0

PAD Name	Register Name	Offset	Default Value
PAD_IO_36	IO_CFG_36_TMP	0x90	0xBD0
PAD_IO_37	IO_CFG_37_TMP	0x94	0xBD0
PAD_IO_38	IO_CFG_38_TMP	0x98	0x890
PAD_IO_39	IO_CFG_39_TMP	0x9c	0x910
PAD_IO_40	IO_CFG_40_TMP	0xa0	0x890
PAD_IO_41	IO_CFG_41_TMP	0xa4	0x910
PAD_IO_42	IO_CFG_42_TMP	0xa8	0x890
PAD_IO_43	IO_CFG_43_TMP	0xac	0x910
PAD_IO_44	IO_CFG_44_TMP	0xb0	0x890
PAD_IO_45	IO_CFG_45_TMP	0xb4	0x910
PAD_IO_46	IO_CFG_46_TMP	0xb8	0x890
PAD_IO_47	IO_CFG_47_TMP	0xbc	0x910
PAD_IO_48	IO_CFG_48_TMP	0xc0	0x890
PAD_IO_49	IO_CFG_49_TMP	0xc4	0x910
PAD_IO_50	IO_CFG_50_TMP	0xc8	0x890
PAD_IO_51	IO_CFG_51_TMP	0xcc	0x910
PAD_IO_52	IO_CFG_52_TMP	0xd0	0x890
PAD_IO_53	IO_CFG_53_TMP	0xd4	0x910
PAD_IO_54	IO_CFG_54_TMP	0xd8	0x89E
PAD_IO_55	IO_CFG_55_TMP	0xdc	0x89F
PAD_IO_56	IO_CFG_56_TMP	0xe0	0x99E
PAD_IO_57	IO_CFG_57_TMP	0xe4	0x99E
PAD_IO_58	IO_CFG_58_TMP	0xe8	0x99E
PAD_IO_59	IO_CFG_59_TMP	0xec	0x99E
PAD_IO_60	IO_CFG_60_TMP	0xf0	0x890
PAD_IO_61	IO_CFG_61_TMP	0xf4	0x890
PAD_IO_62	IO_CFG_62_TMP	0xf8	0x8A9
PAD_IO_63	IO_CFG_63_TMP	0xfc	0x8A9

12.9.2.2 Function IO IO_CFG_X_TMP (X=0~63) register description

bits	Field Name	Access	Field Description
31	DI	RO	Input data from outside the Chip to the PAD
30:14	RSV	RO	Reserved

bits	Field Name	Access	Field Description
13:11	IO_SEL[2:0]	RW	Function mode select: 3'b000: Function 1; 3'b001: Function 2; 3'b010: Function 3; 3'b011: Function 4; 3'b100: Function 5;
10	SL	RW	Slew rate control enable (the dual-voltage PAD does not have this control terminal, and the configuration is invalid). 0: Slew rate is disable. 1: Slew rate is enable.
9	MSC	RW	PAD supported voltage control (single-voltage PAD fixed 1.8V, without this control terminal, the configuration is invalid) : 1 is 1.8V; 0 is 3.3V. The MSC of PAD_IO_2 is valid in the VDD3P3_IO_0(PAD_IO_2 to PAD_IO_13) domain. The MSC of PAD_IO_14 is valid in the VDD3P3_IO_1(PAD_IO_14 to PAD_IO_25) domain. The MSC of PAD_IO_26 is valid in the VDD3P3_IO_2(PAD_IO_26 to PAD_IO_37) domain. The MSC of PAD_IO_38 is valid in the VDD3P3_IO_3(PAD_IO_38 to PAD_IO_49) domain. The MSC of PAD_IO_50 is valid in the VDD3P3_IO_4(PAD_IO_50 to PAD_IO_61) domain. The MSC of PAD_IO_62 is valid in the VDD3P3_IO_5(PAD_IO_62 to PAD_IO_63) domain.
8	IE	RW	Input enable. It can disable or enable IO input. 1 is enable; 0 is disable.
7	OE	RW	Output enable. It can disable or enable IO output. 1 is enable; 0 is disable.
6	PU	RW	Pull Up enable. 1 is pull up.
5	PD	RW	Pull Down enable. 1 is pull down.
4:1	DS[3:0]	RW	Driving selector. Single-voltage PAD is 8-drive PAD, DS[2:0] is valid, DS[3] is invalid, the value is: 3'b000~3'b111; The dual-voltage PAD is a 16-drive PAD, and DS[3:0] is valid. The value ranges from 4'b0000 to 4'b1111.
0	ST	RW	Schmitt trigger control enable. 1: schmitt trigger is enable; 0: schmitt trigger is disable.

12.9.2.3 PMU IO register list

Base Address : 0x9100_0000

PAD Name	Register Name	Offset	Default Value
PAD_IO_64	io_cfg_0	0x80	0x1124
PAD_IO_65	io_cfg_1	0x84	0x1124
PAD_IO_66	io_cfg_2	0x88	0x1124
PAD_IO_67	io_cfg_3	0x8c	0x1124
PAD_IO_68	io_cfg_4	0x90	0x1124
PAD_IO_69	io_cfg_5	0x94	0x1124
PAD_IO_70	io_cfg_6	0x98	0x10A4
PAD_IO_71	io_cfg_7	0x9c	0x10C4

12.9.2.4 PMU IO io_cfg_X (X=0~7) register description

bits	Field Name	Access	Field Description
31:13	RSV	RO	Reserved
12:11	IO_SEL[1:0]	RW	Function mode select: 2'b01: Function 1; 2'b10: Function 2;
10	SL	RW	Slew rate control enable (the dual-voltage PAD does not have this control terminal, and the configuration is invalid). 0: Slew rate is disable. 1: Slew rate is enable.
9	RSV	RO	Reserved
8	IE	RW	Input enable. It can disable or enable IO input. 1 is enable; 0 is disable.
7	OE	RW	Output enable. It can disable or enable IO output. 1 is enable; 0 is disable.
6	PU	RW	Pull Up enable. 1 is pull up.
5	PD	RW	Pull Down enable. 1 is pull down.
4	RSV	RO	Reserved
3:1	DS[2:0]	RW	Driving selector. The value is: 3'b000~3'b111.
0	ST	RW	Schmitt trigger control enable. 1: schmitt trigger is enable; 0: schmitt trigger is disable.

13 Security Modules

13.1 Overview

The security subsystem supports the functional specifications of the symmetric algorithm, asymmetric algorithm and hash algorithm.

- Support following symmetric algorithm
 - AES128/192/256
 - CMAC128/192/256
 - SM4-128
- Support following asymmetric algorithm
 - RSA-1024/2048/3072/4096
 - RSA encrypt/decrypt/signature/verify
 - ECC-163/233/283/409/571/192/224/256/384/521
 - ECDSA-233/283/409/571/224/256/384/521
 - ECC CDH-163/233/283/409/571/192/224/256/384/521
 - SM2 encrypt/decrypt/signature/verify
- Support following hash algorithm
- HMAC
- SHA224/256/384/512, SHA_512/224, SHA_512/256
- SM3

13.2 features

- TRNG with a PUF-based refinement engine. PUFtrng leverages 1Kb of PUF data for initial seeding.
 - 160Mbps@20Mhz
 - Stabilization time minimum than 50us
- OTP program region has raw OTP density 32Kbits, with security protection
- Support security boot with signature
- Public Key Cryptography (digital signature, key agreement)
 - NIST standard Elliptic Curve (up to P521, B571)
 - ECDSA: FIPS 186-4 Elliptic Curve Digital Signature Algorithm
 - ECDH: NIST SP800-56A Elliptic Curve Diffie-Hellman
 - RSA: FIPS 186-4 (digital signature, validation only) (up to 4096)
 - SM2: GMT-0003
- Private Key Cryptography (encryption, decryption, key wrapping)
 - Cipher: AES128/192/256 , SM4 (GMT-0002)
 - Mode of operation: NIST SP800-38A/B/C/D/E
 - Key wrapping: NIST SP800-38F
- Hash Function (data integrity, message authentication code, key derivation)
 - Hash, HMAC: SHA224/256/384/512, SHA512_224/256, SM3 (GMT-0004)
 - KDF: KBKDF (CTR/FB), PBKDF

14 PMU

14.1 overview

PMU (power management unit) is used to control the core power on and off. The power management unit (PMU) controls output IOs to enable or disable the PMIC power supply to CORE domain to achieve low power consumption control in the CORE domain. The PADS inputs, RTC timer interrupts, or configurable register values can be programmed to control the outputs to control the external PMIC.

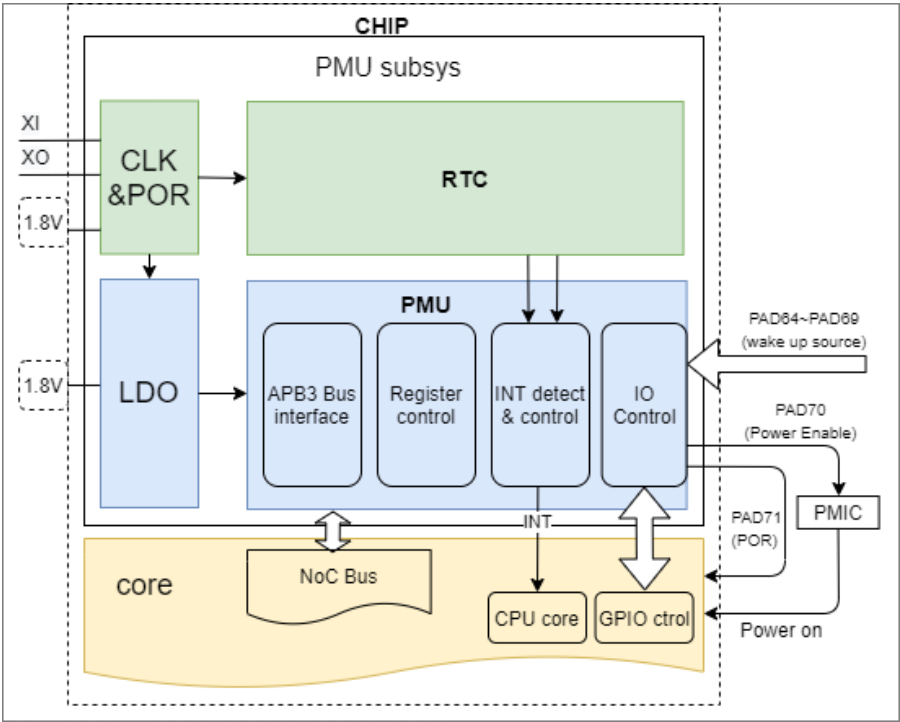


Figure 14-1-1 PMU block diagram

14.2 features

- support RTC function in 32Khz
- support the default 6 inputs and 2 outputs
- support IO configurable, reusable as GPIOs, the highest rate up to 25 MHZ
- support a set of long and short press detection function
- support a set of edge number counter
- support two sets of edge and level detection with debounce
- support four groups of edge and level detection IOs
- support configurable interrupt operation
- support 512 bits system log when it is been shutdown
- support two sets of output IOs with configurable delay time

14.3 Function Description

The following is the detail description of PMU.

name	bit width	direction	edge triggered	level triggered	debounce	edge number detection	long or short press detection	application
	internal interrupt source							

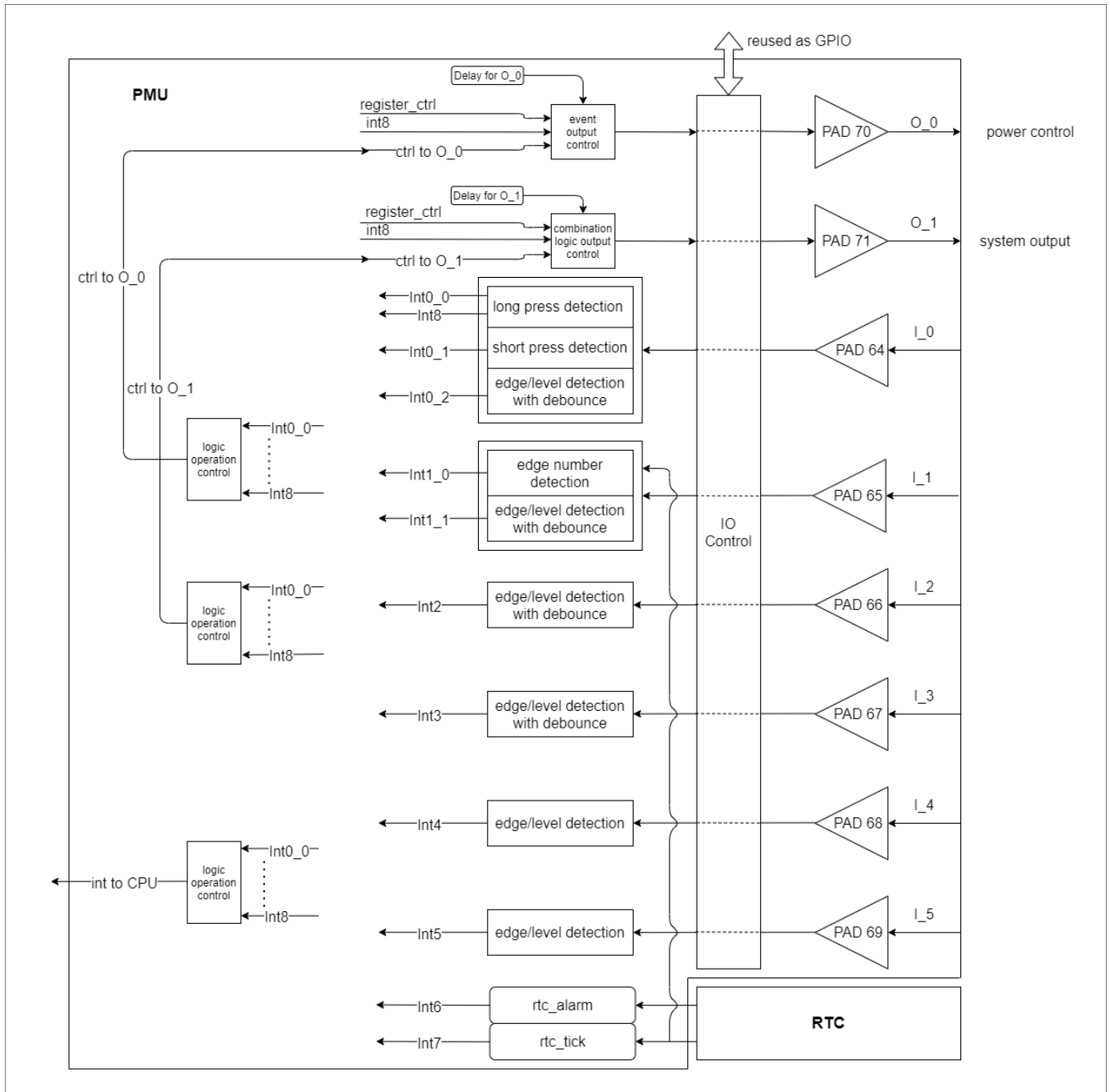
name	bit width	direction	edge triggered	level triggered	debounce	edge number detection	long or short press detection	application
RTC interrupt	2	internal	support	support	NA	-	-	timed wakeup
	external interrupt source							
PAD_64	1	input	support	support	support	-	support	wakeup input/GPIO
PAD_65	1	input	support	support	support	support	-	wakeup input/GPIO
PAD_66	1	input	support	support	-	-	-	wakeup input/GPIO
PAD_67	1	input	support	support	-	-	-	wakeup input/GPIO
PAD_68	1	input	-	support	-	-	-	wakeup input/GPIO
PAD_69	1	input	-	support	-	-	-	wakeup input/GPIO
	interrupt to cpu							
int_to_cpu	1	internal	-	-	-	-	-	int to CPU
	output IO							
PAD_69	1	output	-	-	-	-	-	PMIC power on enable/GPIO
PAD_69	1	output	-	-	-	-	-	system reset

At present, in the cold start state, the pmu only enables the long-press interrupt of PAD64 and the high-level input interrupt of PAD68 as the trigger source of the pmu module startup.

When the pmu receives either of the above two interrupts, the PAD70 will be raised to enable the external PMIC to supply power to the core domain. After the reset signal PAD71 lasts for about 50ms, it will be dereset as the normal system power on reset signal(POR).

There are iso isolated signals on both sides of the Normal domain and the pmu domain. The signals from the normal domain to the pmu domain are in the pmu domain and are automatically removed after the PAD70 is enabled for about 60ms. Signals from the pmu to the normal domain are enabled by iso by default in the normal domain. If you want to access the pmu, you need to enable ISO control registers in sysctl.

When the core domain needs to be powered off, the software needs to ensure that the effective data in the core domain has been stored and the last request is sent after there is no effective operation command.



The PMU supports different combinations logic control of six groups of input PAD signals and interrupt signals of the digital RTC module to generate the required output signals and interrupt to cpu. Six groups of input pads support the following interrupt detection functions:

1. The PAD64 input pair have three detection units, namely long press detection, short press detection, edge or level detection with debounce function. Among them, the long-press detection has two independent detection logic, respectively set different trigger thresholds, for example, 3s can generate the ordinary long-press interrupt signal, and 8s can generate the long-press shutdown interrupt trigger signal. Short press detection can also set a trigger threshold, such as 500ms can trigger short press interrupt. Edge or level detection can be set to detect input signal skipping edge or level to generate edge or level interrupt signal. As a whole, PAD64 can detect and generate four interrupt signals, int8 (long press to power off), int0_0 (long press), int0_1 (short press), int0_2 (edge or level), etc.
2. The PAD65 input pair have two detection units, which are quantity detection, edge detection with debounce function or level detection. Edge quantity detection is to set the quantity threshold of detecting rising edge or falling edge. When the set value is reached, the corresponding interrupt signal can be generated. At the same

time, the detection unit can also enable the use of RTC tick interrupt signal to periodically clear edge number count, for example, when RTC tick interrupt comes, edge number has not reached the preset value, the edge number register will clear to zero and start counting again. Edge or level detection can be set to detect input signal skipping edge or level to generate edge or level interrupt signal. Overall, PAD65 can detect and generate two interrupt signals respectively, int1_0 (along the number), int1_1 (edge or level) and so on.

- 3. The PAD66 input corresponds to an edge or level detection unit with debounce. Edge or level detection can be set to detect input signal skipping edge or level to generate edge or level interrupt signal. PAD66 can be detected to produce int2 (edge or level).
- 4. PAD67 Input corresponds to a edge or level detection unit with debounce. Edge or level detection can be set to detect input signal skipping edge or level to generate edge or level interrupt signal. PAD67 can be detected to produce int3 (edge or level).
- 5. PAD68 input corresponds to an edge or level detection unit. Edge or level detection can be set to detect input signal skipping edge or level to generate edge or level interrupt signal. PAD68 can be detected to produce int4 (edge or level).
- 6. PAD69 input corresponds to an edge or level detection unit. Edge or level detection can be set to detect input signal skipping edge or level to generate edge or level interrupt signal. PAD69 can be detected to produce int5 (edge or level).

There are two interrupt signals from RTC module, namely RTC alarm interrupt and RTC tick interrupt. The interrupt signals generated by the other six input pads can be enabled in the PMU module for logical operation respectively. The events that generate the interrupt signal int_to_cpu and control the output logic of the two output pads trigger the output control unit (O_0) and logic output control unit (O_1) signals. For an introduction to the two RTC interrupts, see Introduction to the RTC Module. The functions of the two output pads are as follows:

- 1. PAD70 corresponds to the event triggering output signal of the control unit. The unit corresponds to three groups of input signals, respectively O_0, register configuration input and long-press shutdown int8, event trigger is pulse detection, when detected along the output will change, the output can also be superimposed preset delay time output, as the enable signal of PMIC power supply.
- 2. PAD71 corresponds to the output signal of the logic output control unit. The unit corresponds to three groups of input signals, respectively O_1, register configuration input and long press off int8, the output changes with the input, but also can overlay preset delay time output, as the reset signal of the system.

14.4 Register Summary

Table 14-1 describes PMU registers.

Offset Address	Register	Description
0x0000~0x003C	PMU_LOGGER_REGISTER	PMU logger registers
0x0040	PMU_INT0_TO_CTL_REGISTER	PMU int0 ctrl register
0x0044	PMU_INT1_TO_CTL_REGISTER	PMU int1 ctrl register
0x0048	PMU_INT0_TO_CPU_REGISTER	PMUint0toCPU
0x004C	PMU_INT_DETECT_EN	PMUintdetectionenableregister

Offset Address	Register	Description
0x0050	PMU_INT_DETECT_TYP	PMUinttypeconfigregister
0x0054	PMU_INT_DETECT_CLR	PMUintclearregister
0x0058	PMU_INT0_LONG_PRESS_TRIGGER_VAL	PMUint0longpressvalue
0x005c	PMU_INT0_LONG_PRESS_SHUTDOWN_VAL	PMUint0longpressshutdownvalue
0x0060	PMU_INT0_SHORT_PRESS_TRIGGER_VAL	PMUint0shortpressvalue
0x0064	PMU_INT0_LEVEL_DEBOUNCE_VAL	PMUint0debouncevalue
0x0068	PMU_INT1_EDGE_NUMBER_VAL	PMUint1edgenumber
0x006c	PMU_INT1_LEVEL_DEBOUNCE_VAL	PMUint1debouncevalue
0x0070	PMU_INT2_LEVEL_DEBOUNCE_VAL	PMUint2debouncevalue
0x0074	PMU_INT3_LEVEL_DEBOUNCE_VAL	PMUint3debouncevalue
0x0078	PMU_SYSCTRL_REG	PMUsoftwareconfigregister
0x007c	PMU_OUT_DELAY_SET	PMUoutputdelaysetregister
0x0080~0x0094	PMU_IO_CFG_0~PMU_IO_CFG_5	PMUioconfigureregister
0x0098	PMU_IO_CFG_6	PMUioconfigureregister
0x009c	PMU_IO_CFG_7	PMUioconfigureregister
0x00a0	PMU_NORMAL_TIMER_VAL	PMUcoredomainreadyvalue
0x00a4	PMU_OUT_EVENT_CTRL	PMUeventtriggeredoutputcontrolregister
0x00a8	PMU_OUT_LOGIC_CTRL	PMUlogictriggeredoutputcontrolregister
0x00ac	PMU_INT_STATE_REG	PMUinterruptstateregister

14.5 Register Description

PMU_LOGGER_REGISTER PMU_LOGGER_REGISTER is the logger register when CPU domain shutdown offset Address: 0x0000~0x003C Total reset value: 0x0

Bits	Access	Name	Description	Reset
[31:0]	R/W	Logger0~Logger15	Core domain information saved by CPUs	0x0

PMU_INT0_TO_CTL_REGISTER PMU_INT0_TO_CTL_REGISTER is the interrupt enable register for for output pad70 offset Address: 0x0040 Total reset value: 0xFFFF

Bits	Access	Name	Description	Reset
[31:13]	--	Reserved		
[12]	R/W	int_to_ctrl_0_logic_set	Interrupt logical Combination: 0: Or operation; 1: And operation	0x0
[11]	R/W	int0_to_CTRL_0_set	int0_0 interrupt 1: Enable. 0: disable	0x1

Bits	Access	Name	Description	Reset
[10]	R/W	int0_to_ctrl_0_1_set	int0_1 interrupt 1: Enable. 0: disable	0x1
[9]	R/W	int0_to_ctrl_0_2_set	int0_2 interrupt 1: Enable. 0: disable	0x1
[8]	R/W	int0_to_ctrl_1_0_set	int1_0 interrupt 1: Enable. 0: disable	0x1
[7]	R/W	int0_to_ctrl_1_1_set	int1_1 interrupt 1: Enable. 0: disable	0x1
[6]	R/W	int0_to_ctrl_2_set	int2 interrupt 1: Enable. 0: disable	0x1
[5]	R/W	int0_to_ctrl_3_set	int3 interrupt 1: Enable. 0: disable	0x1
[4]	R/W	int0_to_ctrl_4_set	Int4 interrupt 1: Enable. 0: disable	0x1
[3]	R/W	int0_to_ctrl_5_set	int5 interrupt 1: Enable. 0: disable	0x1
[2]	R/W	int0_to_ctrl_6_set	Int6 interrupt 1: Enable. 0: disable	0x1
[1]	R/W	int0_to_ctrl_7_set	Int7 interrupt 1: Enable. 0: disable	0x1
[0]	R/W	int0_to_ctrl_8_set	Int8 interrupt 1: Enable. 0: disable	0x1

PMU_INT1_TO_CTL_REGISTER PMU_INT1_TO_CTL_REGISTER is the interrupt enable register for output pad71
offset Address: 0x0044 Total reset value: 0xFFFF

Bits	Access	Name	Description	Reset
[31:13]	--	Reserved		
[12]	R/W	int_to_ctrl_0_logic_set	Interrupt logical Combination: 0: Or operation; 1: And operation	0x0
[11]	R/W	int0_to_CTRL_0_set	int0_0 interrupt 1: Enable. 0: disable	0x1
[10]	R/W	int0_to_ctrl_0_1_set	int0_1 interrupt 1: Enable. 0: disable	0x1
[9]	R/W	int0_to_ctrl_0_2_set	int0_2 interrupt 1: Enable. 0: disable	0x1
[8]	R/W	int0_to_ctrl_1_0_set	int1_0 interrupt 1: Enable. 0: disable	0x1
[7]	R/W	int0_to_ctrl_1_1_set	int1_1 interrupt 1: Enable. 0: disable	0x1

Bits	Access	Name	Description	Reset
[6]	R/W	int0_to_ctrl_2_set	int2 interrupt 1: Enable. 0: disable	0x1
[5]	R/W	int0_to_ctrl_3_set	int3 interrupt 1: Enable. 0: disable	0x1
[4]	R/W	int0_to_ctrl_4_set	Int4 interrupt 1: Enable. 0: disable	0x1
[3]	R/W	int0_to_ctrl_5_set	int5 interrupt 1: Enable. 0: disable	0x1
[2]	R/W	int0_to_ctrl_6_set	Int6 interrupt 1: Enable. 0: disable	0x1
[1]	R/W	int0_to_ctrl_7_set	Int7 interrupt 1: Enable. 0: disable	0x1
[0]	R/W	int0_to_ctrl_8_set	Int8 interrupt 1: Enable. 0: disable	0x1

PMU_INT0_TO_CPU_REGISTER PMU_INT0_TO_CPU_REGISTER is the interrupt enable register for int to CPU
offset Address: 0x0048 Total reset value: 0x0

Bits	Access	Name	Description	Reset
[31:13]	--	Reserved		
[12]	R/W	int_to_ctrl_0_logic_set	Interrupt logical Combination: 0: Or operation; 1: And operation	0x0
[11]	R/W	int0_to_CTRL_0_set	int0_0 interrupt 1: Enable. 0: disable	0x0
[10]	R/W	int0_to_ctrl_0_1_set	int0_1 interrupt 1: Enable. 0: disable	0x0
[9]	R/W	int0_to_ctrl_0_2_set	int0_2 interrupt 1: Enable. 0: disable	0x0
[8]	R/W	int0_to_ctrl_1_0_set	int1_0 interrupt 1: Enable. 0: disable	0x0
[7]	R/W	int0_to_ctrl_1_1_set	int1_1 interrupt 1: Enable. 0: disable	0x0
[6]	R/W	int0_to_ctrl_2_set	int2 interrupt 1: Enable. 0: disable	0x0
[5]	R/W	int0_to_ctrl_3_set	int3 interrupt 1: Enable. 0: disable	0x0
[4]	R/W	int0_to_ctrl_4_set	Int4 interrupt 1: Enable. 0: disable	0x0
[3]	R/W	int0_to_ctrl_5_set	int5 interrupt 1: Enable. 0: disable	0x0

Bits	Access	Name	Description	Reset
[2]	R/W	int0_to_ctrl_6_set	Int6 interrupt 1: Enable. 0: disable	0x0
[1]	R/W	int0_to_ctrl_7_set	Int7 interrupt 1: Enable. 0: disable	0x0
[0]	R/W	int0_to_ctrl_8_set	Int8 interrupt 1: Enable. 0: disable	0x0

PMU_INT_DETECT_EN PMU_INT_DETECT_EN is the interrupt detection enable register offset Address: 0x004C
Total reset value: 0x810

Bits	Access	Name	Description	Reset
[31:13]	--	Reserved		
[12]	R/W	int0_shutdown_press_en	int0 long press to shut down detection 1: Enable. 0: disable	0x0
[11]	R/W	int0_long_press_en	int0 long press detection 1: Enable. 0: disable	0x1
[10]	R/W	int0_short_press_en	int0 short press detection 1: Enable. 0: disable	0x0
[9]	R/W	int0_edge_level_en	int0 edge/level detection 1: Enable. 0: disable	0x0
[8]	R/W	int1_edge_num_en	int1 edge number detection 1: Enable. 0: disable	0x0
[7]	R/W	int1_edge_level_en	int1 edge/level detection 1: Enable. 0: disable	0x0
[6]	R/W	int2_edge_level_en	int2 edge/level detection 1: Enable. 0: disable	0x0
[5]	R/W	int3_edge_level_en	int3 edge/level detection 1: Enable. 0: disable	0x0
[4]	R/W	int4_edge_level_en	Int4 edge/level detection 1: Enable. 0: disable	0x1
[3]	R/W	int5_edge_level_en	int5 edge/level detection 1: Enable. 0: disable	0x0
[2]	R/W	int6_en	Int6 rtc alarm detection 1: Enable. 0: disable	0x0
[1]	R/W	int7_en	Int7 rtc tick detection 1: Enable. 0: disable	0x0
[0]	R/W	int1_edge_num_tick_en	Int1 edge number detection 1: Enable. 0: disable	0x0

PMU_INT_DETECT_TYP PMU_INT_DETECT_TYP is the interrupt detection type configure register offset Address: 0x0050
Total reset value: 0x0

Bits	Access	Name	Description	Reset
[31:21]	--	Reserved		
[20]	R/W	int0_long_press_level_eff_val	int0 effective long press level value 0: high; 1: low	0
[19]	R/W	int0_short_press_level_eff_val	int0 effective short press level value 0: high; 1: low	0
[18]	R/W	int0_level_eff_val	int0 effective level value 0: high; 1: low	0
[17]	R/W	int0_edge_eff_val	int0 effective edge type 0: rise; 1: down	0
[16]	R/W	int0_edge_level_trigger_type	int0 effective trigeer type 0: level; 1: edge	0
[15]	R/W	int1_edge_num_eff_val	int1 effective edge type 0:rise; 1: down	0
[14]	R/W	int1_level_eff_val	int1 effective level type 0: high; 1: low	0
[13]	R/W	int1_edge_eff_val	int1 effective edge type 0: rise; 1: down	0
[12]	R/W	int1_edge_level_trigger_type	int1 effective trigeer type 0: level; 1: edge	0
[11]	R/W	int2_level_eff_val	int2 effective level type 0: high; 1: low	0
[10]	R/W	int2_edge_eff_val	int2 effective edge type 0: rise; 1: down	0
[9]	R/W	int2_edge_level_trigger_type	int2 effective trigeer type 0: level; 1: edge	0
[8]	R/W	int3_level_eff_val	int3 effective level type 0: high; 1: low	0
[7]	R/W	int3_edge_eff_val	int3 effective edge type 0: rise; 1: down	0
[6]	R/W	int3_edge_level_trigger_type	int3 effective trigeer type 0: level; 1: edge	0
[5]	R/W	int4_level_eff_val	int4 effective level type 0: high; 1: low	0
[4]	R/W	int4_edge_eff_val	int4 effective edge type 0: rise; 1: down	0
[3]	R/W	int4_edge_level_trigger_type	int4 effective trigeer type 0: level; 1: edge	0
[2]	R/W	int5_level_eff_val	int5 effective level type 0: high; 1: low	0

Bits	Access	Name	Description	Reset
[1]	R/W	int5_edge_eff_val	int5 effective edge type 0: rise; 1: down	0
[0]	R/W	int5_edge_level_trigger_type	int5 effective trigeer type 0: level; 1: edge	0

PMU_INT_DETECT_CLR PMU_INT_DETECT_CLR is the interrupt clear register offset Address: 0x0054 Total reset value: 0x0

Bits	Access	Name	Description	Reset
[31:10]	--	Reserved		
[9:8]	W1C	int0_long_press_clr	1x: int0 long press interrupt clear; x1 int0 long press to shutdown interrupt clear;	0
[7]	W1C	int0_short_press_clr	Int0 short press interrupt clear	0
[6]	W1C	int0_edge_level_clr	Int0 edge/level interrupt clear	0
[5]	W1C	int1_edge_num_clr	Int1 edge number interrput clear	0
[4]	W1C	int1_edge_level_clr	Int1 edge/level interrupt clear	0
[3]	W1C	int2_edge_level_clr	Int2 edge/level interrupt clear	0
[2]	W1C	int3_edge_level_clr	Int3 edge/level interrupt clear	0
[1]	W1C	int4_edge_level_clr	Int4 edge/level interrupt clear	0
[0]	W1C	int5_edge_level_clr	Int5 edge/level interrupt clear	0

PMU_INT0_LONG_PRESS_TRIGGER_VAL PMU_INT0_LONG_PRESS_TRIGGER_VAL is the long press trigger register for PAD64 offset Address: 0x0058 Total reset value: 0x17700

Bits	Access	Name	Description	Reset
[31:20]	--	Reserved		
[19:0]	R/W	int0_long_press_trigger_val	effetive long press interrupt time in 32Khz	0x17700

PMU_INT0_LONG_PRESS_SHUTDOWN_VAL PMU_INT0_LONG_PRESS_SHUTDOWN_VAL is used to set the shutdown trigger value for PAD64 offset Address: 0x005c Total reset value: 0x4e200

Bits	Access	Name	Description	Reset
[31:20]	--	Reserved		
[19:0 }	R/W	int0_long_press_shutdown_val	effective long press to shutdown interrupt time in 32Khz	0x4E200

PMU_INT0_SHORT_PRESS_TRIGGER_VAL PMU_INT0_SHORT_PRESS_TRIGGER_VAL is the short press trigger register for PAD64 offset Address: 0x0060 Total reset value: 0x3E800

Bits	Access	Name	Description	Reset
[31:16]	--	Reserved		
[15:0]	R/W	int0_short_press_trigger_val	effective short press interrupt time in 32Khz	0x3E800

PMU_INT0_LEVEL_DEBOUNCE_VAL PMU_INT0_LEVEL_DEBOUNCE_VAL is the level debounce value for PAD64 offset Address: 0x0064 Total reset value: 0x100

Bits	Access	Name	Description	Reset
[31:12]	--	Reserved		
[11:0]	R/W	int0_level_debounce_val	effective debounce time for level detection intrrupt in 32Khz	12'h100

PMU_INT1_EDGE_NUMBER_VAL PMU_INT1_EDGE_NUMBER_VAL is edge number trigger value for PAD65 offset Address: 0x0068 Total reset value: 0x1

Bits	Access	Name	Description	Reset
[31:10]	--	Reserved		
[9:0]	R/W	int1_edge_number_val	int1 effective edge numbers in 32Khz	10'b1

PMU_INT1_LEVEL_DEBOUNCE_VAL PMU_INT1_LEVEL_DEBOUNCE_VAL is the debounce value register for PAD65 offset Address: 0x006c Total reset value: 0x100

Bits	Access	Name	Description	Reset
[31:12]	--	Reserved		
[11:0]	R/W	int1_level_debounce_val	int1 effective level debounce time in 32Khz	12'h100

PMU_INT2_LEVEL_DEBOUNCE_VAL PMU_INT2_LEVEL_DEBOUNCE_VAL is the debounce value register for PAD66 offset Address: 0x0070 Total reset value: 0x100

Bits	Access	Name	Description	Reset
[31:12]	--	Reserved		
[11:0]	R/W	int2_level_debounce_val	int2 effecitve level debounce time in 32Khz	12'h100

PMU_INT3_LEVEL_DEBOUNCE_VAL PMU_INT3_LEVEL_DEBOUNCE_VAL is the debounce value register for PAD67 offset Address: 0x0074 Total reset value: 0x100

Bits	Access	Name	Description	Reset
[31:12]	--	Reserved		
[11:0]	R/W	int2_level_debounce_val	int3 effecitve level debounce time in 32Khz	12'h100

PMU_SYSCTRL_REG PMU_SYSCTRL_REG is the software configured value to control the output PAD70 and PAD71 offset Address: 0x0078 Total reset value: 0x1

Bits	Access	Name	Description	Reset
[31:2]	--	Reserved		
[1]	R/W	to_ctrl_1_reg_o	value to combination logic output control block	0
[0]	R/W	to_ctrl_0_reg_o	value to event output control block. edge toggle through configuration is necessary	1

PMU_OUT_DELAY_SET PMU_OUT_DELAY_SET is the output delay value for output PAD70 and PAD71 offset Address: 0x007c Total reset value: 0x1100640

Bits	Access	Name	Description	Reset
[31:26]	--	Reserved		
[25]	R/W	event_delay_en_o	enable output delay on event output control block. 1:enable;0:disable	0
[24]	R/W	logic_delay_en_o	enable output delay on combination logic output control block. 1:enable;0:disable	1
[23:12]	R/W	event_delay_val_o	event output control block delay time in 32khz	12'h100
[11:0]	R/W	logic_delay_val_o	combination logic output control block delay time in 32khz	12'h640

PMU_IO_CFG_0~PMU_IO_CFG_5 PMU_IO_CFG_0~ PMU_IO_CFG_5 is the PAD configuration registers from PAD64 to PAD69 offset Address: 0x0080~0x0094 Total reset value: 0x1124

Bits	Access	Name	Description	Reset
[31:24]	--	Reserved		
[13:11]	R/W	IO_SEL	mode selection 010: interrupt input;001: reuse as GPIO	3'b010
[10]	R/W	SL		1'b0
[9]	-	-		-
[8]	R/W	IE	1:input enable; 0:input disable	1'b1
[7]	R/W	OE	1:output enable; 0:output disable	1'b0
[6]	R/W	PU	pull up	1'b0
[5]	R/W	PD	pull down	1'b1
[4]	-	-		-
[3]	R/W	DS2		1'b0
[2]	R/W	DS1		1'b1
[1]	R/W	DS0		1'b0
[0]	R/W	ST		1'b0

PMU_IO_CFG_6 PMU_IO_CFG_6 is the PAD configuration register to PAD70 offset Address: 0x0098 Total reset value: 0x10a4

Bits	Access	Name	Description	Reset
[31:14]	--	Reserved		
[13:11]	R/W	IO_SEL	mode selection 010: interrupt input;001: reuse as GPIO	3'b010
[10]	R/W	SL		1'b0
[9]	-	-		-

Bits	Access	Name	Description	Reset
[8]	R/W	IE	1:input enable; 0:input disable	1'b0
[7]	R/W	OE	1:output enable; 0:output disable	1'b1
[6]	R/W	PU	pull up	1'b0
[5]	R/W	PD	pull down	1'b1
[4]	-	-		-
[3]	R/W	DS2		1'b0
[2]	R/W	DS1		1'b1
[1]	R/W	DS0		1'b0
[0]	R/W	ST		1'b0

PMU_IO_CFG_7 PMU_IO_CFG_6 is the PAD configuration register to PAD71 offset Address: 0x009c Total reset value: 0x1084

Bits	Access	Name	Description	Reset
[31:14]	--	Reserved		
[13:11]	R/W	IO_SEL	mode selection 010: interrupt input;001: reuse as GPIO	3'b010
[10]	R/W	SL		1'b0
[9]	-	-		-
[8]	R/W	IE	1:input enable; 0:input disable	1'b0
[7]	R/W	OE	1:output enable; 0:output disable	1'b1
[6]	R/W	PU	pull up	1'b0
[5]	R/W	PD	pull down	1'b0
[4]	-	-		-
[3]	R/W	DS2		1'b0
[2]	R/W	DS1		1'b1
[1]	R/W	DS0		1'b0
[0]	R/W	ST		1'b0

PMU_NORMAL_TIMER_VAL PMU_NORMAL_TIMER_VAL is the isolation time from CPU domain power on to stable operation offset Address: 0x00a0 Total reset value: 0x780

Bits	Access	Name	Description	Reset
[31:16]	--	Reserved		
[15:0]	R/W	normal_timer_val_reg	Wait for the core domain power supply to stabilize. The preset time is 60ms	16'h780

PMU_OUT_EVENT_CTRL PMU_OUT_EVENT_CTRL is configuration register for output PAD70 offset Address: 0x00a4 Total reset value: 0x5c

Bits	Access	Name	Description	Reset
[31:13]	--	Reserved		
[12:5]	R/W	normal_sta_timeout_val	No response timeout count in 32Khz to power off request which control output pad Disable PMIC	8'h2
[4:2]	R/W	to_ctrl_0_set	Enable the inputs to event output block. 1: effective; 0: Disable. Bit2: PMU_INT0_TO_CTL_REGISTER Interrupt logic results Bit3: int0 long press to power off int8 Long press power off Bit4: Software register configuration value PMU_SYSCTRL_REG When event_mode is pulse triggered, PMU_INT0_TO_CTL_REGISTER and PMU_SYSCTRL_REG usually used as power-on trigger, and int8 and PMU_SYSCTRL_REG usually used as power-off trigger. When event_mode is in level triggered mode, all these inputs can be used	3'h7
[1]	R/W	to_ctrl_0_logic_set	logical operations for inputs to event output block 0: logic "or" ; 1: logic "and"	0
[0]	R/W	event_mode	trigger mode of the event output control block. 0: pulse. 1: Level	0

PMU_OUT_LOGIC_CTRL PMU_OUT_LOGIC_CTRL is the configuration register for output PAD71 offset Address: 0x00a8 Total reset value: 0x2

Bits	Access	Name	Description	Reset
[31:4]	--	Reserved		
[3:1]	R/W	to_ctrl_0_set	Enable the inputs to event output block. 1: effective; 0: Disable. Bit1: PMU_INT0_TO_CTL_REGISTER Interrupt logic results Bit2: int0 long press to power off int8 Long press power off Bit3: Software register configuration value PMU_SYSCTRL_REG When event_mode is pulse triggered	3'h1
[0]	R/W	to_ctrl_1_logic_set	logical operations for inputs to combination logic output block 0: logic "or" ; 1: logic "and"	0

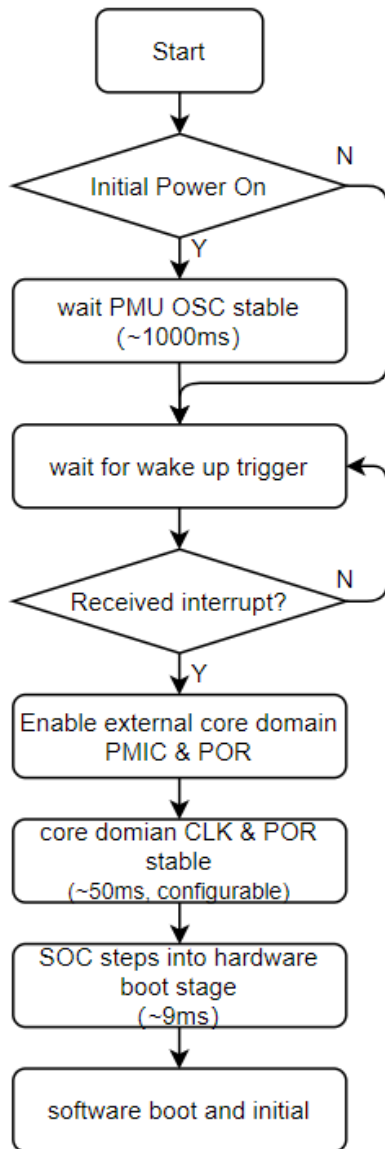
PMU_INT_STATE_REG PMU_INT_STATE_REG is the interrupt state register offset Address: 0x00ac Total reset value: 0x0

Bits	Access	Name	Description	Reset
[31:14]	--	Reserved		
[13]	RO	rtc_alarm_int_state	1: rtc alarm interrupt inputs state; 0: NA	0x0
[12]	RO	rtc_tick_int_state	1: rtc tick interrupt input state; 0: NA	0x0
[11]	RO	int0_0_state	1: int0 long press to power off interrupt triggered ; 0: NA	0x0

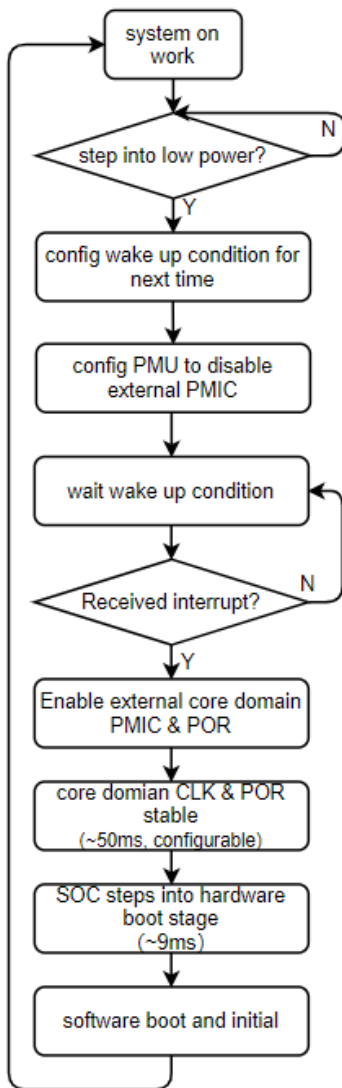
Bits	Access	Name	Description	Reset
[10]	RO	int0_1_state	1: int0 short press interrupt triggered ; 0: NA	0x0
[9]	RO	int0_2_state	1: int0 edge/level press interrupt triggered ; 0: NA	0x0
[8]	RO	int1_0_state	1: int1 edge number interrupt triggered ; 0: NA	0x0
[7]	RO	int1_1_state	1: int1 edge/level interrupt triggered ; 0: NA	0x0
[6]	RO	int2_state	1: int2 edge/level interrupt triggered ; 0: NA	0x0
[5]	RO	int3_state	1: int3 edge/level interrupt triggered ; 0: NA	0x0
[4]	RO	Int4_state	1: int4 edge/level interrupt triggered ; 0: NA	0x0
[3]	RO	Int5_state	1: int5 edge/level interrupt triggered ; 0: NA	0x0
[2]	RO	Int6_state	1: rtc alarm interrupt triggered ; 0: NA	0x0
[1]	RO	Int7_state	1: rtl tick interrupt triggered ; 0: NA	0x0
[0]	RO	Int8_state	1: int0 long press to poweroff interrupt triggered ; 0: NA	0x0

14.6 WorkingFlow

14.6.1 Cold Reset Flow



14.6.2 Working to low power and wake up flow



15 Decompress Engine

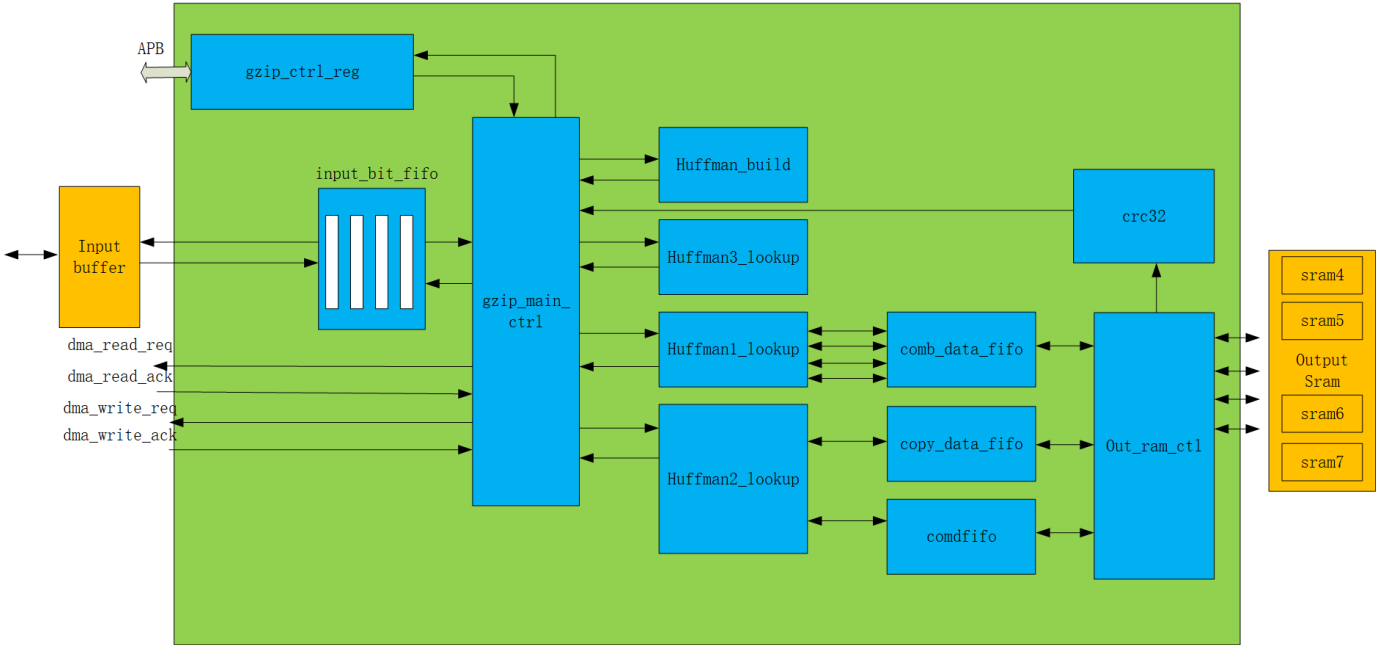
15.1 Overview

Decompression accelerator is mainly used to implement hardware GZIP decompression function. Key features:

- Support GZIP file format version 4.3
- Support DEFLATE compressed file version 1.3
- The maximum dictionary size is 32KB
- Support the minimum word matching length of 3Byte and the maximum word matching length of 258Byte
- Only dynamic huffman compression mode is supported
- Support data stream CRC32 verification
- Support hardware to automatically generate dynamic huffman tree
- The APB3 register interface is used to configure the startup decompression accelerator
- Request the DMA controller to transfer large blocks through hardware, and request to transfer 32Kbyte at most once
- Minimum decompression bandwidth 400MB/s

15.2 Block Diagram

Figure 15-2-1 block diagram



15.3 Programming Guideline

- Configure the decompression work enable. The maximum bit of the offset address 0x4 is 1
- Configure the linked list of DMA controller channel 0 and channel 1. Where channel 0 is written from DDR to decompressed SRAM, and the source address is the address of the stored compressed file The target address is the base address of the decompressed SRAM, the length is the length of the compressed file, and the mode is configured as the hardware request mode. Channel 1 is set to read the DDR from the unzipped SRAM. The source address is the unzipped SRAM address, the target address is the base address stored in the DDR after decompression, and the length is the unzipped length in the packet header.
- Start the DMA controller and wait for the decompression module hardware request.
- Configure the dma of the decompression controller_out_Size (data length after decompression), dma_in_Size (the data length before decompression (this length needs to be 0x80000000)), and then configure the decompression start.
- At this time, the DMA controller and decompression start to shake hands until decompression and DMA are completed. Finally, it is necessary to see the completion interrupt of channel 1. When all the linked lists of channel 1 are completed, decompression is completed.

15.4 register summary

Offset Address	Register	Description
0x0	DECOMP_START	start decompression controller
0x4	GZIP_SRC_SIZE	source data length register
0x8	GZIP_OUT_SIZE	output data length register
0xc	DECOMP_STAT	decompress status register

15.5 Register Description

DECOMP_START DECOMP_START is a register used to start decompression controller
Offset Address : 0x0 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
1	backpress	0x1	RW	0x1: When the input buffer is not ready, back pressure the axi bus.
0	decomp_start	0x0	WO	0x1: Start decompression controller.

GZIP_SRC_SIZE GZIP_SRC_SIZE is a decompress source data length register

Offset Address : 0x4 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
31	decomp_ctrl_en	0x0	RW	0x1: Decompression controller enable.
30:0	dma_in_size	0x0	RW	Decompress the input data length of the controller

GZIP_OUT_SIZE GZIP_OUT_SIZE is a decompress output data length register

Offset Address : 0x8 Total Reset Value: 0x0

bits	Name	Reset	Access	Descriptions
31:0	dma_out_size	0x0	RW	Decompress the output data length of the controller

DECOMP_STAT DECOMP_STAT is a decompress status register

Offset Address : 0xc Total Reset Value: 0x0

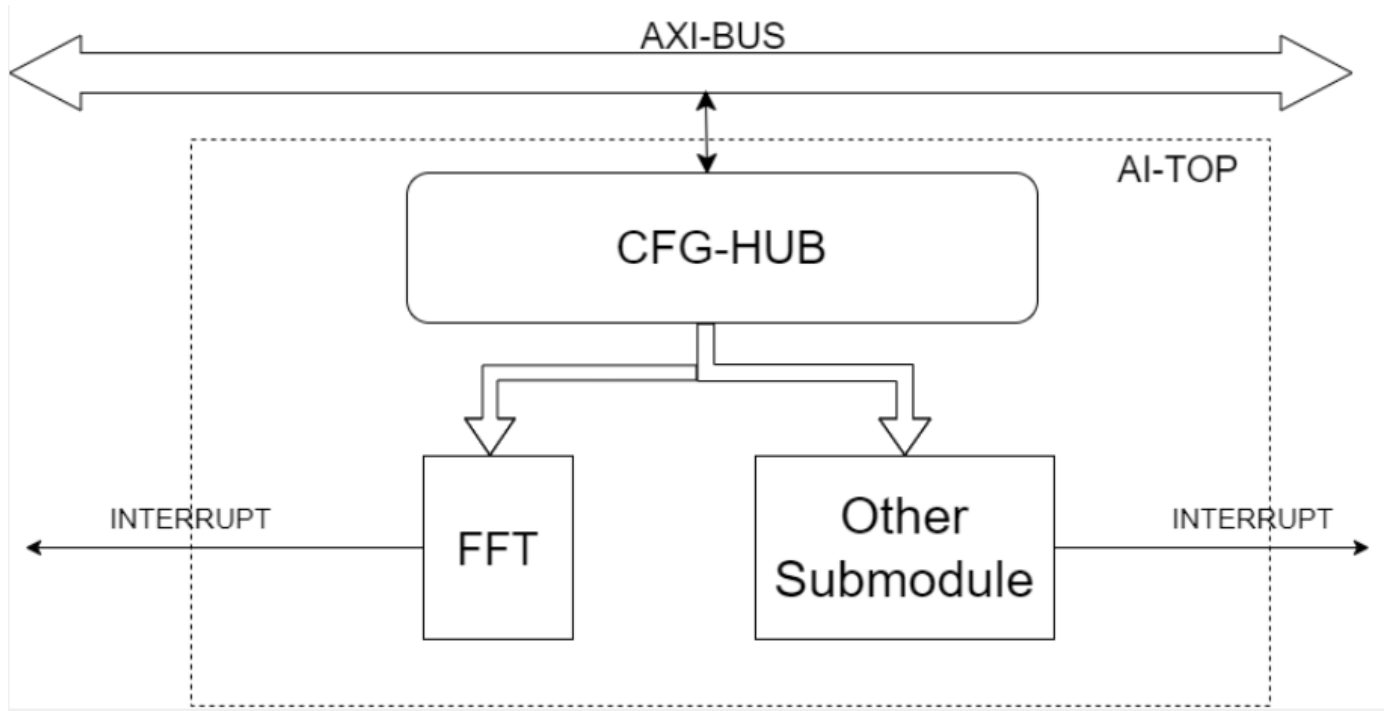
bits	Name	Reset	Access	Descriptions
10	crc_ok	0x0	RO	Crc status flag
3:0	gzip_state	0x0	RO	gzip State machine value.

16 FFT Engine

16.1 Overview

The FFT module is mainly used for hardware-accelerated FFT and IFFT calculations. The external interface is the standard axi4 slave interface and interrupt output. In order to save the number of NOC bus interfaces, FFT and ai_top multiplex the axi slave interface through the 1master to 2slave bridge.

Figure 16-1-1 FFT block diagram



16.2 Features

- Support maximum 4096 point fft, ifft calculation.
- Support int16 calculation accuracy, the real and imaginary parts of input and output are both in int16 format.
- Support the standard axi4 slave interface, which is used for parameter configuration and data transfer.
- The input supports RIRI..., RRRR.... (singal real part), RRRR...IIII... format arrangement, and the output supports RIRI..., RRRR...IIII... format arrangement.
- It adopts the calculation method of base 2-time decimation, and there is only one butterfly operator inside.
- A single clock domain design is adopted, and the bus clock is also used as the operation clock.
- The calculation time of 4096-point fft/ifft is controlled within 1ms, including the total overhead of data movement, calculation, and interrupt interaction.
- Support interrupt mask, original interrupt query.

16.3 Function Description

- 1 AXI4 Slave interface, the interface width is 128bit, and only the lower 64bit is used for register configuration and data transmission in FFT.
- FFT uses in-situ computing features internally, including 16KB storage for storing data input and output and intermediate calculation results.
- FFT内部支持64、128、256、512、1024、2048、4096点fft、ifft计算.
- Taking 500MHz as the internal processing clock to estimate, the 4096-point FFT calculation time is 0.2ms, considering the software configuration and SOC data transfer, the total time is less than 1ms.
- The internal FFT can be self-cleared after a single operation, and supports system soft reset.
- Including abnormal reading and writing, calculation timeout interrupt reporting function.
- Support the Debug function by reading the internal status register.

NOTE

- Abnormal writing status includes multiple data sending, writing in non-writing data status, computing status update configuration register, etc.
- Abnormal read status includes multi-read data and non-read data status read.
- The timeout interrupt can be configured with a minimum timing value of 2.

16.4 Operating Mode

Typical Workflow

The CPU configures the configuration register with offset address 0x0 through the bus, and then configures the enable register with offset address 0x10. At this time, the FFT module is enabled. After the FFT module is enabled, dma starts to move the original calculation to the offset address 0x40. Data, the FFT module receives the original data from the master according to the configuration information, and writes it into the internal storage according to the reverse order rule. After all the data is received, it starts to calculate. The FFT will sequentially take out the corresponding data from the storage and send them to the butterfly algorithm. Sub, and write the calculation result back to the storage, so loop until the calculation is completed, after the calculation is completed, report an interrupt. DMA starts to read the calculation result from the offset address 0x40.

Exception Handling Process

When an abnormal situation (mainly abnormal reading and writing and timeout interrupt) occurs and the fft module needs to be reset, the CPU first configures the 0x10 register and disables the FFT module. At this time, the FFT module stops receiving bus data and will not return data to the bus, and then the CPU soft Reset the DMA to ensure that the residual data is no longer transmitted, and then soft reset the FFT module to complete the entire reset process.

Pattern Configuration Directory

FFT points, FFT mode, FFT input mode, FFT output mode, FFT interrupt mask, FFT offset enable, GATING enable, timeout configuration. [Register](#)

16.5 Register Summary

CHART 4-3-1 FFT Register List (base address : 0x40_0800)

Offset	Name	Description	Page
0x00	CONFIG_MODE_REG	Mode Parameter Configuration Register	Detail
0x10	FFT_ENABLE_REG	FFT enable register	Detail
0x20	FFT_INTR_CLEAR_REG	FFT Interrupt Clear Register	Detail
0x30	FFT_INTR_ORIG_REG	FFT raw interrupt read register	
0x40	FFT_INTER_DATA_REG	FFT data access register	
0x50	FFT_INTR_NUM_REG	FFT internal interrupt number register	Detail
0x60	FFT_DEBUG_REG0	FFT internal DEBUG register 0	Detail
0x70	FFT_DEBUG_REG1	FFT internal DEBUG register 1	Detail

16.6 Register Description

CONFIG_MODE_REG

Mode Configuration Register Indicates the relevant parameters of the FFT operation.

Offset Address: 0x400800 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[2:0]	RW	FFT_POINT	0:64; 1:128; 2:256; 3:512 4:1024; 5:2048; 6:4096;	0x0
[3]	RW	FFT_MODE	0x0:FFT 0x1:IFFT	0x0
[5:4]	RW	FFT_IN_MODE	0x0:RIRI.... 0x1:RRRR....(pure real part) 0x2:RRRR....IIII....	0x0
[6]	RW	FFT_OUT_MODE	0x0:RIRI.... 0x1:RRRR....IIII....	0x0
[7]	RW	FFT_INTR_MASK	0x0:no break mask 0x1:break mask	0x0
[6]	RW	FFT_OUT_MODE	0x0:RIRI.... 0x1:RRRR....IIII....	0x0
[19 : 8]	RW	FFT_SHIFT_EN	[11]Stage 12 right shift enable...[0]First stage right shift enable In FFT/IFFT mode, FFT_SHIFT_EN is configured as 0x555 by default, which corresponds to binary0101_0101_0101;	0x0
[20]	RW	FFT_DISABLE_GATING	0x1:Unenable FFT CLOCK GATING	0x0
[31 : 21]	-	Reserved	Reserved	0x0
[63:32]	RW	FFT_TH_TIMEOUT	Indicates the FFT module calculation timeout threshold after FFT is enabled; writing 0 to this value means that there is no FFT timeout interrupt reporting function The value configured when it is not 0. 1/fc indicates the time threshold	0x0
[127 : 64]	-	Reserved	Reserved	0x0

FFT_ENABLE_REG FFT enable register

Offset Address: 0x400810 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[0]	RW	FFT_ENABLE	0x1:ENABLE FFT	0x0
[127 : 1]	-	Reserved	Reserved	0x0

FFT_INTR_CLEAR_REG FFT Interrupt Clear Register

Offset Address: 0x400820 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[0]	RW	FFT_INTR_CLEAR	0x1:Clear FFT interrupt number	0x0
[127:1]	-	Reserved	Reserved	0x0

FFT_INTR_ORIG_REG Raw interrupt register

Offset Address: 0x400830 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[0]	RO	FFT_INTR_ORIG	Read FFT raw interrupt	0x0

FFT_INTER_DATA_REG FFT data access register

Offset Address: 0x400840 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[63:0]	RW	FFT_INTER_DATA	FFT input and output data	0x0
[127:64]	-	Reserved	Reserved	0x0

FFT_INTR_NUM_REG FFT internal interrupt number register Record interrupts for CPU to read

Offset Address: 0x400850 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[1:0]	RO	FFT_INTR_NUM	Interrupt number register, 0 means normal interrupt, 1 means abnormal writing during calculation, 2 means abnormal reading during calculation, 3 means fft operation timeout	0x0
[127:2]	-	Reserved	Reserved	0x0

FFT_DEBUG_REG0 DEBUG registers

Offset Address: 0x400860 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[1:0]	RO	i_fft_state	FFT internal state machine state	0x0
[14:2]	RO	rx_wr_cnt	FFT internal receive data counter	
0x0				
[19:15]	RO	rx_data_buffer_size_real	FFT internally receives the data count in the data buffer	0x0
[24:20]	RO	tx_data_buffer_byte_size_real	Data count in FFT send data buffer	
0x0				

Bits	Access	Name	Description	Reset
[27:25]	RO	wt_data_buffer_size_real	FFT writes data buffer data count to storage	
0x0				
[30:28]	RO	rd_data_buffer_size_real	FFT reads the data count in the data buffer from the storage	
0x0				
[33:31]	RO	rd_tf_buffer_size_real	FFT reads the data count in the twiddle factor buffer	
0x0				
[37:34]	RO	stage_cnt_tf	Twiddle factor stage counter	
0x0				
[41:38]	RO	stage_cnt_wt	stage flag when writing to storage	
0x0				
[45:42]	RO	stage_cnt_rd	stage flag when reading storage	
0x0				
[56:46]	RO	butter_cnt_tf	Twiddle Factor Butterfly Counter	
0x0				
[63:57]	RO	butter_cnt_wt	Butterfly counter (lower 8 bits) when writing to storage	
0x0				
[127:64]	-	Reserved	Reserved	0x0

FFT_DEBUG_REG1 DEBUG Registers

Offset Address: 0x400870 Total Reset Value: 0x0

Bits	Access	Name	Description	Reset
[3:0]	RO	butter_cnt_wt	Butterfly counter when writing storage (high 4bit)	0x0
[14:4]	RO	butter_cnt_rd	Butterfly counter when reading store	
0x0				
[26:15]	RO	tx_rd_cnt	Counter to read data output from storage	
0x0				
[38:27]	RO	tx_send_cnt	Send data counter to the bus	
0x0				
[47 : 39]	RO	wreq_waddr	FFT module input write request [0], write address [8:1]	0x0
[56:48]	RO	rreq_raddr	FFT module input read request [0], read address [8:1]	0x0
[127:57]	-	Reserved	Reserved	0x0

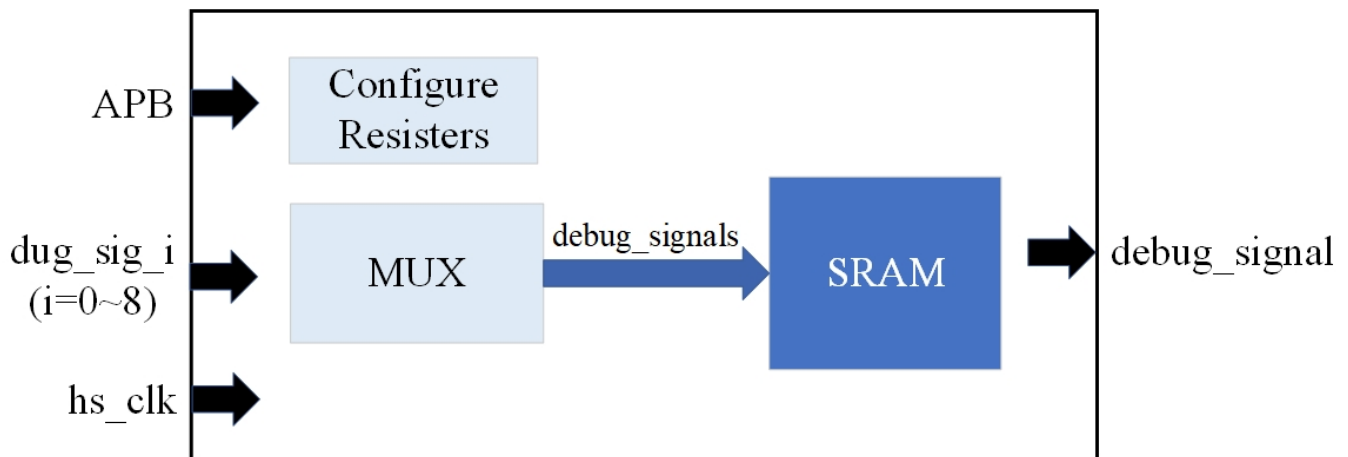
17 Debug Interface

17.1 Overview

HDI(Hardware Debug Interface) is used to debug the key control signals (such as clock, reset, interrupt, state machine) and data signals in SoC, so as to improve the controllability and observability of the SoC internal modules.

17.2 Block Diagram

Figure 17-2-1 HDI Block Diagram The block diagram of HDI is shown in Figure 17-2-1.



17.3 FEATURE

- Interface: APB3.0
- Bus data width: 32
- Debug the internal signal of SoC through the Pin
- Debug the signal stored in the memory through the CPU
- Clock: up to 400M

17.4 Function Description

There are two debug mode to debug signal on-chip, which are Pin debug mode and CPU debug mode.

17.4.1 Pin debug mode

In the Pin debug mode, It must select one of the input signals firstly. Then according to the configuration, several bits of the selected signal will output from the mapped idle Pin, which not work in the function mode.

The HDI will be started to work after the register debug_en configured from 1'b0 to 1'b1. At this time, the output data from the Pin is valid.

17.4.2 CPU debug mode

In the CPU debug mode, It must select one of the input signals firstly. When the trigger conditions are met, the selected signal will be written into the memory. It will not stop to write into the memory until the memory is full. Finally, the data stored in the memory is sent to the CPU through the APB bus.

The HDI will be started to work after the register debug_en configured from 1'b0 to 1'b1. After the memory is read empty, register debug_en is configured from 1'b1 to 1'b0.

17.5 Register Description

17.5.1 Register Summary

Table 17-1 describes HDI registers.

Table 17-1 Summary of HDI registers (base address: 0x9110_7800)

Offset Address	Register	DESCRIPTIONS
0x00	pin0_pin2bit_sel	Debug signal mapped with pin0~pin3 Configuration Register.
0x04	pin1_pin2bit_sel	Debug signal mapped with pin4~pin7 Configuration Register.
0x08	pin2_pin2bit_sel	Debug signal mapped with pin8~pin11 Configuration Register.
0x0C	pin3_pin2bit_sel	Debug signal mapped with pin12~pin15 Configuration Register.
0x10	pin4_pin2bit_sel	Debug signal mapped with pin16~pin19 Configuration Register.
0x14	pin5_pin2bit_sel	Debug signal mapped with pin20~pin23 Configuration Register.
0x18	pin6_pin2bit_sel	Debug signal mapped with pin24~pin27 Configuration Register.
0x1C	pin7_pin2bit_sel	Debug signal mapped with pin28~pin31 Configuration Register.
0x20	bit0_trig_type_sel	One of the trigger type (5 types) mapped with bit0.
0x24	bit1_trig_type_sel	One of the trigger type (5 types) mapped with bit1.
0x28	bit2_trig_type_sel	One of the trigger type (5 types) mapped with bit2.
0x2C	bit3_trig_type_sel	One of the trigger type (5 types) mapped with bit3.
0x30	bit4_trig_type_sel	One of the trigger type (5 types) mapped with bit4.
0x34	bit5_trig_type_sel	One of the trigger type (5 types) mapped with bit5.
0x38	bit6_trig_type_sel	One of the trigger type (5 types) mapped with bit6.
0x3C	bit7_trig_type_sel	One of the trigger type (5 types) mapped with bit7.
0x40	bit8_trig_type_sel	One of the trigger type (5 types) mapped with bit8.
0x44	bit9_trig_type_sel	One of the trigger type (5 types) mapped with bit9.
0x48	bit10_trig_type_sel	One of the trigger type (5 types) mapped with bit10.
0x4C	bit11_trig_type_sel	One of the trigger type (5 types) mapped with bit11.
0x50	bit12_trig_type_sel	One of the trigger type (5 types) mapped with bit12.
0x54	bit13_trig_type_sel	One of the trigger type (5 types) mapped with bit13.
0x58	bit14_trig_type_sel	One of the trigger type (5 types) mapped with bit14.
0x5C	bit15_trig_type_sel	One of the trigger type (5 types) mapped with bit15.
0x60	bit16_trig_type_sel	One of the trigger type (5 types) mapped with bit16.
0x64	bit17_trig_type_sel	One of the trigger type (5 types) mapped with bit17.

Offset Address	Register	DESCRIPTIONS
0x68	bit18_trig_type_sel	One of the trigger type (5 types) mapped with bit18.
0x6C	bit19_trig_type_sel	One of the trigger type (5 types) mapped with bit19.
0x70	bit20_trig_type_sel	One of the trigger type (5 types) mapped with bit20.
0x74	bit21_trig_type_sel	One of the trigger type (5 types) mapped with bit21.
0x78	bit22_trig_type_sel	One of the trigger type (5 types) mapped with bit22.
0x7C	bit23_trig_type_sel	One of the trigger type (5 types) mapped with bit23.
0x80	bit24_trig_type_sel	One of the trigger type (5 types) mapped with bit24.
0x84	bit25_trig_type_sel	One of the trigger type (5 types) mapped with bit25.
0x88	bit26_trig_type_sel	One of the trigger type (5 types) mapped with bit26.
0x8C	bit27_trig_type_sel	One of the trigger type (5 types) mapped with bit27.
0x90	bit28_trig_type_sel	One of the trigger type (5 types) mapped with bit28.
0x94	bit29_trig_type_sel	One of the trigger type (5 types) mapped with bit29.
0x98	bit30_trig_type_sel	One of the trigger type (5 types) mapped with bit30.
0x9C	bit31_trig_type_sel	One of the trigger type (5 types) mapped with bit31.
0xA0	debug_cfg	Debug Configuration Register.
0xA4	dug_sig_sel_cfg	Select one of the input signals to debug.
0xA8	sram	256X32bit SRAM.

17.5.2 Detail Register Description

pin0_pin2bit_sel

Debug signal mapped with pin0~pin3 Configuration Register.

Offset Address: 0x00 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	-
[2:0]	RW	pin2bit_sel	Select one group of the debug bits mapped with the group of the pins. 000: bit0~bit3; 001: bit4~bit7; ... 111: bit28~bit31	0x0

pin1_pin2bit_sel

Debug signal mapped with pin4~pin7 Configuration Register.

Offset Address: 0x04 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	-

Bits	Access	Name	Description	Reset
[2:0]	RW	pin2bit_sel	Select one group of the debug bits mapped with the group of the pins. 000: bit0~bit3; 001: bit4~bit7; ... 111: bit28~bit31	0x0

pin2_pin2bit_sel

Debug signal mapped with pin8~pin11 Configuration Register.

Offset Address: 0x08 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	-
[2:0]	RW	pin2bit_sel	Select one group of the debug bits mapped with the group of the pins. 000: bit0~bit3; 001: bit4~bit7; ... 111: bit28~bit31	0x0

pin3_pin2bit_sel

Debug signal mapped with pin12~pin15 Configuration Register.

Offset Address: 0x0C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	-
[2:0]	RW	pin2bit_sel	Select one group of the debug bits mapped with the group of the pins. 000: bit0~bit3; 001: bit4~bit7; ... 111: bit28~bit31	0x0

pin4_pin2bit_sel

Debug signal mapped with pin16~pin19 Configuration Register.

Offset Address: 0x10 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	-
[2:0]	RW	pin2bit_sel	Select one group of the debug bits mapped with the group of the pins. 000: bit0~bit3; 001: bit4~bit7; ... 111: bit28~bit31	0x0

pin5_pin2bit_sel

Debug signal mapped with pin20~pin23 Configuration Register.

Offset Address: 0x14 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	-
[2:0]	RW	pin2bit_sel	Select one group of the debug bits mapped with the group of the pins. 000: bit0~bit3; 001: bit4~bit7; ... 111: bit28~bit31	0x0

pin6_pin2bit_sel

Debug signal mapped with pin24~pin27 Configuration Register.

Offset Address: 0x18 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	-
[2:0]	RW	pin2bit_sel	Select one group of the debug bits mapped with the group of the pins. 000: bit0~bit3; 001: bit4~bit7; ... 111: bit28~bit31	0x0

pin7_pin2bit_sel

Debug signal mapped with pin28~pin31 Configuration Register.

Offset Address: 0x1C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	-
[2:0]	RW	pin2bit_sel	Select one group of the debug bits mapped with the group of the pins. 000: bit0~bit3; 001: bit4~bit7; ... 111: bit28~bit31	0x0

bit0_trig_type_sel

One of the trigger type (5 types) mapped with bit0 Configuration Register.

Offset Address: 0x20 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	-

Bits	Access	Name	Description	Reset
[2:0]	RW	trig_type_sel	Select one of the trigger type (5 types) mapped with bit0. 000: don't care 001: posedge trigger 010: negedge trigger 011: high level trigger 100: low level trigger Others: Reserved	0x0

bit1_trig_type_sel

One of the trigger type (5 types) mapped with bit1 Configuration Register.

Offset Address: 0x24 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	-
[2:0]	RW	trig_type_sel	Select one of the trigger type (5 types) mapped with bit1. 000: don't care 001: posedge trigger 010: negedge trigger 011: high level trigger 100: low level trigger Others: Reserved	0x0

bit2_trig_type_sel

One of the trigger type (5 types) mapped with bit2 Configuration Register.

Offset Address: 0x28 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	-
[2:0]	RW	trig_type_sel	Select one of the trigger type (5 types) mapped with bit2. 000: don't care 001: posedge trigger 010: negedge trigger 011: high level trigger 100: low level trigger Others: Reserved	0x0

bit3_trig_type_sel

One of the trigger type (5 types) mapped with bit3 Configuration Register.

Offset Address: 0x2C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	-

Bits	Access	Name	Description	Reset
[2:0]	RW	trig_type_sel	Select one of the trigger type (5 types) mapped with bit3. 000: don't care 001: posedge trigger 010: negedge trigger 011: high level trigger 100: low level trigger Others: Reserved	0x0

bit4_trig_type_sel

One of the trigger type (5 types) mapped with bit4 Configuration Register.

Offset Address: 0x30 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	-
[2:0]	RW	trig_type_sel	Select one of the trigger type (5 types) mapped with bit4. 000: don't care 001: posedge trigger 010: negedge trigger 011: high level trigger 100: low level trigger Others: Reserved	0x0

bit5_trig_type_sel

One of the trigger type (5 types) mapped with bit5 Configuration Register.

Offset Address: 0x34 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	-
[2:0]	RW	trig_type_sel	Select one of the trigger type (5 types) mapped with bit5. 000: don't care 001: posedge trigger 010: negedge trigger 011: high level trigger 100: low level trigger Others: Reserved	0x0

bit6_trig_type_sel

One of the trigger type (5 types) mapped with bit6 Configuration Register.

Offset Address: 0x38 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	-

Bits	Access	Name	Description	Reset
[2:0]	RW	trig_type_sel	Select one of the trigger type (5 types) mapped with bit6. 000: don't care 001: posedge trigger 010: negedge trigger 011: high level trigger 100: low level trigger Others: Reserved	0x0

bit7_trig_type_sel

One of the trigger type (5 types) mapped with bit7 Configuration Register.

Offset Address: 0x3C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	-
[2:0]	RW	trig_type_sel	Select one of the trigger type (5 types) mapped with bit7. 000: don't care 001: posedge trigger 010: negedge trigger 011: high level trigger 100: low level trigger Others: Reserved	0x0

bit8_trig_type_sel

One of the trigger type (5 types) mapped with bit8 Configuration Register.

Offset Address: 0x40 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	-
[2:0]	RW	trig_type_sel	Select one of the trigger type (5 types) mapped with bit8. 000: don't care 001: posedge trigger 010: negedge trigger 011: high level trigger 100: low level trigger Others: Reserved	0x0

bit9_trig_type_sel

One of the trigger type (5 types) mapped with bit9 Configuration Register.

Offset Address: 0x44 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	-

Bits	Access	Name	Description	Reset
[2:0]	RW	trig_type_sel	Select one of the trigger type (5 types) mapped with bit9. 000: don't care 001: posedge trigger 010: negedge trigger 011: high level trigger 100: low level trigger Others: Reserved	0x0

bit10_trig_type_sel

One of the trigger type (5 types) mapped with bit10 Configuration Register.

Offset Address: 0x48 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	-
[2:0]	RW	trig_type_sel	Select one of the trigger type (5 types) mapped with bit10. 000: don't care 001: posedge trigger 010: negedge trigger 011: high level trigger 100: low level trigger Others: Reserved	0x0

bit11_trig_type_sel

One of the trigger type (5 types) mapped with bit11 Configuration Register.

Offset Address: 0x4C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	-
[2:0]	RW	trig_type_sel	Select one of the trigger type (5 types) mapped with bit12. 000: don't care 001: posedge trigger 010: negedge trigger 011: high level trigger 100: low level trigger Others: Reserved	0x0

bit12_trig_type_sel

One of the trigger type (5 types) mapped with bit12 Configuration Register.

Offset Address: 0x50 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	-

Bits	Access	Name	Description	Reset
[2:0]	RW	trig_type_sel	Select one of the trigger type (5 types) mapped with bit12. 000: don't care 001: posedge trigger 010: negedge trigger 011: high level trigger 100: low level trigger Others: Reserved	0x0

bit13_trig_type_sel

One of the trigger type (5 types) mapped with bit13 Configuration Register.

Offset Address: 0x54 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	-
[2:0]	RW	trig_type_sel	Select one of the trigger type (5 types) mapped with bit13. 000: don't care 001: posedge trigger 010: negedge trigger 011: high level trigger 100: low level trigger Others: Reserved	0x0

bit14_trig_type_sel

One of the trigger type (5 types) mapped with bit14 Configuration Register.

Offset Address: 0x58 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	-
[2:0]	RW	trig_type_sel	Select one of the trigger type (5 types) mapped with bit14. 000: don't care 001: posedge trigger 010: negedge trigger 011: high level trigger 100: low level trigger Others: Reserved	0x0

bit15_trig_type_sel

One of the trigger type (5 types) mapped with bit15 Configuration Register.

Offset Address: 0x5C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	-

Bits	Access	Name	Description	Reset
[2:0]	RW	trig_type_sel	Select one of the trigger type (5 types) mapped with bit15. 000: don't care 001: posedge trigger 010: negedge trigger 011: high level trigger 100: low level trigger Others: Reserved	0x0

bit16_trig_type_sel

One of the trigger type (5 types) mapped with bit16 Configuration Register.

Offset Address: 0x60 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	-
[2:0]	RW	trig_type_sel	Select one of the trigger type (5 types) mapped with bit16. 000: don't care 001: posedge trigger 010: negedge trigger 011: high level trigger 100: low level trigger Others: Reserved	0x0

bit17_trig_type_sel

One of the trigger type (5 types) mapped with bit17 Configuration Register.

Offset Address: 0x64 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	-
[2:0]	RW	trig_type_sel	Select one of the trigger type (5 types) mapped with bit17. 000: don't care 001: posedge trigger 010: negedge trigger 011: high level trigger 100: low level trigger Others: Reserved	0x0

bit18_trig_type_sel

One of the trigger type (5 types) mapped with bit18 Configuration Register.

Offset Address: 0x68 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	-

Bits	Access	Name	Description	Reset
[2:0]	RW	trig_type_sel	Select one of the trigger type (5 types) mapped with bit18. 000: don't care 001: posedge trigger 010: negedge trigger 011: high level trigger 100: low level trigger Others: Reserved	0x0

bit19_trig_type_sel

One of the trigger type (5 types) mapped with bit19 Configuration Register.

Offset Address: 0x6C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	-
[2:0]	RW	trig_type_sel	Select one of the trigger type (5 types) mapped with bit19. 000: don't care 001: posedge trigger 010: negedge trigger 011: high level trigger 100: low level trigger Others: Reserved	0x0

bit20_trig_type_sel

One of the trigger type (5 types) mapped with bit20 Configuration Register.

Offset Address: 0x70 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	-
[2:0]	RW	trig_type_sel	Select one of the trigger type (5 types) mapped with bit20. 000: don't care 001: posedge trigger 010: negedge trigger 011: high level trigger 100: low level trigger Others: Reserved	0x0

bit21_trig_type_sel

One of the trigger type (5 types) mapped with bit21 Configuration Register.

Offset Address: 0x74 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	-

Bits	Access	Name	Description	Reset
[2:0]	RW	trig_type_sel	Select one of the trigger type (5 types) mapped with bit21. 000: don't care 001: posedge trigger 010: negedge trigger 011: high level trigger 100: low level trigger Others: Reserved	0x0

bit22_trig_type_sel

One of the trigger type (5 types) mapped with bit22 Configuration Register.

Offset Address: 0x78 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	-
[2:0]	RW	trig_type_sel	Select one of the trigger type (5 types) mapped with bit22. 000: don't care 001: posedge trigger 010: negedge trigger 011: high level trigger 100: low level trigger Others: Reserved	0x0

bit23_trig_type_sel

One of the trigger type (5 types) mapped with bit23 Configuration Register.

Offset Address: 0x7C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	-
[2:0]	RW	trig_type_sel	Select one of the trigger type (5 types) mapped with bit23. 000: don't care 001: posedge trigger 010: negedge trigger 011: high level trigger 100: low level trigger Others: Reserved	0x0

bit24_trig_type_sel

One of the trigger type (5 types) mapped with bit24 Configuration Register.

Offset Address: 0x80 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	-

Bits	Access	Name	Description	Reset
[2:0]	RW	trig_type_sel	Select one of the trigger type (5 types) mapped with bit24. 000: don't care 001: posedge trigger 010: negedge trigger 011: high level trigger 100: low level trigger Others: Reserved	0x0

bit25_trig_type_sel

One of the trigger type (5 types) mapped with bit25 Configuration Register.

Offset Address: 0x84 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	-
[2:0]	RW	trig_type_sel	Select one of the trigger type (5 types) mapped with bit25. 000: don't care 001: posedge trigger 010: negedge trigger 011: high level trigger 100: low level trigger Others: Reserved	0x0

bit26_trig_type_sel

One of the trigger type (5 types) mapped with bit26 Configuration Register.

Offset Address: 0x88 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	-
[2:0]	RW	trig_type_sel	Select one of the trigger type (5 types) mapped with bit26. 000: don't care 001: posedge trigger 010: negedge trigger 011: high level trigger 100: low level trigger Others: Reserved	0x0

bit27_trig_type_sel

One of the trigger type (5 types) mapped with bit27 Configuration Register.

Offset Address: 0x8C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	-

Bits	Access	Name	Description	Reset
[2:0]	RW	trig_type_sel	Select one of the trigger type (5 types) mapped with bit27. 000: don't care 001: posedge trigger 010: negedge trigger 011: high level trigger 100: low level trigger Others: Reserved	0x0

bit28_trig_type_sel

One of the trigger type (5 types) mapped with bit28 Configuration Register.

Offset Address: 0x90 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	-
[2:0]	RW	trig_type_sel	Select one of the trigger type (5 types) mapped with bit28. 000: don't care 001: posedge trigger 010: negedge trigger 011: high level trigger 100: low level trigger Others: Reserved	0x0

bit29_trig_type_sel

One of the trigger type (5 types) mapped with bit29 Configuration Register.

Offset Address: 0x94 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	-
[2:0]	RW	trig_type_sel	Select one of the trigger type (5 types) mapped with bit29. 000: don't care 001: posedge trigger 010: negedge trigger 011: high level trigger 100: low level trigger Others: Reserved	0x0

bit30_trig_type_sel

One of the trigger type (5 types) mapped with bit30 Configuration Register.

Offset Address: 0x98 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	-

Bits	Access	Name	Description	Reset
[2:0]	RW	trig_type_sel	Select one of the trigger type (5 types) mapped with bit30. 000: don't care 001: posedge trigger 010: negedge trigger 011: high level trigger 100: low level trigger Others: Reserved	0x0

bit31_trig_type_sel

One of the trigger type (5 types) mapped with bit31 Configuration Register.

Offset Address: 0x9C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	-
[2:0]	RW	trig_type_sel	Select one of the trigger type (5 types) mapped with bit31. 000: don't care 001: posedge trigger 010: negedge trigger 011: high level trigger 100: low level trigger Others: Reserved	0x0

debug_cfg

Debug Configuration Register.

Offset Address: 0xA0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	-
[2]	RW	triger_sel	Select trigger mode of and/or. 1: trigger with condition of and logic 0: trigger with condition of or logic(not support)	0x0
[1]	RW	mode_sel	the mode of observing debug signal. 0: via pin 1: output to CPU or SPI2AXI via sram	0x0
[0]	RW	debug_en	Debug enable,active high.	0x0

dug_sig_sel_cfg

Select one of the input signals to debug.

Offset Address: 0xA4 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	-

Bits	Access	Name	Description	Reset
[3:0]	RW	sig_sel	Select one of the all input signals to debug. 0000: dug_sgl_i[0] 0001: dug_sgl_i[1] ... 0111: dug_sgl_i[7] 1000: dug_sgl_i[8]	0x0

sram

256X32bit SRAM.

Offset Address: 0xA8 Total Reset Value: -

Bits	Access	Name	Description	Reset
[31:0]	-	reserved	Reserved	-

17.6 Interface Descriptions

IO pads of HDI are listed in the following table.

Table 17-2 HDI interface description

Module Pin	Direction	Pad Name
debug_signal[0]	O	PAD_IO_3
debug_signal[1]	O	PAD_IO_4
debug_signal[2]	O	PAD_IO_5
debug_signal[3]	O	PAD_IO_6
debug_signal[4]	O	PAD_IO_19
debug_signal[5]	O	PAD_IO_20
debug_signal[6]	O	PAD_IO_21
debug_signal[7]	O	PAD_IO_22
debug_signal[8]	O	PAD_IO_23
debug_signal[9]	O	PAD_IO_24
debug_signal[10]	O	PAD_IO_25
debug_signal[11]	O	PAD_IO_31
debug_signal[12]	O	PAD_IO_32
debug_signal[13]	O	PAD_IO_33
debug_signal[14]	O	PAD_IO_34
debug_signal[15]	O	PAD_IO_35
debug_signal[16]	O	PAD_IO_36
debug_signal[17]	O	PAD_IO_37

Module Pin	Direction	Pad Name
debug_signal[18]	O	PAD_IO_40
debug_signal[19]	O	PAD_IO_41
debug_signal[20]	O	PAD_IO_42
debug_signal[21]	O	PAD_IO_43
debug_signal[22]	O	PAD_IO_46
debug_signal[23]	O	PAD_IO_47
debug_signal[24]	O	PAD_IO_50
debug_signal[25]	O	PAD_IO_51
debug_signal[26]	O	PAD_IO_52
debug_signal[27]	O	PAD_IO_54
debug_signal[28]	O	PAD_IO_55
debug_signal[29]	O	PAD_IO_56
debug_signal[30]	O	PAD_IO_57
debug_signal[31]	O	PAD_IO_58

17.7 Application Notes

The sections describe how to debug the SoC internal signals using the HDI. There are two debug applications, Pin Debug and CPU Debug.

The mapping relationship between 9 groups of signals and 9 subsystems is as follows.

Table 17-3 mapping relationship

signal	subsystem
dug_sgl_i[0]	sys_ctrl
dug_sgl_i[1]	ddr_controller
dug_sgl_i[2]	lo_sys
dug_sgl_i[3]	disp_sys
dug_sgl_i[4]	hi_sys
dug_sgl_i[5]	shrm_sys
dug_sgl_i[6]	dpu_sys
dug_sgl_i[7]	kpu_sys
dug_sgl_i[8]	isp_sys

Table 17-4 sys_ctrl bits description

bits	signal
[3:0]	pll_test_clk

bits	signal
[4]	pll3_lock_o
[5]	pll2_lock_o
[6]	pll1_lock_o
[7]	pll0_lock_o
[8]	pll_boot_resetN_o
[10:9]	pll3_fsm_stat
[12:11]	pll2_fsm_stat
[14:13]	pll1_fsm_stat
[16:15]	pll0_fsm_stat
[20:17]	soc_ctl_fsm_o
[21]	offchip_rst_sync_24m
[31:22]	Reserved

Table 17-5 ddr_controller bits description

bits	signal
[6:0]	lpr_crdit_cnt
[17:13]	hpr_credit_cnt
[20:14]	wr_credit_cnt
[31:21]	Reserved

Table 17-6 lo_sys bits description

bits	signal
[0]	DI
[1]	DI_NEW
[2]	CI_NEW
[3]	DO
[4]	1'b0
[11:5]	REG_DiFCnt[6:0]
[12]	REG_DiFFull
[13]	REG_DiF_pop
[14]	get_vld
[15]	now_rd
[22:16]	REG_DoFCnt[6:0]
[24:23]	get_ecc[1:0]

bits	signal
[25]	fifo_pop
[26]	REG_DoF_push
[27]	now_wr
[29:28]	do_state[1:0]
[31:30]	di_state[1:0]

Table 17-7 disp_sys bits description

bits	signal
[7:0]	debug_out_gpu
[31:8]	Reserved

Table 17-8 hi_sys bits description(debug_bus0)

bits	signal
[0]	otg1_utmi_clk
[1]	otg1_IDDIG0
[2]	otg1_SUSPENDM0
[4:3]	otg1_OPMODE0
[5]	otg1_TERMSEL0
[7:6]	otg1_XCVRSEL0
[15:8]	otg1_DATAIN0
[16]	otg1_TXVALID0
[17]	otg0_utmi_clk
[18]	otg0_IDDIG0
[19]	otg0_SUSPENDM0
[21:20]	otg0_OPMODE0
[22]	otg0_TERMSEL0
[24:23]	otg0_XCVRSEL0
[31:25]	otg0_DATAIN0[6:0]

Table 17-9 hi_sys bits description(debug_bus1)

bits	signal
[0]	ahb1m7s_hwrite
[2:1]	ahb1m7s_htrans
[3]	ahb1m7s_hready_resp_s7
[4]	ahb1m7s_hready_resp_s6

bits	signal
[5]	ahb1m7s_hready_resp_s5
[6]	ahb1m7s_hready_resp_s4
[7]	ahb1m7s_hready_resp_s3
[8]	ahb1m7s_hready_resp_s2
[9]	ahb1m7s_hready_resp_s1
[10]	ahb1m7s_hsel_s7
[11]	ahb1m7s_hsel_s6
[12]	ahb1m7s_hsel_s5
[13]	ahb1m7s_hsel_s4
[14]	ahb1m7s_hsel_s3
[15]	ahb1m7s_hsel_s2
[16]	ahb1m7s_hsel_s1
[17]	eh2h_mhready
[18]	eh2h_mhwrite
[20:19]	eh2h_mhtrans
[21]	ahb1m2s_hready_resp_s1
[22]	hready_resp_m1
[23]	ahb1m2s_hready_resp_s2
[24]	ahb1m2s_hready_resp_s1
[25]	ahb1m2s_hwrite
[27:26]	ahb1m2s_htrans
[28]	hwrite_m1
[30:29]	htrans_m1

Table 17-10 hi_sys bits description(debug_bus2)

bits	signal
[0]	rready_s1
[1]	rlast_s1
[2]	rvalid_s1
[3]	arready_s1
[4]	arvalid_s1
[5]	bready_s1
[6]	bvalid_s1
[7]	wlast_s1

bits	signal
[8]	wready_s1
[9]	wvalid_s1
[10]	awready_s1
[11]	awvalid_s1
[12]	rready_m2
[13]	rlast_m2
[14]	rvalid_m2
[15]	arready_m2
[16]	arvalid_m2
[17]	bready_m2
[18]	bvalid_m2
[19]	wlast_m2
[20]	wready_m2
[21]	wvalid_m2
[22]	awready_m2
[23]	awvalid_m2
[24]	rready_m1
[25]	rlast_m1
[26]	rvalid_m1
[27]	arready_m1
[28]	arvalid_m1
[29]	bready_m1
[30]	bvalid_m1
[31]	wlast_m1

Table 17-11 hi_sys bits description(debug_bus3)

bits	signal
[0]	rready_s2
[1]	rlast_s2
[2]	rvalid_s2
[3]	arready_s2
[4]	arvalid_s2
[5]	bready_s2
[6]	bvalid_s2

bits	signal
[7]	wlast_s2
[8]	wready_s2
[9]	wvalid_s2
[10]	awready_s2
[11]	awvalid_s2
[12]	rready_m4
[13]	rlast_m4
[14]	rvalid_m4
[15]	arready_m4
[16]	arvalid_m4
[17]	bready_m4
[18]	bvalid_m4
[19]	wlast_m4
[20]	wready_m4
[21]	wvalid_m4
[22]	awready_m4
[23]	awvalid_m4
[24]	rready_m3
[25]	rlast_m3
[26]	rvalid_m3
[27]	arready_m3
[28]	arvalid_m3
[29]	bready_m3
[30]	bvalid_m3
[31]	wlast_m3

Table 17-12 hi_sys bits description(debug_bus4)

bits	signal
[0]	hready_usb_ahb2
[1]	hmastlock_usb_ahb2
[2]	hwrite_usb_ahb2
[4:3]	htrans_usb_ahb2
[5]	hsel_usb_ahb2
[6]	hwrite_l2

bits	signal
[8:7]	htrans_l2
[9]	hgrant_usbm2
[10]	otg1_m_hbusreq
[11]	hwrite_l1
[13:12]	htrans_l1
[14]	hgrant_usbm1
[15]	otg0_m_hbusreq

Table 17-13 hi_sys bits description(debug_bus5)

bits	signal
[0]	rready_ssi0
[1]	rlast_ssi0
[2]	rvalid_ssi0
[3]	arready_ssi0
[4]	arvalid_ssi0
[5]	bready_ssi0
[6]	bvalid_ssi0
[7]	wlast_ssi0
[8]	wready_ssi0
[9]	wvalid_ssi0
[10]	awready_ssi0
[11]	awvalid_ssi0
[12]	hready_resp_usb
[13]	hwrite_usb
[14]	htrans_usb
[15]	hready_resp_usb_ahb2
[16]	rready_usb
[17]	rlast_usb
[18]	rvalid_usb
[19]	arready_usb
[20]	arvalid_usb
[21]	bready_usb
[22]	bvalid_usb
[23]	wlast_usb

bits	signal
[24]	wready_usb
[25]	wvalid_usb
[26]	awready_usb
[27]	awvalid_usb

Table 17-14 shrm_sys bits description

bits	signal
[0]	peri_dev_data_req_vec[20]
[1]	peri_dev_ch_ack_vec[20]
[2]	peri_dev_ch_ack_vec[21]
[3]	peri_dev_ch_ack_vec[21]
[4]	peri_dev_data_req_vec[30]
[5]	peri_dev_ch_ack_vec[30]
[6]	peri_dev_data_req_vec[22]
[7]	peri_dev_ch_ack_vec[22]
[8]	peri_dev_data_req_vec[23]
[9]	peri_dev_ch_ack_vec[23]
[10]	peri_dev_data_req_vec[31]
[11]	peri_dev_ch_ack_vec[31]
[12]	decomp_clk_en_reg
[13]	kpu_clk_en_reg;
[14]	shrm_axis_clk_en_reg
[15]	decomp_read_req
[16]	decomp_read_ack
[17]	decomp_write_req
[18]	decomp_write_ack
[19]	axi0_awvalid
[20]	axi0_awready
[21]	axi0_wvalid
[22]	axi0_wready
[23]	decomp_rbuf0_full
[24]	decomp_rbuf1_full
[25]	gzip_state[0]
[26]	gzip_state[1]

bits	signal
[27]	gzip_state[2]
[28]	gzip_state[3]
[29]	ch1_lli_state[0]
[30]	ch1_lli_state[1]
[31]	ch1_lli_state[2]

Table 17-15 dpu_sys bits description (gb_dpu_debug_sel == 2'b0 (LCN & QLT))

bits	signal
[0]	dbg_tmpl_2_top_transfer
[1]	dbg_top_2_cnt_transfer
[2]	dbg_qlt_2_tx_transfer
[15:3]	13'b0
[16]	dbg_trans_2_win_transfer
[17]	dbg_lcn_2_sad_transfer
[31:18]	14'b0

Table 17-16 dpu_sys bits description (gb_dpu_debug_sel == 2'b1 (SAD))

bits	signal
[0]	w_debug_sad_pe_psum_ready
[1]	w_debug_ref_buf_calc_pp_flag
[2]	w_debug_ref_buf_init_pp_flag
[3]	w_debug_ref_fetch_busy
[14:4]	w_debug_obj_y
[25:15]	w_debug_obj_x
[26]	w_debug_obj_fetch_busy
[27]	w_debug_obj_calc_para_busy
[28]	w_debug_sha_ram_data_enough
[29]	w_debug_obj_ram_data_enough
[30]	w_debug_ref_ram_data_enough
[31]	w_debug_fine_flag

Table 17-17 dpu_sys bits description (gb_dpu_debug_sel == 2'b2 (Post))

bits	signal
[0]	i_data_valid
[1]	o_data_ready

bits	signal
[2]	o_ppeak_we
[3]	o_ppeak_ce
[6:4]	o_ppeak_addr[2:0]
[25:15]	w_debug_obj_x
[7]	o_pre_row_ppeak_we
[8]	o_pre_row_ppeak_ce
[11:9]	o_pre_row_ppeak_addr[2:0]
[12]	o_pnum_we
[13]	o_pnum_ce
[16:14]	o_pnum_addr[2:0]
[17]	o_plabel_we
[18]	o_plabel_ce
[21:19]	o_plabel_addr[2:0]
[25:23]	o_pre_mask_waddr[2:0]
[26]	i_disp_valid
[27]	o_disp_ready
[28]	i_premask_re
[31:29]	i_premask_raddr[2:0]

Table 17-18 dpu_sys bits description (gb_dpu_debug_sel == 2'b3 (Align))

bits	signal
[0]	gb_align_flag_align
[1]	gb_align_flag_median
[2]	gb_align_flag_upscale
[3]	gb_align_flag_depth_disp
[4]	gb_align_flag_msk_disp
[6:5]	align_st
[9:7]	align_ld_st
[17:10]	align_alu_c_blk_idx
[18]	align_buf2align_out_rdy
[19]	align_buf2align_out_ack
[20]	align_alu_sram_dst_wreq
[21]	align_alu_sram_dst_wack
[22]	align_m2_sram_dst_rreq

bits	signal
[23]	align_m2_sram_dst_rack
[24]	align_m2_out_rdy
[25]	align_m2_out_ack
[26]	align_depth_out_vld
[27]	align_depth_out_ack
[28]	align_resize_out_rdy
[29]	align_resize_out_ack
[30]	o_p3_host2biu_wvalid
[31]	i_p3_biu2host_wready

Table 17-19 kpu_sys bits description

bits	signal
[7:0]	w_tcu_debug_bus
[11:8]	dm_inst_working_cnt[3:0]
[15:12]	pu_inst_working_cnt[3:0]
[17:16]	pdp_rd_st
[19:18]	pdp_if_l1_fill_en
[22:20]	pdp_wr_st
[23]	1'b0
[27:24]	w_debug_bus_load
[31:28]	w_debug_bus_store

Table 17-20 isp_sys bits description

bits	signal
[1:0]	csi0_discard_state
[3:2]	csi1_discard_state
[5:4]	csi2_discard_state
[6]	sony_full_0
[9:7]	sony_line_sta_0
[11:10]	ipi_id_csi0
[14:12]	sony_pix_sta_0
[15]	flash_tri1_sta
[16]	flash_tri0_sta
[19:17]	sony_line_sta_334

bits	signal
[22:20]	sony_pix_sta_334
[23]	ipi_vsync_csi0_final
[24]	ipi_hsync_csi0_final
[25]	ipi_pixen_csi0_final
[26]	ipi_vsync_csi1_final
[27]	ipi_hsync_csi1_final
[28]	ipi_pixen_csi1_final
[29]	ipi_vsync_csi2_final
[30]	ipi_hsync_csi2_final
[31]	ipi_pixen_csi2_final

17.7.1 Pin Debug

Step 1: Initialise - Reset the preset and hs_rstn signal.

Step 2: Configure:

1. Select one of the input debug signals according to configure the Register [dug_sig_sel_cfg](#) bit[3:0]
2. Configure the register [debug_cfg](#) bit[1] to 0
3. Select the output bits mapped with the Pins according to configure the Register [pini_pin2bit_sel](#)(i=0,1,...,7) [0x00 ~ 0x1C](#)

Step 3: Start debug: Configure the register [debug_cfg](#) bit[0] to 1

17.7.2 CPU Debug

Step 1: Initialise - Reset the preset and hs_rstn signal.

Step 2: Configure:

- (1) Select one of the input debug signals according to configure the Register [dug_sig_sel_cfg](#) bit[3:0]
- (2) Configure the register [debug_cfg](#) bit[1] to 1
- (3) Configure the register [debug_cfg](#) bit[2] to 1
- (4) Select the trigger type mapped with the bit according to configure the Register [bitj_trig_type_sel](#) (j=0,1,2,...,30,31) [0x20 ~ 0x9C](#)

Step 3: Start debug: Configure the register [debug_cfg](#) bit[0] to 1

Step 4: Read the debug signal from SRAM via APB to CPU.