GM8136S/GM8135S

H.264 IP-CAM SoC

Data Sheet
Rev.: 0.2
Issue Date: October 2014

REVISION HISTORY

GM8136S/GM8135S Data Sheet

Date	Rev.	From	To
Sept. 2014	0.1	-	Original
Oct. 2014	0.2	-	• Updated Figure 2-3 and Figure 2-4 • Updated Table 2-2, Table 4-17, Table 4-24, Table 4-34, and Table 4-36

Printed in Taiwan 2014

Grain Media, Inc.
5F, No. 5, Li-Hsin Road III, Hsinchu Science Park, Hsinchu City, Taiwan 300, R.O.C.

Grain Media's home page can be found at:
http://www.grain-media.com

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

This chapter contains the following sections:

GM8136S/GM8135S is a highly integrated SoC design to provide a cost-effective and easy developing system for the IP-Cam applications. GM8136S/GM8135S includes a wide range of basic components, including ISP, H.264 encoder, MPEG4/JPEG codec, 2D graphic accelerator engine, video capture, display controller, DES/3DES controller, DMA controller, Ethernet RMII controller, USB 2.0 OTG, USB 1.1 OTG, device mode, and built-in 1Gb/512Mb DDRx to reduce the overall system cost.

1.1 Application Scenarios

Figure 1-1 shows the application diagram of GM8136S/GM8135S.

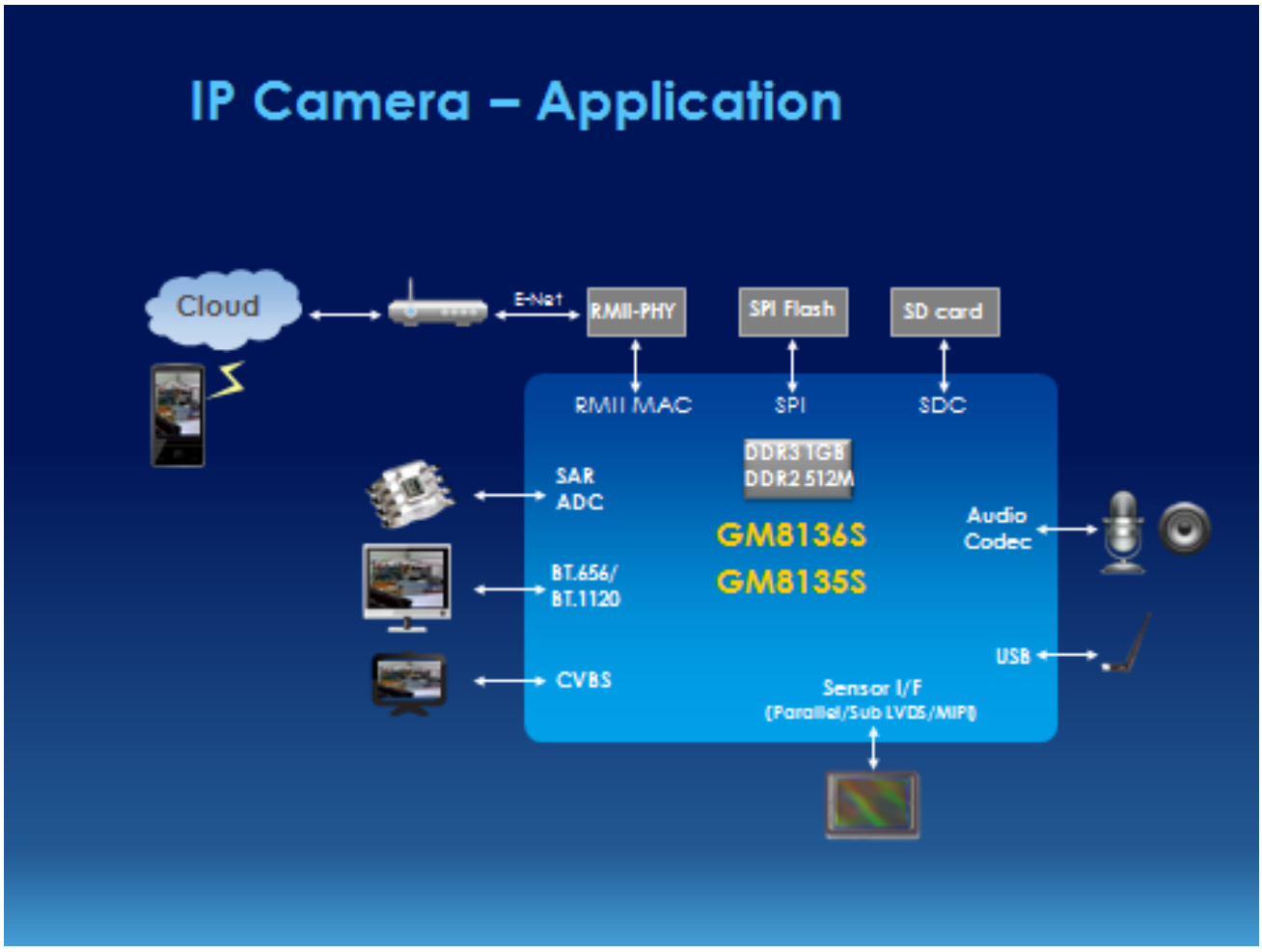


Figure 1-1. Application Diagram of GM8136S/GM8135S

1.2 Features

Embedded Processor

- ARM-based 32bit RISC
- 32KB instruction cache and 32KB data cache

Clock/PLL

- 30MHz main clock input
- 32.768KHz clock input (Optional)
- Programmable frequency core PLL for main clock and audio main clocks
- Programmable clock output for video decoder or CMOS sensor
- Programmable clock output for Ethernet PHY
- Frequency change control

Memory Interface

- Built-in 1Gb DDR3 or 512Mb DDR2 SDRAM (SIP)
- Supports SPI NOR/SPI NAND Flash boot

Two Secure Digital Card (SDC) Controllers

- Compliant with SD Host Controller Standard Specification, Version 3.0
- Compliant with SD Physical Layer Specification, Version 3.0

H.264 Encoder

- Supports H.264 BP/MP/HP encoding up to 2M pixels

MPEG4 Codec

- Compliant with MPEG4 simple profile L0 ~ L3 standards
- Supports encoding up to 720p
- Supports decoding up to 720p

JPEG Codec

- Compliant with JPEG baseline standard
- Supports image size up to 64k×64k

Image Processing

- 3D-deinterlace
  - Supports 3D de-interlace up to 1080i
  - Supports 60i to 30p and 60i to 60p 3D de-interlace function
- 3D-denoise
- Supports Intelligent Video Surveillance (IVS) engine
- Supports Image Signal Processor (ISP) with maximum of 2Mpixels

Video Scaler

- Maximum video input resolution of 1920x1080
- Supports linear scaling-down/up function
- Supports ARGB888 scaling function

Video Capture Interface

- ITU BT.656/ITU BT.1120/SDI 8bit input with maximum resolution of 1080p and maximum frequency at 148.5MHz
- CCIR601 input with maximum resolution of 1080p
- 8/10/12bit parallel Bayer RGB input with maximum resolution of 1080p
- Maximum 2-lane MIPI/subLVDS combo serial CMOS sensor interface with maximum resolution of 2Mpixels
- Supports size-down function
- Supports eight private region masks in each channel
- Supports font-based OSD in recording channel

Display Interface

- Supports ITU BT.1120 and ITU BT.656 digital output formats
- Integrates 27MHz DAC for composite output (NTSC/PAL)
- Parallel RGB565 output for LCD panel

Peripheral Components

- DMA controller
- USB 2.0 OTG controller
- USB 1.1 OTG controller device mode
- AES/DES/3DES
- Timer
- WatchDog timer
- Real time clock
- Interrupt controller
- PWM x8
- I^2C
- I^2S x2 (Only 1-set external pin)
- UART x3
- SPI

High-speed I/O interface

- USB OTG 2.0 x1
- OTG 1.1 x1 (Device only)
- Ethernet interface to support 10Mbps/100Mbps RMII

Embedded Analog Device

- USB 2.0 OTG PHY
- Serial Combo PHY (MIPI/subLVDS)
- Audio Codec
- SAR ADC with two channels
- Channel video DAC

Operating Voltage

- 1.1V ±5% for core power
- 1.5V ±5% for DDR3 SDRAM or 1.8V ±5% for DDR2 SDRAM
- 3.3V ±5% for inputs and outputs
- (1.8 ~ 3.3)V ±5% for Bayer

Operating Temperature

- From 0°C to 85°C (Case temperature)

Package

- GM8136S-QC-A/GM8135S-QC-A
  - ROHS, 196pin TFBGA
- GM8136S-BC-A/GM8135S-BC-A
  - ROHS, 128pin LQFP

1.3 Functional Block Diagram

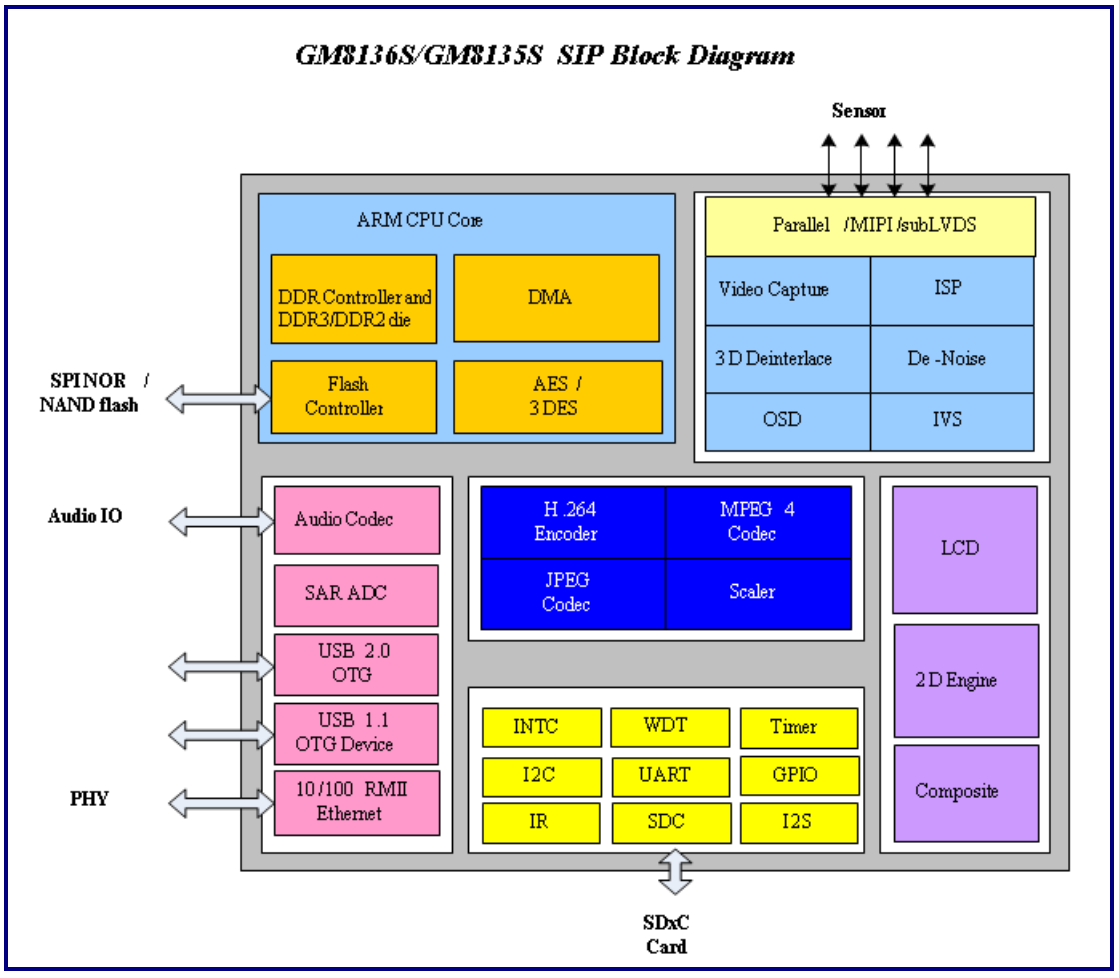


Figure 1-2. Functional Block Diagram

1.3.1 DDR3/DDR2 SDRAM Controller (DDRC)

The DDR3/DDR2 SDRAM controller supports DDR3/DDR2 SDRAMs and uses a DDR3/DDR2 burst length of eight to accelerate the read/write speeds. The features of DDRC include:

- 16bit data bus width
- Supports maximum 256Mbytes for each rank
- Provides two arbitration strategies for channel arbitrations:
  - Two-level round-robin arbitration with channel grant counts
  - Read/Write group arbitration with R/W commands grant counts
- Enters self-refresh mode by using hardware handshake signal or software configuration
- Automatically enters SDRAM power-down mode when controller idles for user-defined clock cycle
- Pre-fetches sequential read data for the AHB/AXI burst read command
- Supports data FIFO depth of 32 for each channel to enhance performance of read/write commands
- Supports MA tables corresponding to different sizes and types

1.3.2 Serial Peripheral Interface Controller (SPIC)

The Serial Peripheral Interface (SPI) controller provides an interface to access NAND/NOR Flash. The features of SPIC include:

- Supports SPI NAND/NOR Flash
- Supports DTR mode
- Supports DMA handshake mode and CPU PIO mode
- Supports SPI serial mode

1.3.3 H.264 Encoder (H264E)

The H264 high-profile encoder is a high-performance hardware video encoder of the MPEG4 AVC/JVT/H.264 video coding standard. The encoder is designed to compress a sequence of YCbCr 4:2:2 pictures into a compressed video bitstream. The features of H264 encoder include the following MPEG4 AVC/JVT/H.264 (ISO/IEC 14496-10) video coding standards:

- High profile level 4.1
- Supports I, B, and P frame encodings

- Supports CAVLC and CABAC entropy codings
- Includes de-interlace and de-noise preprocessor
- Configures horizontal and vertical search ranges as 16pixels ~ 256pixels with configuration steps of 8pixels
- CBR and VBR (Rate control by firmware)
- Supports user-defined quantization matrix
- Supports mono (4:0:0) encoding
- Supports 4x4 and 8x8 transform codings
- Supports 4x4 and 8x8 intra predictions

1.3.4 MPEG4/JPEG Engine

The MPEG4/JPEG codec includes the hardware engines to accelerate the computation-intensive tasks, such as the motion estimation, DCT/IDCT, quantization/inverse quantization, and motion compensation. By initializing the control registers of this codec, MPEG4/JPEG codec will automatically perform the motion estimation calculation task for 16x16 or 8x8 block. MPEG4/JPEG codec will also automatically perform the DCT/quantization, IDCT/inverse quantization, AC/DC prediction, zigzag scan, and VLC/VLD calculation task for a macro-block. Therefore, CPU loading can be released from the timing critical tasks in the video encoding process. The features of the MPEG4/JPEG engine include:

- Compliant with MPEG4 (ISO/IEC 14496-2) simple profile L0 ~ L3 standards with resolutions of subQCIF, QCIF, CIF, VGA, D1, and 720p with steps of 16 units
- Compliant with JPEG (ISO/IEC 10918-1) baseline standard
- Includes hardware engines for motion estimation/motion compensation, DCT/IDCT, quantization/inverse quantization, AC/DC prediction, and coding/decoding with variable length
- Uses local memory controller to control local memory shared by CPU, MPEG4/JPEG codec, and DMA master
- Uses DMA controller to control data transfers between system memory and local memory
- Supports automatic power-down mechanism to reduce power consumption
- Motion estimation search range: -16 ~ +15.5 (Optionally -32 ~ +31) with half-pixel accuracy
- Supports short video header (H.263 baseline)
- Supports H.263/MPEG/JPEG quantization methods

JPEG features

- Supports YUV 4:4:4, 4:2:2, and 4:2:0 formats
- Supports codec performance up to Mpixel/sec
- Supports image size up to 64k×64k
- Supports four user-defined Huffman tables (2 AC and 2 DC)
- Supports four programmable quantization tables
- Supports interleave and non-interleave scans
- Supports full-duplex operations (Such as video phone and video conference) by S/W to switch encoding and decoding tasks on same H/W

1.3.5 3D De-interlace and De-noise Filter (3DI/DN)

The de-interlace process is an important stage, which converts the ordinary TV interlaced sequences into progressive sequences for the sake of displaying on the progressive devices (Such as computers, LCD, plasma display, and projection TV). The de-noise process is important for enhancing the scene quality and reducing the encoded bitrate in a network. In a real-time IP camera system, the de-noise function is very useful for eliminating the Gauss-distribution noise, so that the encoder can reduce the effort of encoding the source scene. On the other hand, the encoded files will be smaller and the bit-rate transferred in network will be reduced. The 3D de-interlace and de-noise module adopts the pixel-based measurement to perform the intra-field and inter-field interpolations, depending on the motion state, which the pixel belongs to the motion or stationary. Its goal is to generate a high-quality progressive frame.

The features include:

- Link-list module with 256 commands buffer
- Supports frame size up to 1920x1080
- Multi-resolution Noise Reduction in spatial domain
- Luma de-noise strength learning in time domain
- Edge Enhancement on de-noise image
- Supports 60i to 30p and 60i to 60p functions
- Only supports format of packet YCbCr422

1.3.6 2D Graphic Accelerator Engine

The 2D graphic accelerator is a graphics subsystem for accelerating the 2D contents. The graphic accelerator is capable of targeting the multiple types of surfaces, seamlessly move raster images, accelerating drawing of lines and filled rectangles and performing alpha blending in numerous blending modes.

1.3.7 Image Signal Pocessor (ISP)

ISP is a powerful processor for the image signals, which captures the Bayer format data from sensor and outputs the YCbCr data to video capture after color reproduction, noise reduction, and image enhancement processes. Besides, it also provides 3A statistic and histogram for users to develop 3A algorithm.

ISP contains the following features:

- Supports maximum resolution of 1920x1080
- Sensor interface
  - Receive various types of incoming CMOS image data in 8/10/12bit Bayer raw data format
  - Receive 14bit command raw data input for WDR sensor
- Wide Dynamic Range (WDR)
- Noise reduction
- Lens shading correction
- Statistics collection for 3A
  - AE/AF: 9 windows, AWB: 1 window
- Histogram
  - RGB domain and YUV domain
- DMA support for quick and efficient
  - Parameters and register data moving
- Uses YCbCr 4:2:2 data output format

1.3.8 Intelligent Video Surveillance Engine (IVS)

The IVS engine is a hardware accelerator for generic intelligent video processing. The engine supports the color space conversion, SAD (Sum of Absolute Differences) computation, integral image generation, image convolution, morphological operations, and image histogram. A built-in DMA uses the AXI master interface for data accessing between the external and internal memories.

The features of IVS include:

- Supports YCbCr-to-HSI conversion
- Supports YCbCr-to-RGB conversion
- Supports de-interleaving 4:2:2 input image in two separate planes
- Supports generating integral image
- Supports computing SAD (Sum of Absolute Differences) of 1x1/3x3/5x5 block sizes
- Supports image threshold operations
- Supports any binary operation between two images
- Supports generating image histogram
- Supports 5x5 erosion/dilation
- Supports 3x3 opening/closing
- Supports 5x5 convolution
- Supports 5x5 gradient operators
- Supports cascaded weak classifiers

1.3.9 Video Scaler

The video scaler is used to change the size of a video image. The features of the video scaler include:

- Maximum video input resolution of 1920x1080
- Supports linear scaling-down/up function and two output resolutions per scan line
- Supports front-end image source cropping function
- Supports line de-noise
- Supports line false color suppression
- Supports line sharpness function for four output resolutions
- Supports single/packed-job mode
- Dedicates high-performance DMA
- Supports ARGB888/RGB565/RGB555 scaling and image format conversion

1.3.10 Video Capture

The video capture is in charge of capturing the video data from the ITU-R BT.656 or BT.1120 interface to a memory. The features of the video capture include:

- Maximum input capture resolution of 1920x1080
- Supports 54MHz/108MHz and 74.25MHz/148.5MHz byte/frame interleaving modes
- Supports 1-channel 1080p 30fps BT.1120 input or 2-channel 720p 30fps BT.656 input
- Supports multiplexer function up to four D1/960H channels
- Supports CCIR601 input with maximum resolution of 1080p
- Supports linear scaling-down/up function and four output resolutions per scan line
- Supports auto frame rate control function for four output resolutions per channel
- Supports eight private region masks in each channel
- Supports motion detection function
- Supports internal display loopback path
- Supports line de-noise
- Supports line false color suppression
- Supports line sharpness function for four output resolutions per channel
- Supports VBI detection and extraction
- Supports front-end image source cropping function
- Supports back-end image cropping function with four output resolutions per channel
- Supports channel arbitration
- Supports font-based OSD
- Supports OSD border and marquee functions for four output resolutions per channel
- Supports maximum four-image mark function
- Enhances front-end signal tolerancefor errors
- Dedicates high performance DMA
- Supports loopback mode function
- Provides clock output to video decoder to save external oscillator (27MHz, 54MHz, or 108MHz)

1.3.11 Display Controller

The display controller acquires the video data from the frame buffer and output to provide all necessary control signals for various TFT LCD monitors or TV encoders. The features of LCDC include:

- Supports TFT color display with 16bit BT.1120 or 16bit RGB565 bus interface
- Built-in digital TV data for composite output (NTSC/PAL)
- Programmable polarity and duration for output enable, vertical sync., horizontal sync., and pixel clock
- Supports PiP/PoP functions
- Cursor
- Supports three graphic planes and hardware cursor

1.3.12 I^2S Controller

The I^2S controller is a full-duplex synchronous serial interface that can connect to a variety of devices, such as the external analog-to-digital (A/D) converters, audio, and telecom codecs.

The features of the I^2S controller include:

- Supports serial data formats ranging from 4bits to 128bits in length
- Provides independent clock to ease the bit-clock generation
- Supports master and slave modes
- Internally or externally controlled serial bit clock
- Internally or externally controlled frame/sync.
- Programmable frame/sync. polarity
- Programmable serial bit clock polarity, phase, and frequency
- Programmable serial bit data sequence (MSB or LSB first)
- Programmable threshold interrupt of transmit/receive FIFOs
- Independently programmable interrupt enable/disable function
- Provides 32-word transmit FIFO and 32-word receive FIFO
- Supports DMA REQ/ACK for large data transfers
- Programmable I^2S format (Including the zero-bit padding, right-justified, or left-justified)

1.3.13 Mega Ethernet Controller (RMII MAC)

RMII MAC is a high-quality Ethernet controller with the DMA function. It includes the AHB wrapper, DMA engine, on-chip memories (TX FIFO and RX FIFO), MAC, and RMII interfaces. RMII MAC is an Ethernet controller that provides the AHB master capability. This controller is fully compliant with the IEEE 802.3 specification for 10/100Mbps RMII Ethernet. The RMII MAC Ethernet controller with the DMA function is used to handle all data transfers between the system memory and on-chip memories. With the DMA engine, this controller can reduce the CPU loading, maximize the performance, and minimize the FIFO size. RMII MAC contains the on-chip memories for buffering so that the external local buffer memory is not needed. The RMII interface supports two data rates, 10Mbps and 100Mbps. MAC provides the Wake-On-LAN function. It supports three wake-up events: Link status change, magic packet, and wake-up frame. This function allows the systems containing MAC to be woken up by the remote side. Other features of RMII MAC are described below:

- Supports network acceleration TOE engine (TCP/UDP/IP offload engine)
- Provides DMA engine for transmitting and receiving packets
- Supports zero-copy data transfer
- Supports programmable AHB burst size
- Supports transmit and receive interrupt mitigations
- Supports Wake-On-LAN function and three wake-up events: Link status change, Magic packet, and Wake-up frame
- Supports four Wake-On-LAN signals (Active high, active low, positive pulse, and negative pulse)
- Contains independent TX/RX FIFOs (2KB each)
- Supports half-duplex and full-duplex modes
- Supports flow control for full-duplex mode and backpressure for half-duplex mode

1.3.14 Secure Digital Card Controller (SDC)

The SDC controller not only supports both the SD and MMC interface protocols, but also supports the hot insertion/removal detections. For the SD interface protocol, the SDC controller provides write protection and the 1bit or 4bit wide bus for large data transfers without the CPU intervention. The features of the SDC controller include:

- Compliant with SD Host Controller Standard Specification, Version 3.0
- Compliant with SD Physical Layer Specification, Version 3.0
- Compliant with SDIO Card Specification, Version 2.0
- Compliant with MMC Card Specification, Version 4.3
- Supports UHS50/UHS104 UHS speed class card
- Supports 4bit wide SD/MMC data bus
- Supports data FIFO depth of 1Kbytes
- Built-in generation and check for 7bit and 16bit CRC data
- Variable clock rates: 0MHz ~ 50MHz for memory card
- Card detection (Insertion/Removal)
- Supports Read Wait mechanism for SDIO function
- Supports Suspend/Resume mechanism for SDIO function
- Write-protect for SD card
- Supports single-block or multiple-block card accesses for read, write, or erase operations
- Supports DMA REQ/ACK for large data transfers

1.3.15 AES/DES/TDES Cipher Controller

The AES/DES/TDES cipher controller provides efficient hardware implementation of the DES/Triple-DES /AES algorithm for high-performance encryption and decryption, which can be applied to various applications. Additionally, this controller provides the DMA function, which can reduce the data transfer overhead on the processor and improve the system performance. The features of the AES/DES/TDES cipher controller include:

- DES/Triple-DES encryption/decryption compliant with NIST standard
- AES 128/192/256bit encryption/decryption compliant with NIST standard
- Block cipher mode supports the following modes:
  - DES/Triple-DES
    - ECB mode
    - CBC mode
    - CFB mode
    - OFB mode
  - AES
    - ECB mode
    - CBC mode
    - CFB mode
    - OFB mode
    - CTR mode

1.3.16 Timer

The timer provides three independent sets of sub-timers. Each sub-timer can use the system clock (OSC) for increment or decrement counting. Two match registers are provided for each sub-timer. Whenever the value of the match registers equals to any one of the sub-timers, the timer interrupt will be immediately triggered. The issuance of a timer interrupt can be decided by the register setting when the overflow occurs. The features of the timer include:

- Supports three independent 32bit timer programming models
- Supports interrupts issued upon overflow and time-up
- Supports two match registers in each sub-timer
- Programmable decrementing/incrementing modes on counter

1.3.17 WatchDog Timer (WDT)

The WatchDog timer is used to prevent the system from infinite looping when the software trapped in a deadlock. In the normal operation, users restart WDT at the regular intervals before the counter counts down to zero. WDT generates one or a combination of the following signals: Reset, interrupt, or external interrupt signal. The features of WDT include:

- 32bit down counter
- Upon timeout, WDT outputs one or a combination of system reset/system interrupt/external interrupt signals
- Supports variable timeout periods of reset
- Access protection

1.3.18 GPIO

GPIO is used to input or output data from or to the system and device. It can be programmed as an input, an output, or an interrupt input. GPIO supports the rising edge, falling edge, both edges, and high-level or low-level interrupt sense type. The features of GPIO include:

- Separately triggers GPIO interrupt of each port
- Triggers at rising edge, falling edge, both edges, or high-level or low–level interrupt sense type for interrupt generation of each port
- Pulls high or pulls low for each port
- Provides de-bounce function of each port
- Separately sets or clears output data bits
- Sets to input mode at hardware reset for all ports

1.3.19 I²C Bus

I^2C is a two-wire bidirectional serial bus that provides a simple and efficient method of data exchange when minimizing the interconnection between devices. The I^2C bus interface controller allows the host processor to serve as a master or slave residing on the I^2C bus. Data are transmitted to and received from the I^2C bus via a buffered interface. The features of the I^2C bus include:

- Supports standard and fast modes through programming clock division register
- Supports 7bit, 10bit, and general-call addressing modes
- Glitch suppression throughout de-bounce circuits
- Programmable slave address
- Supports master-transmit, master-receive, slave-transmit, and slave-receive modes
- Slave mode general-call address detection

1.3.20 UART Controller

The UART controller is a serial communications element that is backward compatible with 16550 to support the existing communication software. The features of the UART controller include:

- High-speed NS 16C550A-compatible UART
- Programmable baud rates up to 1152Kbps
- Ability to add or delete standard asynchronous communication bits (Start, stop, and parity) in serial data
- Programmable baud rate generator that allowed internal clock to be divided by 1 to (2^{16} - 1) to generate an internal 16X clock
- Fully programmable serial interface:
  - 5bit, 6bit, 7bit, or 8bit character
  - Even, odd, and no parity detections
  - 1, 1.5, or 2 stop-bit generation
- Complete status reporting capability
- Capable of generating and detecting the line breaks
- Fully prioritized interrupt system controls
- Separated DMA requests for data transmission and reception services
- Break, parity, overrun, and framing error simulation in the UART mode

Chapter 2

Signal Description

This chapter contains the following sections:

- 2.1 Signal Descriptions
- 2.2 Pin Assignments

2.1 Signal Descriptions

Table 2-1 and Table 2-2 list and describe the signals of GM8136S/GM8135S. In these tables, "I" indicates the input signals, "O" indicates the output signals, "I/O" indicates the bidirectional signals, "A" indicates the analog signals, "SUP" indicates the power supply pins, "SMT" indicates the input Schmitt-trigger signals , "PU" indicates the internal pull-up signals, "PD" indicates the internal pull-down signals, and "PP" indicates the programmable internal pull-up or pull-down signals.

Table 2-1. TFBGA-196 Signal Descriptions

Signal Name	Direction	BGA Ballmap	LQFP Pin Number	Description
OM	I, PD	E7	113	Test enable signal, active high Keep this signal open or floating for the normal function
PWM0	O	B7	114	Pulse Width Modulation output
GPIO_1[28]	I/O, PP			General-purpose I/O
DMIC_DATA	I, PU			Digital microphone data
PWM1	O	A7	115	Pulse Width Modulation output
GPIO_1[29]	I/O, PP			General-Purpose I/O
DMIC_CLK	O			Digital microphone clock
CAP_RST	I/O, PD	A5	116	General-Purpose I/O using GPIO_0[5] for the external sensor reset
CAP_CLKOUT	O	B4	117	Provide the external sensor clock source
GPIO_0[6]	I/O, PP			General-Purpose I/O
BAYER_CLK	I	A4	118	Bayer sensor clock input
CAP0_CLK	I			BT.1120/CCIR601 clock input
CAP1_CLK	I			BT.656 clock input
GPIO_0[9]	I/O, PP			General-Purpose I/O
MPRX_RBIAS	A	A6	122	Serial Combo PHY BIAS
MPRX_DN0	A	A3	123	Differential data lane 0-
CAP0_CLK	I			BT.1120/BT.656 clock input
CAP0_HS	I			CCIR601 horizontal sync.
BAYER_HS	I			Bayer sensor horizontal sync.

Signal Name	Direction	BGA Ballmap	LQFP Pin Number	Description
MPRX_DP0	A	B3	124	Differential data lane 0+
CAP0_VS	I			CCIR601 vertical sync.
BAYER_VS	I			Bayer sensor vertical sync.
MPRX_CKN	A	A2	125	Differential clock lane-
CAP0_D[15]	I			BT.1120/CCIR601 data bit 15
BAYER_D11	I			Bayer sensor data bit 11
CAP1_D[7]	I			BT.656 data bit 7
MPRX_CKP	A	B2	126	Differential clock lane +
CAP0_D[14]	I			BT.1120/CCIR601 data bit 14
BAYER_D10	I			Bayer sensor data bit 10
CAP1_D[6]	I			BT.656 data bit 6
MPRX_DN1	A	A1	127	Differential data lane 1-
GPIO_0[7]	I/O, PP			General-Purpose I/O
CAP0_D[13]	I			BT.1120/CCIR601 data bit 13
BAYER_D9	I			Bayer sensor data bit 9
CAP1_D[5]	I			BT.656 data bit 5
MPRX_DP1	A	B1	128	Differential data lane 1+
GPIO_0[8]	I/O, PP			General-Purpose I/O
CAP0_D[12]	I			BT.1120/CCIR601 data bit 12
BAYER_D8	I			Bayer sensor data bit 8
CAP1_D[4]	I			BT.656 data bit 4
BAYER_D7	I, PU	C1	1	Bayer sensor data bit 7-
CAP0_D[11]	I PU			BT.1120/CCIR601 data bit 11
GPIO_0[20]	I/O, PP			Gereral-Purpose IO
SD1_CD	I, PU			SD card detect signal
CAP1_D[3]	I, PU			BT.656 data bit 3
BAYER_D6	I, PU	C2	2	Bayer sensor data bit 6
CAP0_D[10]	I, PU			BT.1120/CCIR601 data bit 10
GPIO_0[21]	I/O, PP			General-Purpose I/O
SD1_DAT[1]	I/O, PU			SD card data
CAP1_D[2]	I, PU			BT.656 data bit 2

Signal Name	Direction	BGA Ballmap	LQFP Pin Number	Description
BAYER_D5	I, PU	D1	3	Bayer sensor data bit 5
CAP0_D[9]	I, PU			BT.1120/CCIR601 data bit 9
GPIO_0[22]	I/O, PP			General-Purpose I/O
SD1_DAT[0]	I/O, PU			SD card data
CAP1_D[1]	I, PU			BT.656 data bit 1
BAYER_D4	I, PU	D2	4	Bayer sensor data bit 4
CAP0_D[8]	I, PU			BT.1120/CCIR601 data bit 8
GPIO_0[23]	I/O, PP			General-Purpose I/O
SD1_CLK	O			SD card clock
CAP1_D[0]	I, PU			BT.656 data bit 0
CAP0_D[7]	I, PU	E1	5	BT.656/BT.1120/CCIR601 data bit 7
GPIO_0[10]	I/O, PP			General-Purpose I/O
BAYER_D3	I, PU			Bayer sensor data bit 3
SD1_CMD_RSP	I/O, PU			SD card command/response signal
CAP0_D[6]	I, PU	E2	6	BT.656/BT.1120/CCIR601 data bit 6
GPIO_0[11]	I/O, PP			General-Purpose I/O
BAYER_D2	I, PU			Bayer sensor data bit 2
SD1_DAT[3]	I/O, PU			SD card data
CAP0_D[5]	I, PU	F1	8	BT.656/BT.1120/CCIR601 data bit 5
GPIO_0[12]	I/O, PP			General-Purpose I/O
BAYER_D1	I, PU			Bayer sensor data bit 1
SD1_DAT[2]	I/O, PU			SD card data
CAP0_D[4]	I, PU	F2	10	BT.656/BT.1120/CCIR601 data bit 4
GPIO_0[13]	I/O, PP			General-Purpose I/O
BAYER_D0	I, PU			Bayer sensor data bit 0
CAP0_D[3]	I, PU	G1	11	BT.656/BT.1120/CCIR601 data bit 3
GPIO_0[14]	I/O, PP			General-Purpose I/O
I2C_SCL	I/O, PU			I²C clock
SSP1_FS	I/O, PU			SSP/I²S frame sync
CAP0_D[2]	I, PU	G2	12	BT.656/BT.1120/CCIR601 data bit 2
GPIO_0[15]	I/O, PP			General-Purpose I/O
I2C_SDA	I/O, PU			I²C data
SSP1_RXD	I, PU			SSP/I²S data-in

Signal Name	Direction	BGA Ballmap	LQFP Pin Number	Description
CAP0_D[1]	I, PU	H1	13	BT.656/BT.1120/CCIR601 data bit 1
GPIO_0[16]	I/O, PP			General-Purpose I/O
SSP1_TXD	O, PU			SSP/I²S data-out
CAP0_D[0]	I, PU	H2	14	BT.656/BT.1120/CCIR601 data bit 0
GPIO_0[17]	I/O, PP			General-Purpose I/O
SSP1_SCLK	I/O			SSP/I²S bit clock
I2C_SCL	I/O, PU	J1	15	I²C clock
GPIO_0[18]	I/O, PP			General-Purpose I/O
PWM2	O			Pulse Width Modulation output
I2C_SDA	I/O, PU	J2	16	I²C data
GPIO_0[19]	I/O, PP			General-Purpose I/O
PWM3	O			Pulse Width Modulation output
SPI_TXD	I/O	L2	21	SPI serial data output (For 1xI/O)/serial data input and output (For 2x I/O)
SPI_SCLK	I/O	L1	20	SPI clock
SPI_FS	O, PU	K1	17	SPI memory chip selection Active low
SPI_RXD	I/O, PU	K2	19	SPI serial data input (For 1xI/O)/serial data input and output (For 2x I/O mode)
SD_CMD_RSP	I/O, PU	N2	27	SD card command/response signal
GPIO_1[2]	I/O, PP			General-Purpose I/O
SD_CLK	O	P1	25	SD card clock signal
GPIO_1[1]	I/O, PP			General-Purpose I/O
SD_DAT[0]	I/O, PU	N1	24	SD card data
GPIO_0[24]	I/O, PP			General-Purpose I/O
SD_DAT[1]	I/O, PU	M2	23	SD card data
GPIO_0[25]	I/O, PP			General-Purpose I/O
SD_DAT[2]	I/O, PU	N3	29	SD card data
GPIO_0[26]	I/O, PP			General-Purpose I/O
SD_DAT[3]	I/O, PU	P2	28	SD card data
GPIO_0[27]	I/O, PP			General-purpose I/O
SD_CD	I, PU	M1	22	SD card detect signal
GPIO_1[0]	I/O, PP			General-Purpose I/O

Signal Name	Direction	BGA Ballmap	LQFP Pin Number	Description
UART2_SIN	I, PU	L3	31	UART2 receive
GPIO_1[3]	I/O, PP			General-Purpose I/O
UART2_SOUT	O	M3	32	UART2 transmit
GPIO_1[4]	I/O, PP			General-Purpose I/O
DAC_COMP	A	P3	33	Analog common-mode voltage
DAC_IOUTA	A	P4	35	Positive current output
SAR_ADC_XAIN0	A	N4	36	ADC XAIN input channel 0
SAR_ADC_XAIN1	A	M4	37	ADC XAIN input channel 1
ADDA_MICIN	A	N5	38	Analog microphone input of the negative channel
ADDA_VCM	A	P5	40	Analog common-mode voltage
ADDA_SPKOUTN	A	P6	42	Analog speaker out of the negative channel
ADDA_SPKOUTP	A	N6	43	Analog speaker out of the positive channel
OTG_DM	A	P8	47	OTG differential data D-
OTG_DP	A	N8	48	OTG differential data D+
OTG11_DN	A	P9	49	OTG 1.1 differential data D-
GPIO_1[30]	I			General-Purpose I/O, only input
OTG11_DP	A	N9	50	OTG 1.1 differential data D+
GPIO_1[31]	I			General-Purpose I/O, only input
SSP1_FS	I/O, PU	N10	52	SPI/SSP/I²S frame sync
GPIO_1[24]	I/O, PP			General-Purpose I/O
UART1_SIN	I, PU			UART1 receive
SSP1_TXD	O, PU	M10	53	SPI/SSP/I²S data-out
GPIO_1[25]	I/O, PP			General-Purpose I/O
UART1_SOUT	O, PU			UART1 transmit
SSP1_RXD	I, PU	M11	54	SPI/SSP/I²S data-in
GPIO_1[26]	I/O, PP			General-Purpose I/O
LC_VS	O			LCD vertical sync
SSP1_SCLK	I/O	M12	56	SPI/SSP/I²S bit clock
GPIO_1[27]	I/O, PP			General-Purpose I/O
LC_HS	O			LCD horizontal sync

Signal Name	Direction	BGA Ballmap	LQFP Pin Number	Description
RMII_MDIO	IO, PU	P10	57	Data of PHY management for RMII
GPIO_1[7]	I/O, PP			General-purpose I/O
LC_DATA[0]	O			RGB565 data out [0]
TV_DATA[0]	O			BT.1120/BT.656 data out [0]
RMII_MDC	O	P11	58	Clock of PHY management for RMII
GPIO_1[8]	I/O, PP			General-Purpose I/O
LC_DATA[1]	O			RGB565 data out [1]
TV_DATA[1]	O			BT.1120/BT.656 data out [1]
RMII_RX_ER	I	N11	59	RMII receive error
GPIO_1[12]	I/O, PP			General-Purpose I/O
LC_DATA[2]	O			RGB565 data out [2]
TV_DATA[2]	O			BT.1120/BT.656 data out[2]
RMII_RX_CRS_DV	I, PU	P12	60	Receive carrier sense/data valid
GPIO_1[9]	I/O, PP			General-Purpose I/O
LC_DATA[3]	O			RGB565 data out [3]
TV_DATA[3]	O			BT.1120/BT.656 data out[3]
RMII_RXD[0]	I, PU	N12	61	RMII receive data[0]
GPIO_1[10]	I/O, PP			General-Purpose I/O
LC_DATA[4]	O			RGB565 data out [4]
TV_DATA[4]	O			BT.1120/BT.656 data out [4]
RMII_RXD[1]	I, PU	P13	62	RMII receive data[1]
GPIO_1[11]	I/O, PP			General-Purpose I/O
LC_DATA[5]	O			RGB565 data out [5]
TV_DATA[5]	O			BT.1120/BT.656 data out [5]
RMII_PHYLINK	I, PU	N13	63	PHY link status for RMII
GPIO_1[13]	I/O, PP			General-Purpose I/O
LC_DATA[6]	O			RGB565 data out [6]
TV_DATA[6]	O			BT.1120/BT.656 data out [6]
RMII_CKO	O	P14	64	RMII PHY clock out
GPIO_1[14]	I/O, PP			General-Purpose I/O
LC_DATA[7]	O			RGB565 data out [7]
TV_DATA[7]	O			BT.1120/BT.656 data out [7]

Signal Name	Direction	BGA Ballmap	LQFP Pin Number	Description
RMII_TXD[0]	O	N14	65	RMII transmit data[0]
GPIO_1[15]	I/O, PP			General-Purpose I/O
LC_DATA[8]	O			RGB565 data out [8]
TV_DATA[8]	O			BT.1120 data out [8]
RMII_TXD[1]	O	M13	66	RMII transmit data[1]
GPIO_1[16]	I/O, PP			General-Purpose I/O
LC_DATA[9]	O			RGB565 data out [9]
TV_DATA[9]	O			BT.1120 data out [9]
RMII_TX_EN	O	M14	67	Transmit enable for the RMIII mode
GPIO_1[17]	I/O, PP			General-Purpose I/O
LC_DATA[10]	O			RGB565 data out [10]
TV_DATA[10]	O			BT.1120 data out [10]
RMII_CK_IN	I	L14	69	RMII clock source input
GPIO_1[18]	I/O, PP			General-Purpose I/O
LC_DATA[11]	O			RGB565 data out [11]
TV_DATA[11]	O			BT.1120 data out [11]
GPIO_1[19]	I/O, PP	L13	70	General-Purpose I/O
SD1_CD	I, PU			SD card detect signal
LC_DATA[12]	O			RGB565 data out [12]
TV_DATA[12]	O			BT.1120 data out [12]
GPIO_1[20]	I/O, PP	K14	71	General-Purpose I/O
SD1_DAT[1]	IO, PU			SD card data
LC_DATA[13]	O			RGB565 data out [13]
TV_DATA[13]	O			BT.1120 data out [13]
GPIO_1[21]	I/O, PP	K13	73	General-Purpose I/O
SD1_DAT[0]	IO, PU			SD card data
LC_DATA[14]	O			RGB565 data out [14]
TV_DATA[14]	O			BT.1120 data out [14]
GPIO_1[22]	I/O, PP	J14	74	General-Purpose I/O
SD1_CLK	O			SD card clock signal
LC_DATA[15]	O			RGB565 data out [15]
TV_DATA[15]	O			BT.1120 data out [15]

Signal Name	Direction	BGA Ballmap	LQFP Pin Number	Description
GPIO_1[23]	I/O, PP	J13	75	General-Purpose I/O
LC_PCLK	O			RGB656 clock
TV_PCLK	O			BT.1120/BT.656 clock
GPIO_0[28]	I/O, PP	H14	76	General-Purpose I/O
SSP1_FS	IO, PU			SPI/SSP/I²S frame sync
GPIO_0[29]	I/O, PP	H13	77	General-Purpose I/O
SSP1_TXD	O, PU			SPI/SSP/I²S data-out
SD1_CMD_RSP	I/O, PU			SD card command/response signal
GPIO_0[30]	I/O, PP	G14	78	General-Purpose I/O
SSP1_RXD	I, PU			SPI/SSP/I²S data-in
SD1_DAT[3]	I/O, PU			SD card data
OTG_48M_dbg	O			OTG 48MHz out for debugging
GPIO_0[31]	I/O, PP	G13	79	General-Purpose I/O
SSP1_SCLK	I/O, PU			SPI/SSP/I²S bit clock
SD1_DAT[2]	I/O			SD card data
OTG_30M_dbg	O			OTG 30MHz out for debugging
NTRST	I, PD	H12	80	ICE reset
GPIO_0[0]	I/O, PP			General-Purpose I/O
TDI	I, PU	H11	81	ICE data input
GPIO_0[4]	I/O, PP			General-Purpose I/O
PWM7	O			Pulse Width Modulation output
DMIC_CLK	O			Digital microphone clock
TMS	I, PU	G12	82	ICE mode selection
GPIO_0[2]	I/O, PP			General-purpose I/O
PWM5	O			Pulse Width Modulation output
UART2_SOUT	O			UART2 transmit
TCK	I, PU	G11	83	ICE clock input
GPIO_0[1]	I/O, PP			General-Purpose I/O
PWM4	O			Pulse Width Modulation output
UART2_SIN	I, PU			UART2 receive
TDO	O, PU	F12	84	ICE data output
GPIO_0[3]	I/O, PP			General-Purpose I/O
PWM6	O			Pulse Width Modulation output
DMIC_DATA	O			Digital microphone data

Signal Name	Direction	BGA Ballmap	LQFP Pin Number	Description
UART0_SIN	I, PU	F14	86	UART0 receive
GPIO_1[5]	I/O, PP			General-Purpose I/O
UART0_SOUT	O	F13	87	UART0 transmit
GPIO_1[6]	I/O, PP			General-Purpose I/O
DDR_VREF	SUP	E14	88	Reference voltage for the receivers
ZQ	SUP	E13 (package GM8136S-BC-A)	105 (package GM8136S-QC-A)	Reference pin for the ZQ calibration
VCC11A_O_REG	A	A12	107	Output voltage of the linear regulator
OSCHIN	A	B9	109	30MHz crystal input
OSCHIO	A	A9	108	30MHz crystal output
OSCLIN	A	A8	111	32768Hz crystal input
OSCLIO	A	B8	112	32768Hz crystal output
VCCK	SUP	A13, A14, B12, B13, B14, C6, C7, C8, C9, C10, C11, C12, D6, D7, D8, D9, D10, D11	9, 18, 30, 45, 55, 72, 85, 89, 94, 98, 100, 103, 119	Positive supply of the internal logic This signal must be connected to the 1.1V voltage supply on PCB.
VCCIO	SUP	D5, E5, F5, G5, H5, J5, K5, N7, P7	26, 51, 68	Positive supply of all I/O pins, except for DDR This signal must be connected to the 3.3V voltage supply on PCB.
VCC15O_DDRCKA	SUP	C13	101	Positive supply of the DDR command/address PHY I/O pins This signal must be connected to the 1.5V voltage supply on PCB.
VCC15O_DDRCKD	SUP	D13	93	Positive supply of the DDR data PHY I/O pins This signal must be connected to the 1.5V voltage supply on PCB
VCCDDR	SUP	C14, D12, D14, E11, E12, F10, F11, G10	90, 91, 92, 95, 96, 97, 99, 102, 104	Positive supply of the DDR I/O pins This signal must be connected to the 1.5V voltage supply on PCB.

Signal Name	Direction	BGA Ballmap	LQFP Pin Number	Description
GND	SUP	B10, C3, C4, D3, D4, E3, E4, E6, E8, E9, E10, F3, F4, F6, F7, F8, F9, G3, G4, G6, G7, G8, G9, H3, H4, H6. H7, H8, H9, H10, J3, J4, J6, J7, J8, J9, J10, J11, K3, K4, K7, K8, K9, K10, K11, K12, L4, L7, L8, L9, L10, L11, L12, M8, M9	Down_bond	Grounded supply of the DDR I/O pins Ground supply of all I/O pins and internal logic
VCC33A_REG	SUP	A11	106	3.3V analog power supply
VCC33A_OSCL	SUP	A10	110	3.3V analog power supply
VCC3IO_BAYER	SUP	B5	120	1.8V ~ 3.3V analog/Capture/Bayer power supply
VCC3IO_CAP0	SUP	C5	7	1.8V ~ 3.3V Capture/Bayer power supply
GND33A_REG	SUP	B11	Down_bond	Analog ground
VCC11A_MPRX	SUP	B6	121	1.1V analog power supply
VCC33_HSRT	SUP	M7	46	3.3V analog power supply
VCC33A_DAC	SUP	L5	34	3.3V analog power supply
VCC33A_SPK_ADDA	SUP	M6	44	3.3V analog power supply
VCC33A_ADDA	SUP	M5	39	3.3V analog power supply
GND33A_SPK_ADDA	SUP	L6	41	Analog ground
GND33A_ADDA	SUP	K6	Down_bond	Analog ground

Video in Power Domain	Pin/Ball Name
VCC3IO_BAYER	X_OM, X_PWM0, XPWM1, X_CAP_RST, X_CAP_CLKOUT, X_BAYER_CLK, X_MPRX_DN0, X_MPRX_DP0, X_MPRX_CKN, X_MPRX_CKP, X_MPRX_DN1, X_MPRX_DP1, X_BAYER_D7, X_BAYER_D6, X_BAYER_D5, and X_BAYER_D4
VCC3IO_CAP0	X_CAP0_D[7:0]

Table 2-2. Pin/Ball Initial State and Default State

Pin/Ball Name	Power-On-Reset State	Multi-Function Pin and Function Default State				Control Register and Default	Unused PCB Setting and Comment
-	-	regxx bit[y:x] = 2'b00	regxx bit[y:x] = 2'b01	regxx bit[y:x] = 2'b10	regxx bit[y:x] = 2'b11	-	-
X_OM	I, internal pull-down	-	-	-	-	-	Jumper setting 13, internal pull-down
X_PWM0	-	GPIO_1[28], I, pull-up	PWM0, O	DMIC_DATA, I, PU	-	reg64[13:12], default = 2'b00	Floating
X_PWM1	-	GPIO_1[29], I, pull-up	PWM1, O	DMIC_CLK, O	-	reg64[15:14], default = 2'b00	Floating
X_CAP_RST	I, internal pull-down	GPIO_0[5], I, pull-down	-	-	-	-	Jumper setting 0, internal pull-down
X_CAP_CLKOUT	I, internal pull-up	GPIO_0[6], I, pull-up	CAP_CLKOUT, O	-	-	reg50[13:12], default = 2'b00	Floating
X_BAYER_CLK	I, internal pull-up	GPIO_0[9], I, pull-up	CAP0_CLK, O	BAYER_CLK /CAP1_CLK, O	-	reg50[15:14], default = 2'b00	Floating
X_MPRX_DN0	I	CAP0_CLK, I	CAP0_HS, I	BAYER_HS, I	-	reg50[17:16], default = 2'b00	Floating, based on 9ba00060[3] = 1'b1
X_MPRX_DP0	I	-	CAP0_VS, I	CAP0_VS, I	-	reg50[19:18], default = 2'b00	Floating, based on 9ba00060[3] = 1'b1
X_MPRX_CKN	I	-	CAP0_D[15], I	BAYER_D[11] /CAP1_D[7], I	-	reg50[21:20], default = 2'b00	Floating, based on 9ba00060[3] = 1'b1
X_MPRX_CKP	I	-	CAP0_D[14], I	BAYER_D[10] /CAP1_D[6], I	-	reg50[23:22], default = 2'b00	Floating, based on 9ba00060[3] = 1'b1
X_MPRX_DN1	I	GPIO_0[7], I	CAP0_D[13], I	BAYER_D[9] /CAP1_D[5], I	-	reg50[25:24], default = 2'b00	Floating, based on 9ba00060[3] = 1'b1
X_MPRX_DP1	I	GPIO_0[8], I	CAP0_D[12], I	BAYER_D[8] /CAP1_D[4], I	-	reg50[27:26], default = 2'b00	Floating, based on 9ba00060[3] = 1'b1
X_BAYER_D7	I, internal pull-up	GPIO_0[20], I, pull-up	CAP0_D[11], I, pull-up	BAYER_D7, I, pull-up	SD1_CD, I, pull-up	reg50[29:28], default = 2'b00	Floating
X_BAYER_D6	I, internal pull-up	GPIO_0[21], I, pull-up	CAP0_D[10], I, pull-up	BAYER_D6, I, pull-up	SD1_D1, I, pull-up	reg50[31:30], default = 2'b00	Floating

Pin/Ball Name	Power-On-Reset State	Multi-Function Pin and Function Default State				Control Register and Default	Unused PCB Setting and Comment
X_BAYER_D5	I, internal pull-up	GPIO_0[22], I, pull-up	CAP0_D[9], I, pull-up	BAYER_D5, I, pull-up	SD1_D0, I, pull-up	reg54[1:0], default = 2'b00	Floating
X_BAYER_D4	I, internal pull-up	GPIO_0[23], I, pull-up	CAP0_D[8], I, pull-up	BAYER_D4, I, pull-up	SD1_CLK, O	reg54[3:2], default = 2'b00	Floating
X_CAP0_D[7]	I, internal pull-up	GPIO_0[10], I, pull-up	CAP0_D[7], I, pull-up	BAYER_D3, I, pull-up	SD1_CMD_RSP, O, 1	reg54[7:6], default = 2'b00	Floating
X_CAP0_D[6]	I, internal pull-up	GPIO_0[11], I, pull-up	CAP0_D[6], I, pull-up	BAYER_D2, I, pull-up	SD1_D3, I, pull-up	reg54[9:8], default = 2'b00	Floating
X_CAP0_D[5]	I, internal pull-up	GPIO_0[12], I, pull-up	CAP0_D[5], I, pull-up	BAYER_D1, I, pull-up	SD1_D2, I, pull-up	reg54[11:10], default = 2'b00	Floating
X_CAP0_D[4]	I, internal pull-up	GPIO_0[13], I, pull-up	CAP0_D[4], I, pull-up	BAYER_D0, I, pull-up	-	reg54[13:12], default = 2'b00	floating
X_CAP0_D[3]	I, internal pull-up	GPIO_0[14], I, pull-up	CAP0_D[3], I, pull-up	I2C_SCL, I, pull-up	I2S1_FS, O, pull-up	reg54[15:14], default = 2'b00	Floating
X_CAP0_D[2]	I, internal pull-up	GPIO_0[15], I, pull-up	CAP0_D[2], I, pull-up	I2C_SDA, I, pull-up	I2S1_RXD, I, pull-up	reg54[17:16], default = 2'b00	Floating
X_CAP0_D[1]	I, internal pull-up	GPIO_0[16], I, pull-up	CAP0_D[1], I, pull-up		I2S1_TXD, I, pull-up	reg54[19:18], default = 2'b00	Floating
X_CAP0_D[0]	I, internal pull-up	GPIO_0[17], I, pull-up	CAP0_D[0], I, pull-up		I2S1_SCLK, O	reg54[21:20], default = 2'b00	Floating
X_I2C_SCL	I, internal pull-up	GPIO_0[18], I, pull-up	I2C_SCL, I, pull-up	PWM2, O	-	reg54[23:22], default = 2'b00	Floating
X_I2C_SDA	I, internal pull-up	GPIO_0[19], I, pull-up	I2C_SDA, I, pull-up	PWM3, O	-	reg54[25:24], default = 2'b00	Floating
X_SPI_TXD	I	SPI_TXD, I	-	-	-	-	Pull-up
X_SPI_SCLK	I, jumper setting	SPI_SCLK, O, 0	-	-	-		Jumper setting 6
X_SPI_FS	O, internal pull-up	SPI_FS, O, pull-up	-	-	-	-	Floating
X_SPI_RXD	I, internal pull-up	SPI_RXD, I, pull-up	-	-	-	-	Floating
X_SD_CD	I, internal pull-up	GPIO_1[0], I, pull-up	SD_CD, I, pull-up	-	-	reg58[19:18], default = 2'b00	Floating
X_SD_DAT[1]	I, internal pull-up	GPIO_0[25], I, pull-up	SD_DAT[1], I, pull-up	-	-	reg58[5:4], default = 2'b00	Floating
X_SD_DAT[0]	I, internal pull-up	GPIO_0[24], I, pull-up	SD_DAT[0], I, pull-up	-	-	reg58[3:2], default = 2'b00	Floating
X_SD_CLK	I, internal pull-down	GPIO_1[1], I, pull-down	SD_CLK, O	-	-	reg58[21:20], default = 2'b00	Jumper setting 9, internal pull-down
X_SD_CMD_RSP	I, internal pull-up	GPIO_1[2], I, pull-up	SD_CMD_RSP, O, 1, pull-up	-	-	reg58[23:22], default = 2'b00	Floating

Pin/Ball Name	Power-On-Reset State	Multi-Function Pin and Function Default State				Control Register and Default	Unused PCB Setting and Comment
X_SD_DAT[3]	I, internal pull-up	GPIO_0[27], I, pull-up	SD_DAT[3], I, pull-up	-	-	reg58[9:8], default = 2'b00	Floating
X_SD_DAT[2]	I, internal pull-up	GPIO_0[26], I, pull-up	SD_DAT[2], I, pull-up	-	-	reg58[7:6], default = 2'b00	Floating
X_UART2_SIN	I, internal pull-up	GPIO_1[3], I, pull-up	UART2_SIN, I, pull-up	-	-	reg58[25:24], default = 2'b00	Floating
X_UART2_SOUT	I, internal pull-up	GPIO_1[4], I, pull-up	UART2_SOUT, O, 1	-	-	reg58[27:26], default = 2'b00	Jumper setting 8, internal pull-up
X_OTG11_DN	I	GPIO_1[30], I	OTG11_DN	-	-	reg64[17:16], default = 2'b00	Floating
X_OTG11_DP	I	GPIO_1[31], I	OTG11_DP	-	-	reg64[17:16], default = 2'b00	Floating
X_I2S1_FS	I, internal pull-up	GPIO_1[24], I, pull-up	I2S1_FS, O, pull-up	UART1_SIN, I, pull-up	-	reg64[3:2], default = 2'b00	Floating
X_I2S1_TXD	I, internal pull-down	GPIO_1[25], I, pull-down	I2S1_TXD, I, pull-up	UART1_SOUT, O, 1, pull-up	-	reg64[5:4], default = 2'b00	Jumper setting
X_I2S1_RXD	I, internal pull-up	GPIO_1[26], I, pull-up	I2S1_RXD, I, pull-up	LC_VS, O	-	reg64[7:6], default = 2'b00	Floating
X_I2S1_SCLK	I, internal pull-up	GPIO_1[27], I, pull-up	I2S1_SCLK, O	LC_HS, O	-	reg64[1:0], default = 2'b00	Floating
X_RMII_MDIO	I, internal pull-up	GPIO_1[7], I, pull-up	RMII_MDIO, I, pull-up	LC_DATA[0], O	TV_DATA[0], O	reg5c[1:0], default = 2'b00	Floating
X_RMII_MDC	I, internal pull-up	GPIO_1[8], I, pull-up	RMII_MDC, O, 0	LC_DATA[1], O	TV_DATA[1], O	reg5c[3:2], default = 2'b00	Floating
X_RMII_RX_ER	I, internal pull-up	GPIO_1[12], I, pull-up	RMII_RER, I	LC_DATA[2], O	TV_DATA[2], O	reg5c[11:10], default = 2'b00	Floating
X_RMII_RX_CRS_DV	I, internal pull-up	GPIO_1[9], I, pull-up	RMII_RCRS_DV, I, pull-up	LC_DATA[3], O	TV_DATA[3], O	reg5c[5:4], default = 2'b00	Floating
X_RMII_RXD[0]	I, internal pull-up	GPIO_1[10], I, pull-up	RMII_RXD[0], I, pull-up	LC_DATA[4], O	TV_DATA[4], O	reg5c[7:6], default = 2'b00	Floating
X_RMII_RXD[1]	I, internal pull-up	GPIO_1[11], I, pull-up	RMII_RXD[1], I, pull-up	LC_DATA[5], O	TV_DATA[5], O	reg5c[9:8], default = 2'b00	Floating
X_RMII_PHYLINK	I, internal pull-up	GPIO_1[13], I, pull-up	RMII_PHYLINK, I, pull-up	LC_DATA[6], O	TV_DATA[6], O	reg5c[13:12], default = 2'b00	Floating
X_RMII_CKO	I, internal pull-up	GPIO_1[14], I, pull-up	RMII_CKO, O	LC_DATA[7], O	TV_DATA[7], O	reg5c[15:14], default = 2'b00	Floating
X_RMII_TXD[0]	I, internal pull-up	GPIO_1[15], I, pull-up	RMII_TXD[0], O	LC_DATA[8], O	TV_DATA[8], O	reg5c[17:16], default = 2'b00	Floating
X_RMII_TXD[1]	I, internal pull-up	GPIO_1[16], I, pull-up	RMII_TXD[1], O	LC_DATA[9], O	TV_DATA[9], O	reg5c[19:18], default = 2'b00	Floating

Pin/Ball Name	Power-On-Reset State	Multi-Function Pin and Function Default State				Control Register and Default	Unused PCB Setting and Comment
X_RMII_TX_EN	I, internal pull-down	GPIO_1[17], I, pull-down	RMII_TX_EN, O	LC_DATA[10], O	TV_DATA[10], O	reg5c[21:20], default = 2'b00	Floating
X_RMII_RST	I, internal pull-down	GPIO_1[18], I, pull-down	RMII_CK_IN, I	LC_DATA[11], O	TV_DATA[11], O	reg5c[23:22], default=2'b00	Floating
X_GPIO_DAT[4]	I, internal pull-up	GPIO_1[19], I, pull-up	SD1_CD, I, pull-up	LC_DATA[12], O	TV_DATA[12], O	reg5c[25:24], default = 2'b00	Floating
X_GPIO_DAT[5]	I, internal pull-up	GPIO_1[20], I, pull-up	SD1_DAT[1], I, pull-up	LC_DATA[13], O	TV_DATA[13], O	reg5c[27:26], default = 2'b00	Floating
X_GPIO_DAT[6]	I, internal pull-up	GPIO_1[21], I, pull-up	SD1_DAT[0], I, pull-up	LC_DATA[14], O	TV_DATA[14], O	reg5c[29:28], default = 2'b00	Floating
X_GPIO_DAT[7]	I, internal pull-up	GPIO_1[22], I, pull-up	SD1_CLK, O	LC_DATA[15], O	TV_DATA[15], O	reg5c[31:30], default = 2'b00	Floating
X_TV_PCLK	I, internal pull-up	Don't use, I, pull-up	-	LC_PCLK, O	TV_PCLK, O	reg64[9:8], default = 2'b00	Floating
X_GPIO_DAT[0]	I, internal pull-up	GPIO_0[28], I, pull-up	I2S1_FS, O, pull-up	-	-	reg58[11:10], default = 2'b00	Floating
X_GPIO_DAT[1]	I, internal pull-up	GPIO_0[29], I, pull-up	I2S1_TXD, I, pull-up	SD1_CMD_RSP, O, 1, pull-up	-	reg58[13:12], default = 2'b00	Floating
X_GPIO_DAT[2]	I, internal pull-up	GPIO_0[30], I, pull-up	I2S1_RXD, I, pull-up	SD1_DAT[3], I, pull-up	OTG_48M_dbg, O	reg58[15:14], default = 2'b00	Floating
X_GPIO_DAT[3]	I, internal pull-up	GPIO_0[31], I, pull-up	I2S1_SCLK, O	SD1_DAT[2], I, pull-up	OTG_30M_dbg, O	reg58[17:16], default = 2'b00	Floating
X_NTRST	I, internal pull-down	NTRST, I, pull-down	GPIO_0[0], I, pull-down	-	-	reg50[1:0], default = 2'b00	Floating
X_TDI	I, internal pull-up	TDI, I, pull-up	GPIO_0[4], I, pull-up	PWM7, O	DMIC_CLK, O	reg50[9:8], default = 2'b00	Floating
X_TMS	I, internal pull-up	TMS, I, pull-up	GPIO_0[2], I, pull-up	PWM5, O	UART2_SOUT, O	reg50[5:4], default = 2'b00	Floating
X_TCK	I, internal pull-up	TCK, I, pull-up	GPIO_0[1], I, pull-up	PWM4, O	UART2_SIN, I, PU	reg50[3:2], default = 2'b00	Floating
X_TDO	I, internal pull-up	TDO, I, pull-up	GPIO_0[3], I, pull-up	PWM6, O	DMIC_DATA, I, PU	reg50[7:6], default = 2'b00	Floating
X_UART0_SIN	I, internal pull-up	GPIO_1[5], I, pull-up	UART0_SIN, I, pull-up	-	-	reg58[29:28], default = 2'b00	Floating
X_UART0_SOUT	I, internal pull-up	GPIO_1[6], I, pull-down	UART0_SOUT, O, 1, pull-up	-	-	reg58[31:30], default = 2'b00	Jumper setting 7, internal pull-down

Table 2-3. Capture in Application Settings

Application	Reg. Offset	Bit	Value	Comment
MIPI + BT656 in	32'h9ba00060	[3]	1'b0	Mux MIPI PHY to the MIPI PHY mode
	scu reg7c	[23]	1'b0	Mux isp_clk to the isp mode
	scu reg54	[21:6]	16'h5555	Mux to CAP0 656 (Source pins: X_CAP0_D[0:7])
	scu reg50	[15:14]	2'b01	Mux to CAP0_CLK (Source pin: X_BAYER_CLK)
BT656 x2 in	32'h9ba00060	[3]	1'b1	Mux MIPI PHY to the parallel mode
	scu reg50	[17:16]	2'h0	Mux to CAP0_CLK (Source pin: X_MPRX_DN0), must set scu reg50{15:14] ! = 2'b01
	scu reg50	[15:14]	2'b10	-
	scu reg54	[21:6]	16'h5555	Mux to CAP0_D[0:7] (Source pins: X_CAP0_D[0:7])
	scu reg50	[15:14]	2'b10	Mux to CAP1_CLK (Source pin: X_BAYER_CLK)
	scu reg7c	[23:22]	2'b11	-
	scu reg50	[31:20]	12'haaa	Mux to CAP1_D[0:7] (Source pins: X_MPRX_CKN, X_MPRX_CKP, X_MPRX_DN1, X_MPRX_DP1, and X_BAYER_D7 ~ X_BAYER_D4)
	scu reg54	[3:0]	4'ha	-
BT1120/ CCIR601 in	32'h9ba00060	[3]	1'b1	Mux MIPI PHY to parallel mode
	scu reg50	[15:14]	2'b01	Mux to CAP0_CLK (Source pin: X_BAYER_CLK)
	scu reg50	[17:16]	2'b10	Mux to CAP0_HS (Source pin: X_MPRX_DN0)
	scu reg50	[19:18]	2'b10	Mux to CAP0_VS (Source pin: X_MPRX_DP0)
	scu reg50	[31:20]	12'h555	Mux to CAP0_D[10:15] (Source pins : X_BAYER_D6 ~ 7, X_MPRX_DP1, X_MPRX_DN1, X_MPRX_CKP, and X_MPRX_CKN)
	scu reg54	[3:0]	4'h5	Mux to CAP0_D[8:9] (Source pins: X_BAYER_D4 and X_BAYER_D5)
	scu reg54	[21:6]	16'h5555	Mux to CAP0_D[0:7] (Source pins: X_CAP0_D[0:7])

Application	Reg. Offset	Bit	Value	Comment
BAYER in	32'h9ba00060	[3]	1'b1	Mux MIPI PHY to the parallel mode
	scu reg7c	[22]	1'b1	Mux bayer_clk to the bayer mode
	scu reg7c	[23]	1'b0	Mux isp_clk to the isp mode
	scu reg50	[15:14]	2'b10	Mux to BAYER_CLK (Source pin: X_BAYER_CLK)
	scu reg50	[17:16]	2'b10	Mux to BAYER_HS (Source pin: X_MPRX_DN0)
	scu reg50	[19:18]	2'b10	Mux to BAYER_VS (Source pin: X_MPRX_DP0)
	scu reg50	[31:20]	12'haaa	Mux to BAYER_D6 ~ 11 (Source pins: X_BAYER_D6~7, X_MPRX_DP1, X_MPRX_DN1, X_MPRX_CKP, and X_MPRX_CKN)
	scu reg54	[3:0]	4'ha	Mux to BAYER_D4 ~ 5 (Source pins: X_BAYER_D4 and X_BAYER_D5)
	scu reg54	[13:6]	8'haa	Mux to BAYER_D0 ~ 3 (Source pins: X_CAP0_D[4:7])

Table 2-4. Jumper Settings

Function	jpset	Pin/Ball Name	Description	Default Value
BUS frequency	jpset[1:0]	X_SSP1_TXD, X_CAP_RST	CPU/Bus/DDR3 output frequency setting 00: DDR_CLK-400MHz AHB-133MHz AXI-133MHz AXI2-200MHz CPU-400MHz 01: DDR_CLK-540MHz AHB-200MHz AXI-200MHz AXI2-270MHz CPU-810MHz 10: DDR_CLK-500MHz AHB-200MHz AXI-200MHz AXI2-250MHz CPU-500MHz 11: DDR_CLK-540MHz AHB-200MHz AXI-200MHz AXI2-270MHz CPU-540MHz	0x00
-	jpset[5:2]	-	Reserved Don't use	-
Firmware	jpset[6]	X_SPI_SCLK	1: Firmware update managed by ROM	0x0
SPI type select	jpset[7]	X_UART0_SOUT	SPI Flash type selection 0: SPI NOR Flash 1: SPI NAND Flash	0x0
M40c auto wdr	jpset[8]	X_UART2_SOUT	M40c auto WDR (WatchDog Run) 0: Auto WDR 1: Non-auto WDR	0x1
-	jpset[12:9]	-	Reserved	-
-	jpset[13]	X_OM	Reserved Don't use	0x0

Function	jpset	Pin/Ball Name	Description	Default Value
-	jpset[18:14]	-	Reserved	-
SPI type select	jpset[19]	X_SD_CLK	SPI Flash type selection 0: 3-byte SPI Flash device 1: 4-byte SPI Flash device	0x0
-	jpset[24:20]	-	Reserved	-

2.2 Pin Assignments

Figure 2-1through Figure 2-4 show the pin assignments of GM8136S/GM8135S.

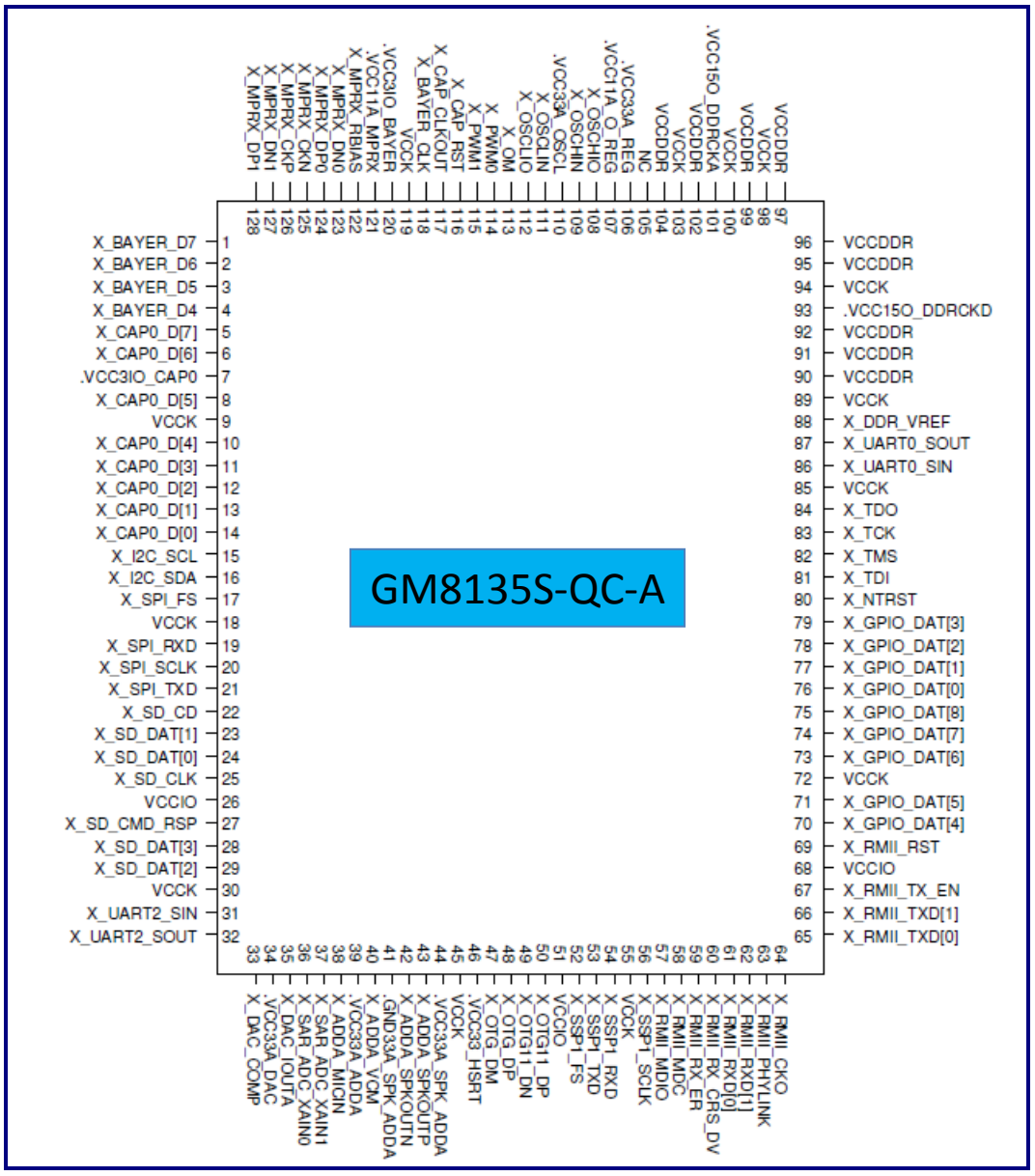


Figure 2-1. Pin Assignments of LQFP (LQFP128) of GM8135S-QC-A (Top View)

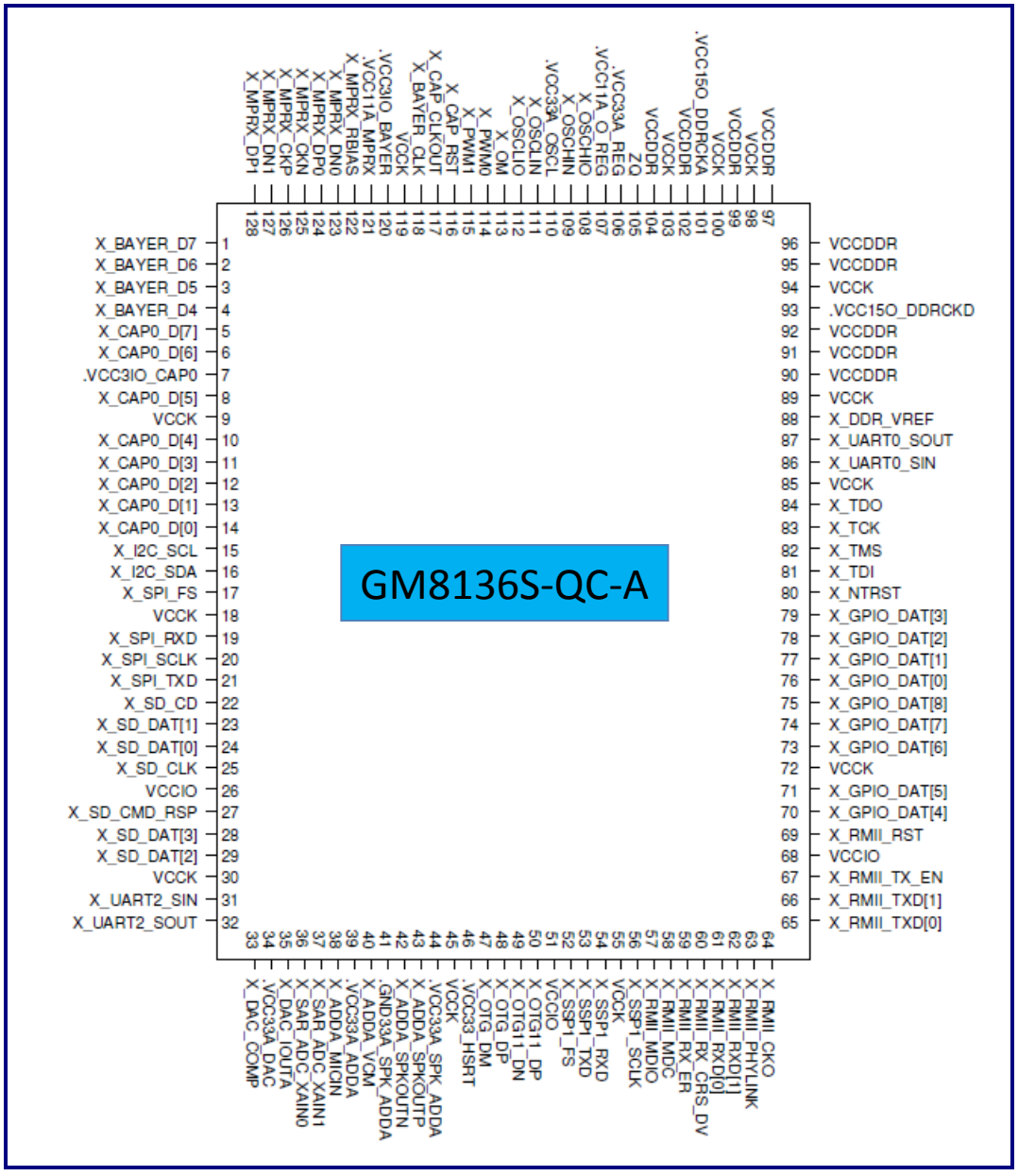


Figure 2-2. Pin Assignments of LQFP (LQFP128) of GM8136S-QC-A (Top View)

	1	2	3	4	5	6	7
A	X_MPRX_DN1	X_MPRX_CKN	X_MPRX_DN0	X_BAYER_CLK	X_CAP_RST	X_MPRX_RBIAS	X_PWM1
B	X_MPRX_DP1	X_MPRX_CKP	X_MPRX_DP0	X_CAP_CLKOUT	VCC3IO_BAYER	VCC11A_MPRX	X_PWM0
C	X_BAYER_D7	X_BAYER_D6	GND	GND	VCC3IO_CAP0	VCCK	VCCK
D	X_BAYER_D5	X_BAYER_D4	GND	GND	VCCIO	VCCK	VCCK
E	X_CAP0_D[7]	X_CAP0_D[6]	GND	GND	VCCIO	GND	X_OM
F	X_CAP0_D[5]	X_CAP0_D[4]	GND	GND	VCCIO	GND	GND
G	X_CAP0_D[3]	X_CAP0_D[2]	GND	GND	VCCIO	GND	GND
H	X_CAP0_D[1]	X_CAP0_D[0]	GND	GND	VCCIO	GND	GND
J	X_I2C_SCL	X_I2C_SDA	GND	GND	VCCIO	GND	GND
K	X_SPI_FS	X_SPI_RXD	GND	GND	VCCIO	GND33A_ADDA	GND
L	X_SPI_SCLK	X_SPI_TXD	X_UART2_SIN	GND	VCC33A_DAC	GND33A_SPK_ADDA	GND
M	X_SD_CD	X_SD_DAT[1]	X_UART2_SOUT	X_SAR_ADC_XAIN1	VCC33A_ADDA	VCC33A_SPK_ADDA	VCC33_HSRT
N	X_SD_DAT[0]	X_SD_CMD_RSP	X_SD_DAT[2]	X_SAR_ADC_XAIN0	X_ADDA_MICIN	X_ADDA_SPKOUTP	VCCIO
P	X_SD_CLK	X_SD_DAT[3]	X_DAC_COMP	X_DAC_IOUTA	X_ADDA_VCM	X_ADDA_SPKOUTN	VCCIO
	1	2	3	4	5	6	7

8	9	10	11	12	13	14	
X_OSCLIN	X_OSCHIO	VCC33A_OSCL	VCC33A_REG	VCC11A_O_REG	VCCK	VCCK	A
X_OSCLIO	X_OSCHIN	GND	GND33A_REG	VCCK	VCCK	VCCK	B
VCCK	VCCK	VCCK	VCCK	VCCK	VCC15O_DDRCKA	VCCDDR	C
VCCK	VCCK	VCCK	VCCK	VCCDDR	VCC15O_DDRCKD	VCCDDR	D
GND	GND	GND	VCCDDR	VCCDDR	**ZQ**	**X_DDR_VREF**	E
GND	GND	VCCDDR	VCCDDR	X_TDO	X_UART0_SOUT	X_UART0_SIN	F
GND	GND	VCCDDR	X_TCK	X_TMS	X_GPIO_DAT[3]	X_GPIO_DAT[2]	G
GND	GND	GND	X_TDI	X_NTRST	X_GPIO_DAT[1]	X_GPIO_DAT[0]	H
GND	GND	GND	GND	X_RSTN	X_TV_PCLK	X_GPIO_DAT[7]	J
GND	GND	GND	GND	GND	X_GPIO_DAT[6]	X_GPIO_DAT[5]	K
GND	GND	GND	GND	GND	X_GPIO_DAT[4]	X_RMII_RST	L
GND	GND	X_SSP1_TXD	X_SSP1_RXD	X_SSP1_SCLK	X_RMII_TXD[1]	X_RMII_TX_EN	M
X_OTG_DP	X_OTG11_DP	X_SSP1_FS	X_RMII_RX_ER	X_RMII_RXD[0]	X_RMII_PHYLINK	X_RMII_TXD[0]	N
X_OTG_DM	X_OTG11_DN	X_RMII_MDIO	X_RMII_MDC	X_RMII_RX_CRS_DV	X_RMII_RXD[1]	X_RMII_CKO	P
8	9	10	11	12	13	14	

Figure 2-3. Pin Assignments of Ball Grid Array (TFBGA196) of GM8136S-BC-A (Top View)

	1	2	3	4	5	6	7
A	X_MPRX_DN1	X_MPRX_CKN	X_MPRX_DN0	X_BAYER_CLK	X_CAP_RST	X_MPRX_RBIAS	X_PWM1
B	X_MPRX_DP1	X_MPRX_CKP	X_MPRX_DP0	X_CAP_CLKOUT	VCC3IO_BAYER	VCC11A_MPRX	X_PWM0
C	X_BAYER_D7	X_BAYER_D6	GND	GND	VCC3IO_CAP0	VCCK	VCCK
D	X_BAYER_D5	X_BAYER_D4	GND	GND	VCCIO	VCCK	VCCK
E	X_CAP0_D[7]	X_CAP0_D[6]	GND	GND	VCCIO	GND	X_OM
F	X_CAP0_D[5]	X_CAP0_D[4]	GND	GND	VCCIO	GND	GND
G	X_CAP0_D[3]	X_CAP0_D[2]	GND	GND	VCCIO	GND	GND
H	X_CAP0_D[1]	X_CAP0_D[0]	GND	GND	VCCIO	GND	GND
J	X_I2C_SCL	X_I2C_SDA	GND	GND	VCCIO	GND	GND
K	X_SPI_FS	X_SPI_RXD	GND	GND	VCCIO	GND33A_ADDA	GND
L	X_SPI_SCLK	X_SPI_TXD	X_UART2_SIN	GND	VCC33A_DAC	GND33A_SPK_ADDA	GND
M	X_SD_CD	X_SD_DAT[1]	X_UART2_SOUT	X_SAR_ADC_XAIN1	VCC33A_ADDA	VCC33A_SPK_ADDA	VCC33_HSRT
N	X_SD_DAT[0]	X_SD_CMD_RSP	X_SD_DAT[2]	X_SAR_ADC_XAIN0	X_ADDA_MICIN	X_ADDA_SPKOUTP	VCCIO
P	X_SD_CLK	X_SD_DAT[3]	X_DAC_COMP	X_DAC_IOUTA	X_ADDA_VCM	X_ADDA_SPKOUTN	VCCIO
	1	2	3	4	5	6	7

8	9	10	11	12	13	14	
X_OSCLIN	X_OSCHIO	VCC33A_OSCL	VCC33A_REG	VCC11A_O_REG	VCCK	VCCK	A
X_OSCLIO	X_OSCHIN	GND	GND33A_REG	VCCK	VCCK	VCCK	B
VCCK	VCCK	VCCK	VCCK	VCCK	VCC15O_DDRCKA	VCCDDR	C
VCCK	VCCK	VCCK	VCCK	VCCDDR	VCC15O_DDRCKD	VCCDDR	D
GND	GND	GND	VCCDDR	VCCDDR	**NC**	**X_DDR_VREF**	E
GND	GND	VCCDDR	VCCDDR	X_TDO	X_UART0_SOUT	X_UART0_SIN	F
GND	GND	VCCDDR	X_TCK	X_TMS	X_GPIO_DAT[3]	X_GPIO_DAT[2]	G
GND	GND	GND	X_TDI	X_NTRST	X_GPIO_DAT[1]	X_GPIO_DAT[0]	H
GND	GND	GND	GND	X_RSTN	X_TV_PCLK	X_GPIO_DAT[7]	J
GND	GND	GND	GND	GND	X_GPIO_DAT[6]	X_GPIO_DAT[5]	K
GND	GND	GND	GND	GND	X_GPIO_DAT[4]	X_RMII_RST	L
GND	GND	X_SSP1_TXD	X_SSP1_RXD	X_SSP1_SCLK	X_RMII_TXD[1]	X_RMII_TX_EN	M
X_OTG_DP	X_OTG11_DP	X_SSP1_FS	X_RMII_RX_ER	X_RMII_RXD[0]	X_RMII_PHYLINK	X_RMII_TXD[0]	N
X_OTG_DM	X_OTG11_DN	X_RMII_MDIO	X_RMII_MDC	X_RMII_RX_CRS_DV	X_RMII_RXD[1]	X_RMII_CKO	P
8	9	10	11	12	13	14	

Figure 2-4. Pin Assignments of Ball Grid Array (TFBGA196) of GM8135S-BC-A (Top View)

Chapter 3

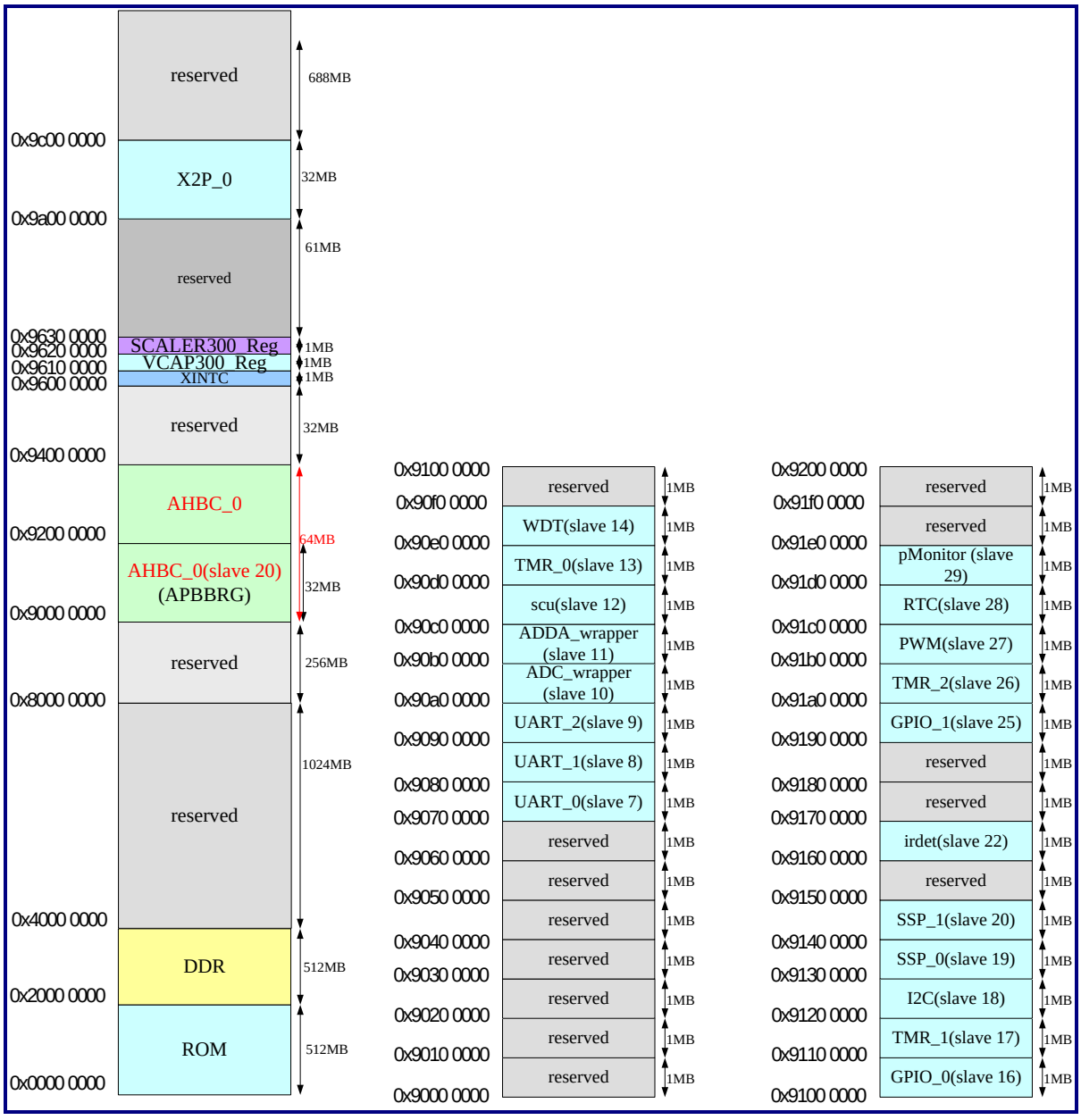
Memory Map

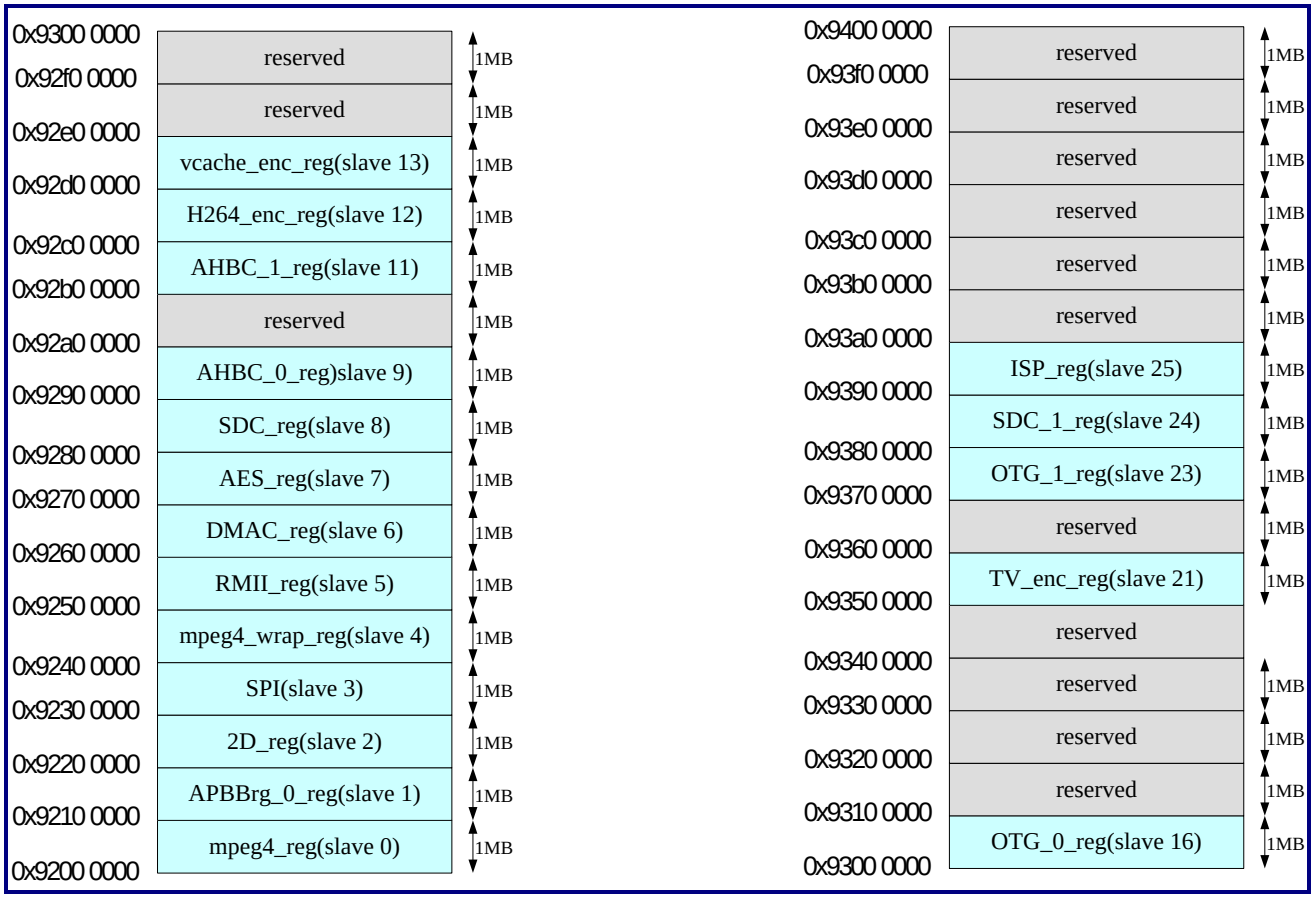
This chapter contains the following section:

- 3.1 Address Map

3.1 Address Map

GM8136S/GM8135S with a 32bit address bus is capable of addressing a 4GB physical address space. This space is divided into the section regions within which the various memories and peripherals are mapped. Figure 3-1 shows the detailed address map of GM8136S/GM8135S.




0x9300 0000
reserved
1MB
0x92f0 0000
reserved
1MB
0x92e0 0000
vcache_enc_reg(slave 13)
1MB
0x92d0 0000
H264_enc_reg(slave 12)
1MB
0x92c0 0000
AHBC_1_reg(slave 11)
1MB
0x92b0 0000
reserved
1MB
0x92a0 0000
AHBC_0_reg)slave 9)
1MB
0x9290 0000
SDC_reg(slave 8)
1MB
0x9280 0000
AES_reg(slave 7)
1MB
0x9270 0000
DMAC_reg(slave 6)
1MB
0x9260 0000
RMII_reg(slave 5)
1MB
0x9250 0000
mpeg4_wrap_reg(slave 4)
1MB
0x9240 0000
SPI(slave 3)
1MB
0x9230 0000
2D_reg(slave 2)
1MB
0x9220 0000
APBBrg_0_reg(slave 1)
1MB
0x9210 0000
mpeg4_reg(slave 0)
1MB
0x9200 0000
0x9400 0000
reserved
1MB
0x93f0 0000
reserved
1MB
0x93e0 0000
reserved
1MB
0x93d0 0000
reserved
1MB
0x93c0 0000
reserved
1MB
0x93b0 0000
reserved
1MB
0x93a0 0000
ISP_reg(slave 25)
1MB
0x9390 0000
SDC_1_reg(slave 24)
1MB
0x9380 0000
OTG_1_reg(slave 23)
1MB
0x9370 0000
reserved
1MB
0x9360 0000
TV_enc_reg(slave 21)
1MB
0x9350 0000
reserved
0x9340 0000
reserved
1MB
0x9330 0000
reserved
1MB
0x9320 0000
reserved
1MB
0x9310 0000
OTG_0_reg(slave 16)
1MB
0x9300 0000

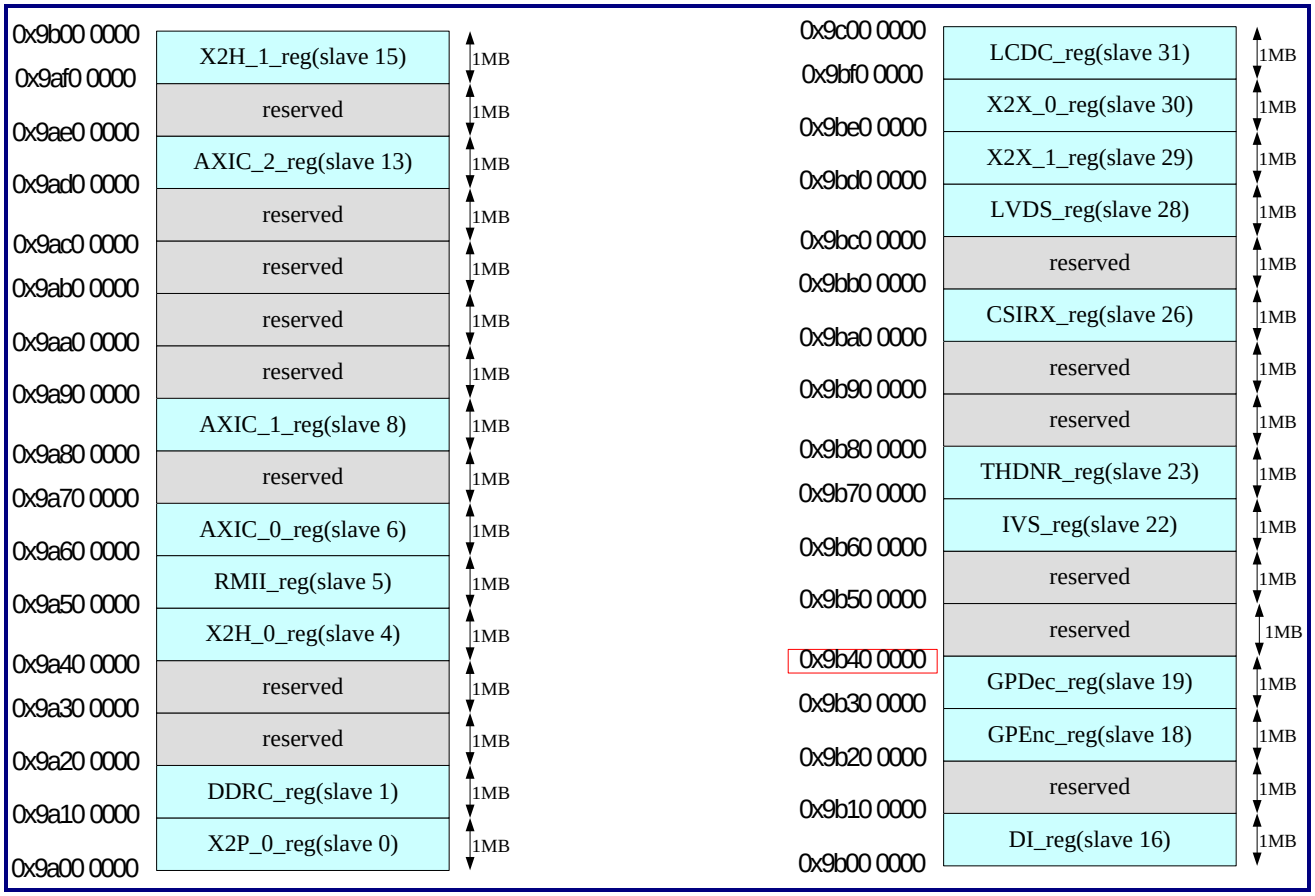


Figure 3-1. Physical Address Map

Chapter 4

System Control Unit

This chapter contains the following sections:

4.1 System Controller

4.1.1 General Description

System Control Unit (SCU) provides the fixed clocks for each peripheral unit. Many peripheral clocks for the devices can be disabled by using the AXI Module Clock Off Control Register (AXIMCLKOFF), the AHB Module Clock Off Control Register (AHBMCLKOFF) or APB Module Clock Off Control Registers (APBMCLKOFF0 and APBMCLKOFF1), or through the bits in the control registers of the peripheral. When the clock is not used by any unit this clock will be turned off to minimize the power consumption. SCU also provides a method to change the PLL frequency.

4.1.2 Features

- Supports hardware reset (Asserted X_RSTN) for non-maskable resets
- Supports watchdog reset to assert WDT for resetting system except for SCU units, and used as code monitor
- Supports normal mode to enable all power supplies and clocks
- Supports FCS mode to change PLL1/PLL2 core settings for operating at various system frequencies

4.1.3 Clock Manager

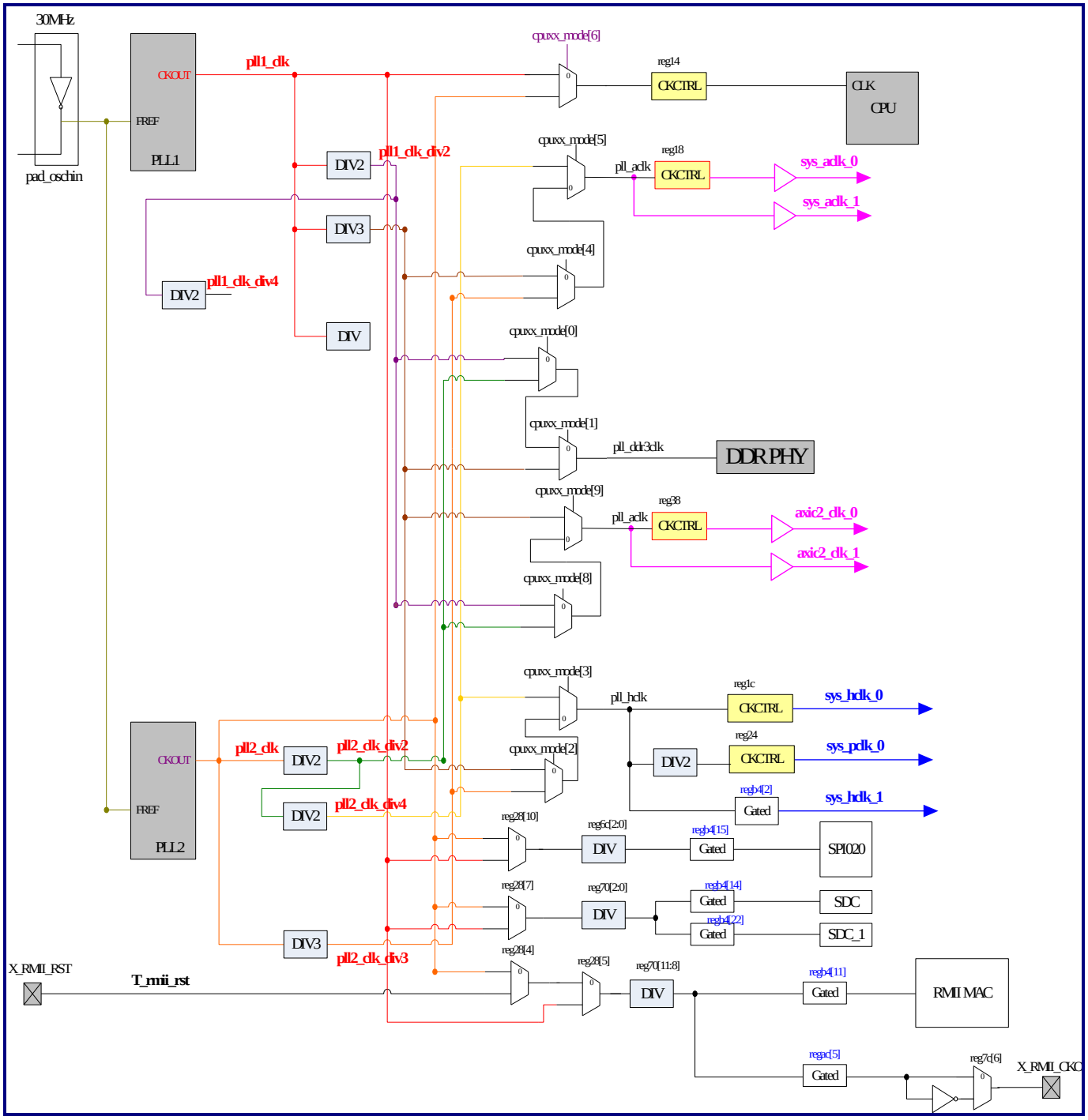
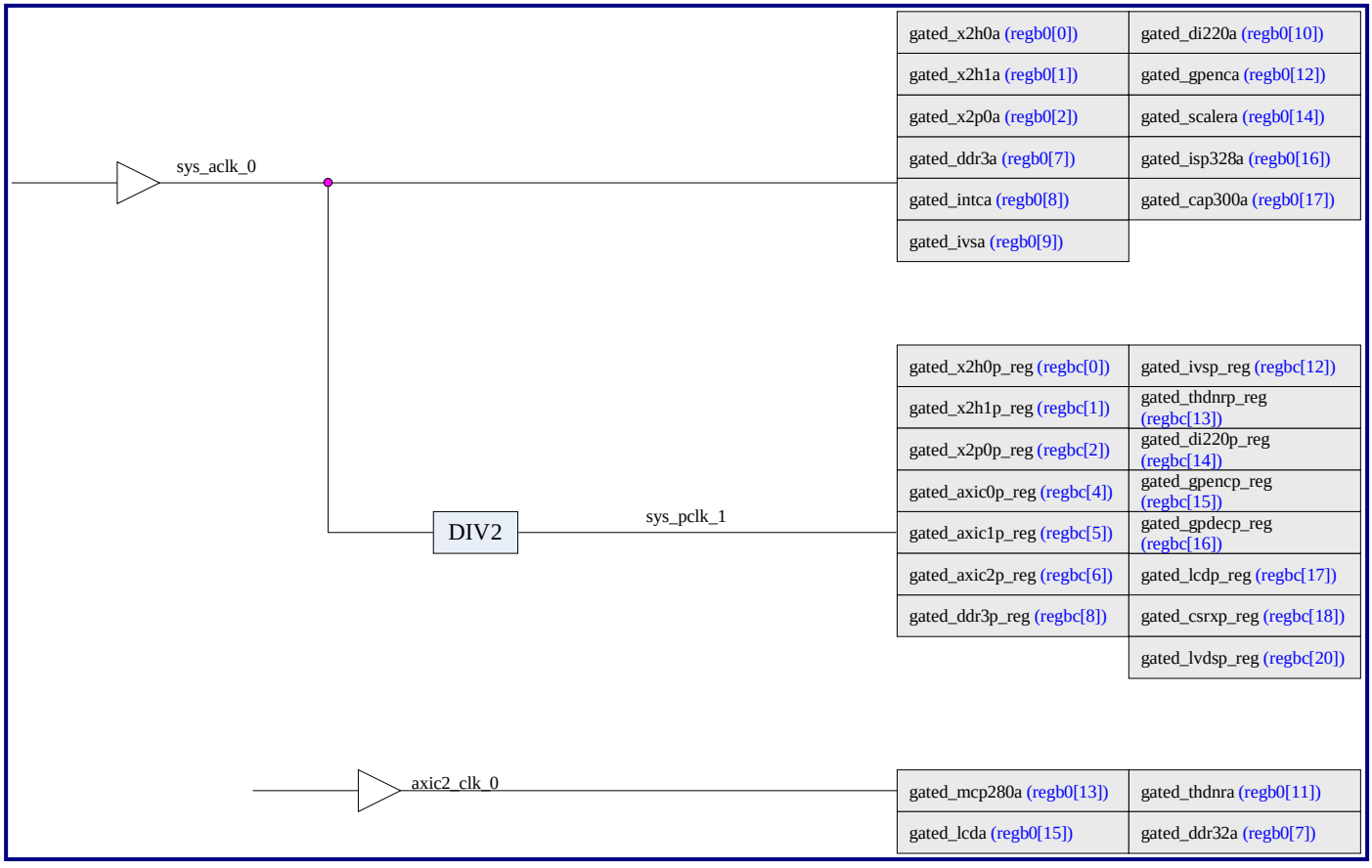


Figure 4-1. SCU Block Diagram for PLL1 and PLL2



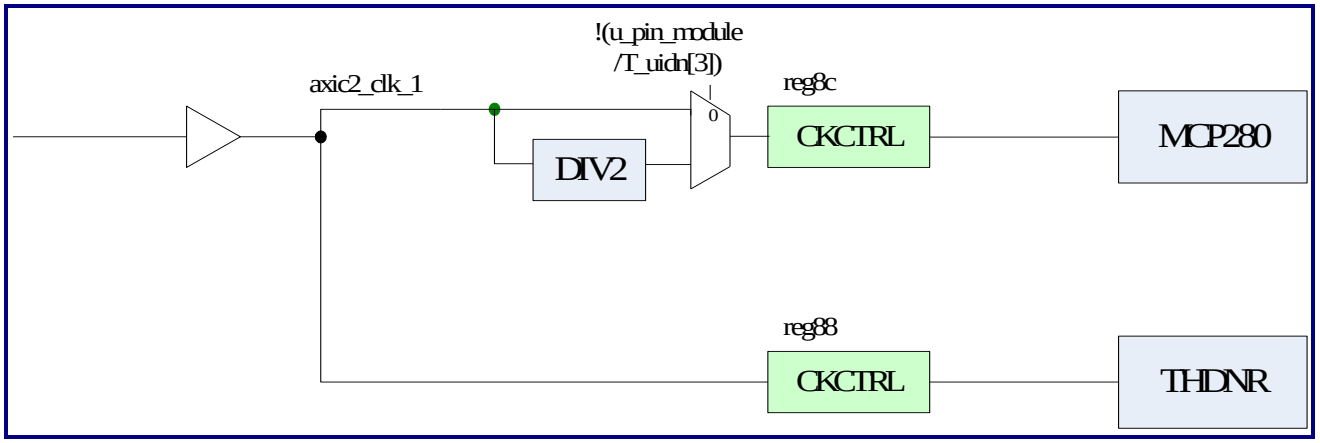


Figure 4-2. SCU Block Diagram for AXI/AHB/APB Bus (1)

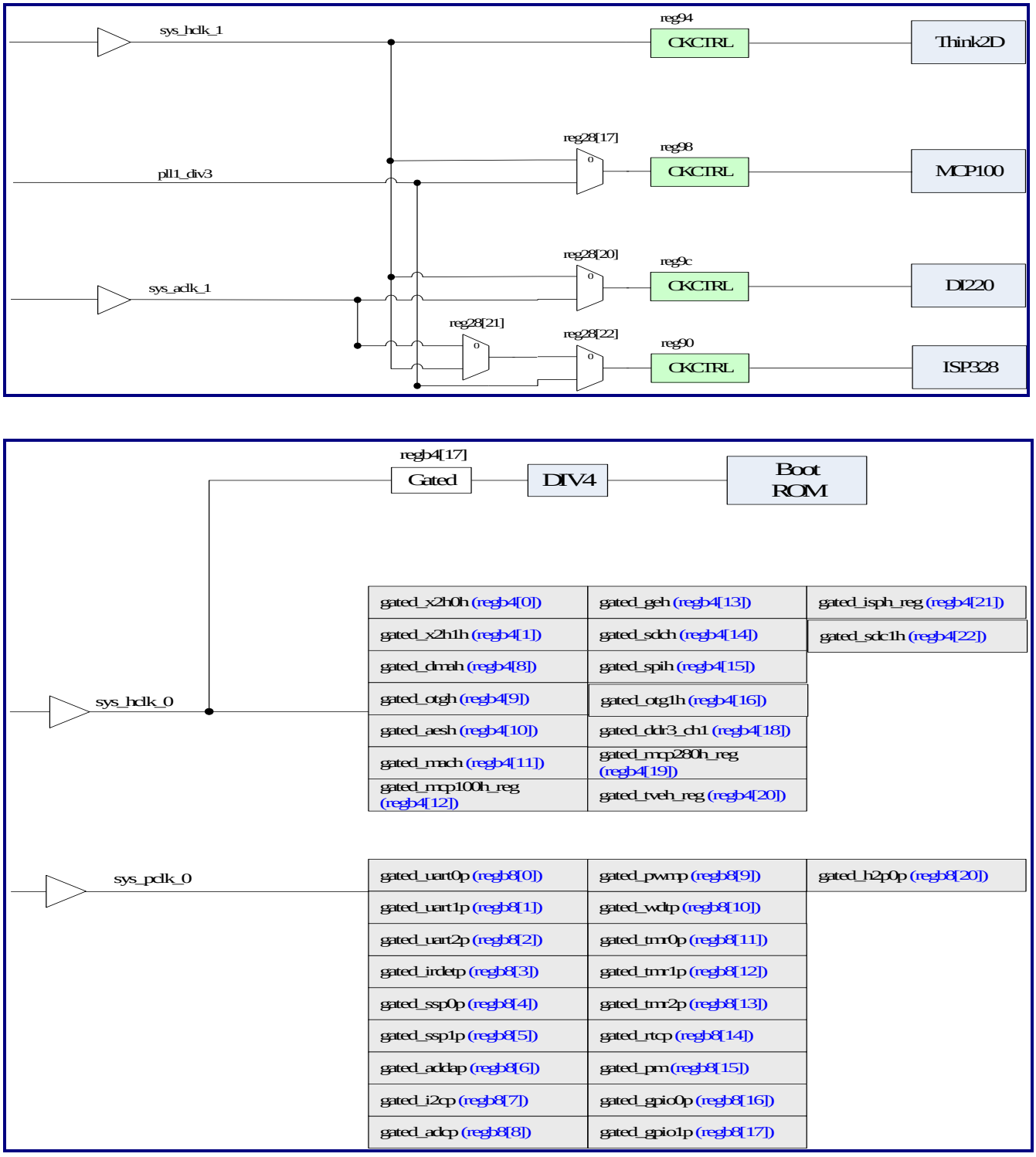


Figure 4-3. SCU Block Diagram for AXI/AHB/APB Bus (2)

GM8136S/GM8135S clocking system incorporates the following clock sources:

- 30MHz oscillator
- Programmable frequency of the core PLL

4.1.3.1 Core PLL (PLL1, PLL2, and PLL3)

Core PLL uses the 30MHz oscillator as a reference and multiplies its frequency by programming the PLLxNS[6:0] and PPLxMS[4:0] bits in the PLLxCR register. The following equations show the calculations of the core PLL frequency output:

- PLL frequency (CKOUT) = (NS/MS) * 30MHz, 500MHz ≤ freq (CKOUT) ≤ 1000MHz for frang = '11'
- PLL frequency (CKOUT) = (NS/MS) * 30MHz/2, 250MHz ≤ freq (CKOUT) ≤ 500MHz for frang = '10'
- PLL frequency (CKOUT) = (NS/MS) * 30MHz/4, 125MHz ≤ freq (CKOUT) ≤ 250MHz for frang = '01'
- PLL frequency (CKOUT) = (NS/MS) * 30MHz/8, 62.5MHz ≤ freq (CKOUT) ≤ 125MHz for frang = '00'

Note: "x" can be 1, 2, and 3.

PLL1 and PLL2 are the clock sources of CPU, DDR3 controller, AXI/AXI_2 devices, AHB bus devices, APB bus devices, audio devices, SD card, and SPI devices. The AXI_2 bus clock (ACLK_2) and DDR3 PHY input clock come from PLL1 CKOUT divided by 2 or 3 or PLL2 CKOUT divided by 2. The AXI bus clock (ACLK) and AHB bus clock (HCLK) come from PLL1 CKOUT divided by 3 or PLL2 CKOUT divided by 3 or 4. The APB0 bus clock (PCLK) is the AHB bus frequency divided by 2. The APB1 bus clock (PCLK_1) is the AXI bus frequency divided by 2. The CPU input clock comes from PLL1 CKOUT divided by 1 or PLL2 CKOUT divided by 1. PLL3 generates the clock sources to be used by the audio devices, sensor, and display pixel clock.

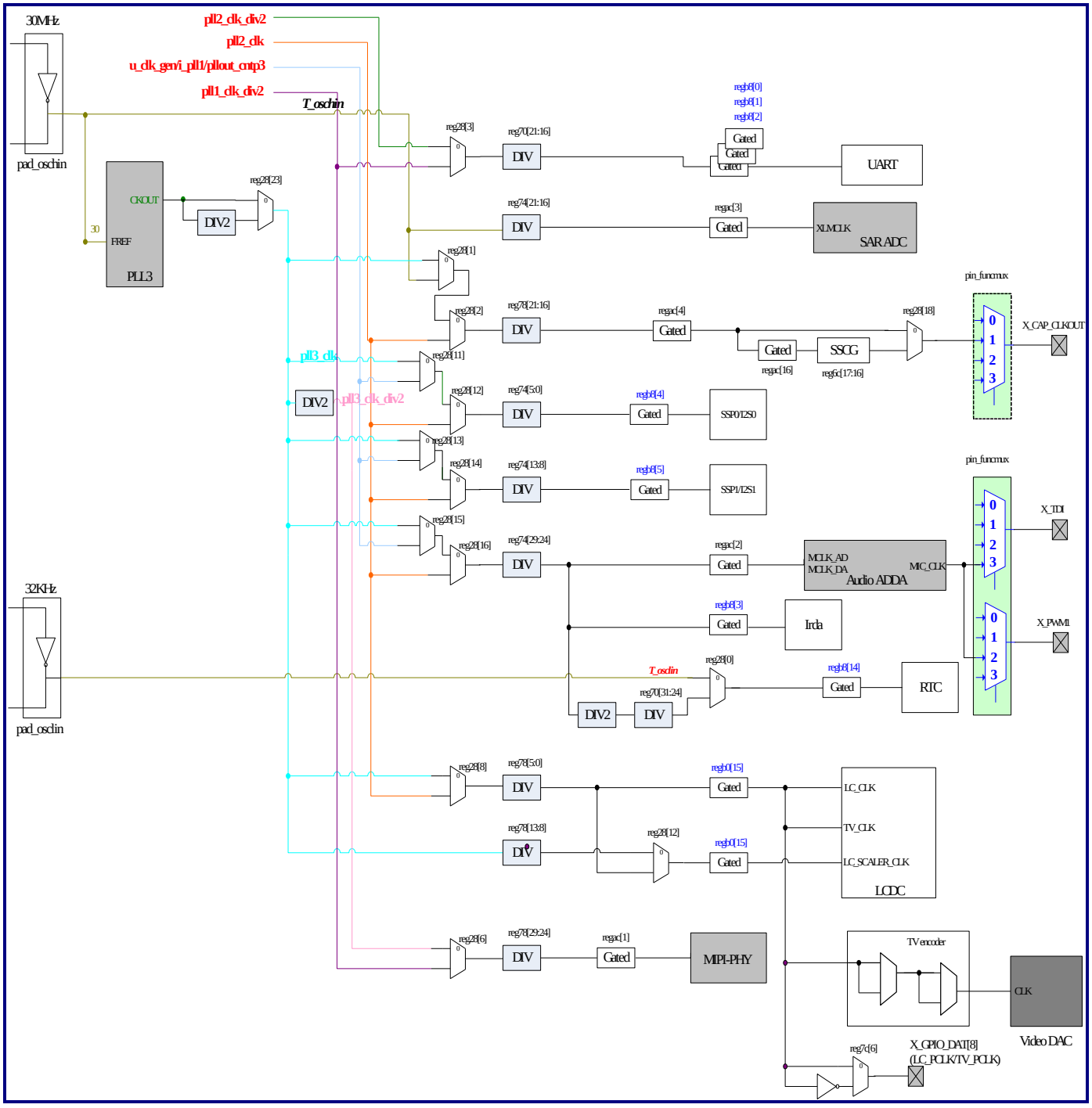


Figure 4-4. SCU Block Diagram for PLL3

4.1.3.2 Clock Gating

SCU includes the AXIMCLKOFF, AHBMCLKOFF, APBMCLKOFF0, APBMCLKOFF1, and HARDCOREOFF registers. These registers contain the configuration bits that can disable the clocks to individual units. The configuration bits will be used when a module is not used. After the hardware reset, the related clock of any module that is not used must be disabled.

4.1.3.3 Hardware Reset

The hardware reset will be invoked when the X_RSTN pin is pulled low by an external source. GM8136S/GM8135S does not provide the method for masking or disabling the propagation of the external pins. When the X_RSTN pin is asserted, the hardware reset will be invoked regardless of the operating mode.

4.1.3.4 Normal Mode

In the normal mode, all enabled power supplies and all functionally enabled clocks are running. The normal mode can be active subsequent to any power mode, power sequence, or after a reset completes its sequence.

- Entering condition: (Chip initial → Normal mode)
  The normal mode will be active after the system power-on or after a reset is complete.
- Exiting condition: (Normal mode → IDLE mode, Normal mode → Turbo mode)
  The normal mode will be exited when the IDLE mode is executed, or when a reset begins. For the detailed sequence, please refer to the IDLE mode.

State definition: The normal mode is the normal operating mode of the application processor. In the normal mode, software can use the AXIMCLKOFF (Offset = 0xb0)/AHBMCLKOFF (Offset = 0xb4)/ APBMCLKOFF0 (Offset = 0xb8)/APBMCLKOFF1 (Offset = 0xbc) registers to turn on or turn off the related module clock.

4.1.3.5 IDLE Mode

- Entering condition: (Normal mode → IDLE mode)
  Software writes the coprocessor instruction to CPU. __asm MCR P15, 0, temp, c7, c0, 4.

- Exiting condition: (IDLE mode → Normal mode)
  The IDLE mode exits when any reset or IRQ/FIQ interrupt is asserted by INTC.
- State definition (In the IDLE mode):
  - CPU clocks stop.
  - The remainder of the application processor operates normally. For example, the LCD controller can continue refreshing the screen with the same frame buffer data in the memory.
  - Interrupts are recognized as wake-up sources.

4.1.3.6 Frequency Change Sequence (FCS)

The Frequency Change Sequence (FCS) is used to change the system clock frequency. The system clocks stop during the course of FCS. In this sequence, the frequency is changed from the default condition at initial boot-up. FCS can be used as a power-saving feature that allows GM8136S/GM8135S to operate at the minimum required frequency.

Entering/Exiting Sequence: (S → Software action, H → Hardware action)

1. Initiate the interrupt controller (S)
2. Disable the AXI/AHB master peripherals to stop the events from coming into the system (S)
3. Program a suitable value to the refresh interval counter of the DDR3/DDR2 controller (S)
4. Write a new frequency value to PLL1CR[pll1ms, pll1ns, cpuxx_mode] (Offset = 0x30) (S)
5. Write a new frequency value to PLL2CR[pll2ms, pll2ns] (Offset = 0x34) (S)
6. Set the FCS bit in the PMODE register (Offset = 0x0c) (S)
7. Block the interrupts to CPU (H)
8. Write the CPU power-down command (S)
9. Wait for the idle signal of CPU, CPUOFFN (H)
10. Issue a self-refresh command to the DDR3/DDR2 SDRAM controller
11. Wait for acknowledge of self-refresh (H)
12. Gate the clocks (pclk, hclk, aclk, then fclk) (H)
13. Load the new frequency value to PLL (H)

14. Wait for the stable signal of PLL (H)
15. Run the clocks free (fclk, then aclk, hclk, and pclk) (H)
16. Remove gating of the master requests/interrupts (H)
17. Clock SCU for issuing an interrupt to CPU to enter the interrupt subroutine (H)
18. Software must examine the PMSR Register (PMSR[IntFCS]) (Offset = 0x20) to determine the cause for waking up. Software must clear the FCS bit in the PMODE register (Offset = 0x0c). (S)

State definition: FCS (Frequency Change Sequence) is implemented to change the PLL1 and PLL2 clock frequencies. During the course of FCS, the PLL1 and PLL2 clocks stop. This mode is used to change PLL1 and PLL2 from the default condition under the initial boot-up condition. Software must complete the following steps before initiating FCS:

1. Configure the memory controller to ensure that the content of DDR3/DDR2 is maintained during FCS. The refresh timer of the memory controller must be programmed to match the maximum refresh time associated with the slower of two frequencies (Current frequency and desired frequency).
2. Disable the LCD controller or configure this controller to avoid the effects of an interruption in the LCD clocks and data from the application processor.
3. The interrupts generated during FCS will be ignored.
4. In FCS, the PLL1 and PLL2 clock generators are locking to the correct frequency and cannot be used.

4.1.4 Programming Model

4.1.4.1 Summary of Clocks and Power Manager Registers

Table 4-1 provides a summary of the SCU registers.

Table 4-1. Summary of SCU Registers

Offset	Name	Reset Value
0x00	ID number	0x0813 6100
0x04	Jump setting	It depends on the jumper setting.
0x0C	Power mode	0x0100 0000
0x10	Power manager control	0x0000 0000
0x14	fclk clock control	0x0000 0000

Offset	Name	Reset Value
0x18	sys_aclk_0 clock control	0x0000 0000
0x1C	sys_hclk_0 clock control	0x0000 0000
0x20	Power manager status	0x0000 0400
0x24	sys_pclk_0 clock control	0x0000 0000
0x28	IP main clock select setting	0x0000 0001
0x2c	PLL1, 2 and 3 bandwidth option	0x0000 002a
0x30	PLL1 control and cpuxx_mode setting	0x---- ----
0x34	PLL2/PLL3 control	0x1B6F ----
0x38	axic_2 clock control	0x0000 0000
0x40	Driving capacity and slew rate and hold time control 0	0x0000 0000
0x44	Driving capacity and slew rate and hold time control 1	0x0000 0000
0x48	Bus bridge control	0x0000 1FFF
0x50	MFPSR0 (Multi-function pin)	0x0000 0000
0x54	MFPSR1 (Multi-function pin)	0x0000 0000
0x58	MFPSR2 (Multi-function pin)	0x0000 0000
0x5C	MFPSR3 (Multi-function pin)	0x0000 0000
0x60	MFPSR4 (Multi-function pin)	0x0000 0000
0x64	MFPSR5 (Multi-function pin)	0x0000 0000
0x6C	SPI x-bit count	0x0000 0004
0x70	SDC/MAC/UART/RTC x-bit counter	0xB609 0144
0x74	SSP/ADC/ADDA x-bit counter	0x2C04 2C2C
0x78	EXT/Panel pixel/LCDC scalar x-bit counter	0x0913 1313
0x7C	System control	0x8000 0000
0x80	DDR3/DDR2 PHY power control	0x0000 00FF
0x88	THDNR (3D noise reduction) clock control	0x0000 0000
0x8C	H264 encoder clock control	0x0000 0000
0x90	ISP (Image Signal Processing) clock control	0x0000 0000
0x94	2D-graphic engine clock control	0x0000 0000
0x98	MPEG 4/JPEG main clock control	0x0000 0000
0x9C	De-interlace main clock control	0x0000 0000
0xA0	IP software reset control register 0	0xFB7F FFFF
0xA4	IP software reset control register 1	0xFFFF FFFF
0xA8	Software interrupt set/clear control	0x0000 0000
0xAC	HARDCORECLKOFF	0xFFFF FFFF

Offset	Name	Reset Value
0xB0	AXIMCLKOFF	0xFFFF FF00
0xB4	AHBMCLKOFF	0xFFFD F7F0
0xB8	APBMCLKOFF0	0xFFEF FFFF
0xBC	APBMCLKOFF1	0xFFFF F000
0xC0	OTG 0 (OTG 2.0)/OTG 1(OTG 1.1) PHY control	0x0308 301C
0xC4	OTG 0 (OTG 2.0) PHY control	0x0000 0000
0xd0 ~ 0xdc	Software register	0x0000 0000

4.1.4.2 Register Descriptions

The following subsections describe the SCU registers in details.

The abbreviations below represent the access types used throughout the register descriptions:

- R/W: Read/Write
- RO: Read Only
- IO trimming
- HR: Hardware Reset

4.1.4.2.1 ID Number (IDNMBR, Offset = 0x00)

Table 4-2. ID Number (IDNMBR, Offset = 0x00)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:28]	-	-	reserved	-	-
[27:12]	product ID	RO	Product ID	0x8136	-
[11:8]	version ID	RO	Version ID	0x1	-
[7:0]	ID [7:0]	RO	ID, [6:2] – package ID	I/O trimming	-

4.1.4.2.2 Jump Setting (Offset = 0x04)

Software can read the jump setting status via this register.

Table 4-3. Jump Setting (Offset = 0x04)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:28]	-	-	Reserved	-	-
27	jpset[24]	RO	Reserved	-	-
26	jpset[23]	RO	Reserved	-	-
25	jpset[22]	RO	Reserved	-	-
24	jpset[20]	RO	Reserved	-	-
23	jpset[19]	RO	X_SD_CLK 3/4byte type selection of the SPI Flash 0: 3byte type (Pin internal pull-down) 1: 4byte type	-	-
22	jpset[18]	RO	Reserved	-	-
21	jpset[17]	RO	Reserved	-	-
20	-	-	Reserved	-	-
19	-	-	Reserved	-	-
18	-	-	Reserved	-	-
17	-	-	Reserved	-	-
[16:14]	jpset[16:14]	RO	Reserved	-	-
13	jpset[13]	RO	T_om DIS_OSC_CNT 0: Enable (Default) 1: Disable	-	-
[12:9]	jpset[12:9]	RO	Reserved	-	-
8	Jpset[8]	RO	X_UART2_SOUT m40c auto WDR (WatchDog Run) 1: Non-auto WDR (Pin internal pull-up) 0: Auto WDR		
7	jpset[7]	RO	X_UART0_SOUT NAND/NOR SPI Flash type selection 0: NOR SPI Flash (Pin internal pull-down) 1: NAND SPI Flash	-	-

Bit	Name	Type	Description	Reset Value	Reset Type
6	jpset[6]	RO	X_SPI_SCLK FW upgrade 0: Enable 1: Disable	-	-
[5:2]	jpset[5:2]	RO	Reserved	-	-
[1:0]	jpset[1:0]	RO	{X_SSP1_TXD, X_CAP_RST} PLL setting 00: 400(DDR_CLK)/133(AHB)/133(AXI)/ 400(CPU)/200(AXI_2) {pin internal pull-down, internal pull-down} 01: 540(DDR_CLK)/200(AHB)/200(AXI)/ 810(CPU)/270(AXI_2) 10: 500(DDR_CLK)/200(AHB)/200(AXI)/ 500(CPU)/250(AXI_2) 11: 540(DDR_CLK)/200(AHB)/200(AXI)/ 540(CPU)/270(AXI_2)	-	-

4.1.4.2.3 Power Mode Register (PMR, Offset = 0x0C)

Table 4-4. Power Mode Register (PMR, Offset = 0x0C)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:24]	cpu idle counter	R/W	The wait time is {[31:24], 2'b0} * osch_period. The default value is 8'h1.	0x1	HR
[23:22]	-	-	Reserved	-	-
21	cpuidle_sel	R/W	1: Select cpuoffn 0: Select the down-counting timer	0x0	HR
[20:9]	-	-	Reserved	-	-
8	clr_cpu_hold	R/W	Clear cpu_hold	0x0	HR
[7:6]	-	-	Reserved	-	-
5	cpuclkoffdis	-	1: CPU clock is off. 0: CPU clock is on.	0x0	HR
4	oschoffdis	R/W	Disable OSCH when entering the standby mode 1: OSCH clock is off.	0x0	HR
3	turbo	R/W	Turbo mode	0x0	HR

Bit	Name	Type	Description	Reset Value	Reset Type
2	fcs	R/W	Frequency Change Sequence 1: Enter Frequency Change Sequence When changing the frequency, users must write the CPU coprocessor instruction (Power-down command) after the power-mode instruction. 0: Do not enter Frequency Change Sequence	0x0	HR
[1:0]	mode	R/W	Enter the low-power mode when CPU executes the WFI (Wait For Interrupt) instruction 0: No effect 1: Idle mode (CPU clock off) 2: Standby mode (All clocks off, PLL disabled, and DRAM in the self-refresh mode)	0x0	HR

4.1.4.2.4 Power Manager Control Register (PMCR, Offset = 0x10)

Table 4-5. Power Manager Control Register (PMCR, Offset = 0x10)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:17]	-	-	Reserved	-	-
16	ls_pm_wertc	R/W	RTC Weakup Enable Write set/Clear pulse Others: 0	0x0	HR
[15:0]	-	-	Reserved	-	-

4.1.4.2.5 fclk Clock Control Register (Offset = 0x14)

Table 4-6. fclk Clock Control Register (Offset = 0x14)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:16]	pwm_period	R/W	PWM Period If setting to 1, it will be gating 15 clock cycles.	0x0000	HR
[15:3]	-	-	Reserved	-	-
[2:1]	mode_sel	R/W	Mode Select 0: Always on 1: PWM mode 2 or 3: Always off	0x0	HR

Bit	Name	Type	Description	Reset Value	Reset Type
0	reg_set	R/W	If setting to 1, it will update the mode_sel and PWM periods (Self-cleared).	0x0	HR

4.1.4.2.6 sys_aclk_0 Clock Control Register (Offset = 0x18)

Table 4-7. sys_aclk_0 Clock Control Register (Offset = 0x18)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:16]	pwm_period	R/W	PWM period If setting to 1, it will be gating 15 clock cycles.	0x0000	HR
[15:3]	-	-	Reserved	-	-
[2:1]	mode_sel	R/W	Mode selection 0: Always on 1: PWM mode 2 or 3: Always off	0x0	HR
0	reg_set	R/W	If setting to 1, it will update the mode_sel and PWM periods (Self-cleared).	0x0	HR

4.1.4.2.7 sys_hclk_0 Clock Control Register (Offset = 0x1C)

Table 4-8. sys_hclk_0 Clock Control Register (Offset = 0x1C)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:16]	pwm_period	R/W	PWM period If setting to 1, it will be gating 15 clock cycles.	0x0000	HR
[15:3]	-	-	Reserved	-	-
[2:1]	mode_sel	R/W	Mode selection 0: Always on 1: PWM mode 2 or 3: Always off	0x0	HR
0	reg_set	R/W	If setting to 1, it will update the mode_sel and PWM periods (Self-cleared).	0x0	HR

4.1.4.2.8 Power Manager Status Register (Offset = 0x20)

Table 4-9. Power Manager Status Register (Offset = 0x20)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:20]	-	-	Reserved	-	-
19	intstdby	R/W	Interrupt for waking up CPU after exiting from the standby mode Cleared by software programming	0x0	HR
18	intidle	R/W	Interrupt for waking up CPU after exiting from the idle mode Cleared by software programming	0x0	HR
17	intfcs	R/W	Interrupt for waking up CPU after completing the frequency change Cleared by software programming	0x0	HR
16	intturbo	R/W	Turbo mode interrupt	0x0	HR
[15:10]	-	-	Reserved	-	-
9	watchdog timer reset	-	Reboot by the watchdog reset 0: Watchdog reset is not occurred since the last CPU execution or since the hardware cleared this bit. 1: Watchdog reset is occurred since the last CPU execution or since the hardware cleared this bit. Set this bit when the watchdog reset occurs Cleared by software programming	0x0	HR
[8:0]	-	-	Reserved	-	-

4.1.4.2.9 sys_pclk_0 Clock Control Register (Offset = 0x24)

Table 4-10. sys_pclk_0 Clock Control Register (Offset = 0x24)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:16]	pwm_period	R/W	PWM period If setting to 1, it will be gating 15 clock cycles.	0x0000	HR
[15:3]	-	-	Reserved	-	-
[2:1]	mode_sel	R/W	Mode selection 0: Always on 1: PWM mode 2 or 3: Always off	0x0	HR

Bit	Name	Type	Description	Reset Value	Reset Type
0	reg_set	R/W	If setting to 1, it will update the mode_sel and PWM periods (Self-cleared).	0x0	HR

4.1.4.2.10 IP Main Clock Select Setting Register (Offset = 0x28)

Table 4-11. IP Main Clock Control Setting Register (Offset = 0x28)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:27]	-	-	Reserved	-	-
[26:24]	PLL1 cntp3	R/W	PLL1 cntp3 divider register	0x2	HR
23	PLL3 OUT	R/W	PLL3 OUT selection 0: PLL3 1: PLL3 div2	0x0	HR
[22:21]	ISP CLK	R/W	ISP CLK 00: ACLK 01: HCLK 1x: PLL1 div3	0x0	HR
20	DI CLK	R/W	DI (De-noise) CLK 0: HCLK 1: ACLK	0x0	HR
19	-	-	Reserved	-	-
18	EXT_CKO_SSCG	R/W	EXT_CKO_SSCG 0: Bypass 1: SSCG (Spread Spectrum Clock Generator)	0x0	HR
17	MPEG4 CLK	R/W	MPEG4/JPEG CLK 0: HCLK 1: PLL1 div3	0x0	HR
[16:15]	ADDA CLK	R/W	ADDA CLK 00: PLL3 (594MHz/540MHz) 01: PLL1 (810MHz/540MHz) cntp3 1x: PLL2 (600MHz)	0x2	HR

Bit	Name	Type	Description	Reset Value	Reset Type
[14:13]	SSP1 CLK	R/W	SSP1 CLK 00: PLL3 (594MHz/540MHz) 01: PLL1 (810MHz/540MHz) cntp3 1x: PLL2 (600MHz)	0x2	HR
[12:11]	SSP0 CLK	R/W	SSP0 CLK 00: PLL3 (594MHz/540MHz) 01: PLL1 (810HMz/540MHz) cntp3 1x: PLL2 (600MHz)	0x2	HR
10	SPI CLK	R/W	SPI CLK 0: PLL2 (600MHz) 1: PLL1 (810MHz/540MHz) cntp3	0x0	HR
9	LCD SCALER CLK	R/W	LCD SCALER CLK 0: lcd_scarclk_cntp (From PLL3) 1: lcd_pixclk_cntp	0x0	HR
8	LCD CLK	R/W	LCD CLK 0: PLL3 (594MHz/540MHz) 1: PLL2 (600MHz)	0x0	HR
7	SDC CLK	R/W	SDC CLK 0: PLL2 (600MHz) 1: PLL1 (810MHz/540MHz)	0x0	HR
6	MIPI phy clock select	R/W	MIPI phy clock selection (27MHz) 0: PLL3 div2 (297MHz/270MHz) 1: PLL1 div2	0x0	HR
[5:4]	RMII MAC PHY CLK select	R/W	RMII MAC PHY CLK selection 00: PLL2 (600MHz) 01: RMII_CK_IN (X_RMII_RST) 1x: PLL1 (810MHz/540MHz)	0x2	HR
3	uart_clk_sel	R/W	UART CLK 0: PLL0 div2 (300MHz) 1: PLL1 div2 (405MHz/270MHz)	0x0	HR
[2:1]	ext_clk_sel	R/W	External clock source selection 00: PLL3 (594MHz/540MHz) 01: OSCH (30MHz) 1x: PLL2 cntp3 (600MHz)	0x0	HR

Bit	Name	Type	Description	Reset Value	Reset Type
0	rtc_clk_sel	R/W	RTC clock source selection 0: OSCL (32768Hz) 1: adda_clk/2 (6.0MHz)	0x1	HR

4.1.4.2.11 PLL1, PLL2, and PLL3 Bandwidth Option Register (Offset = 0x2C)

Table 4-12. PLL1, PLL2, and PLL3 Bandwidth Option Register (Offset = 0x2C)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:6]	-	-	Reserved	-	-
[5:4]	PLL3 bandwidth	R/W	PLL3CC PLL1 band width option 00: 0.25 * BW (Band Width) 01: 0.5 * BW 10: 1 * BW 11: 1.25 * BW	0x2	HR
[3:2]	PLL2 bandwidth	R/W	PLL2CC PLL1 band width option 00: 0.25 * BW (Band Width) 01: 0.5 * BW 10: 1 * BW 11: 1.25 * BW	0x2	HR
[1:0]	PLL1 bandwidth	R/W	PLL1CC PLL1 band width option 00: 0.25 * BW (Band Width) 01: 0.5 * BW 10: 1 * BW 11: 1.25 * BW	0x2	HR

4.1.4.2.12 PLL1 Control Register and cpuxx_mode Setting (Offset = 0x30)

Table 4-13. PLL1 Control Register and cpuxx_mode Setting (Offset = 0x30)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:16]	cpuxx_mode[15:0]	R/W	PLL1 control register and cpuxx_mode setting	It depends on the jump setting.	HR

cpuxx_mode	Function	Description
[1:0]	DDR CLK	00: PLL1 (810MHz /540MHz) CKOUT div 2 01: PLL2 (600MHz) CKOUT div 2 1x: PLL1 (810MHz /540MHz) CKOUT div 3
[3:2]	hclk	00: PLL1 (810MHz /540MHz) CKOUT div 3 01: PLL2 (600MHz) CKOUT div 3 1x: PLL2 (600MHz) CKOUT div 4
[5:4]	aclk	00: PLL1 (810MHz /540MHz) CKOUT div 3 01: PLL2 (600MHz) CKOUT div 3 1x: PLL2 (600MHz) CKOUT div 4
[6]	cpu_fclk	0: cpu_fclk = PLL1 CKOUT 1: cpu_fclk = PLL2 CKOUT
[7]	-	Reserved
[9:8]	aclk_2-	00: PLL1 (810MHz /540MHz) CKOUT div 2 01: PLL2 (600MHz) CKOUT div 2 1x: PLL1 (810MHz/ 540MHz) CKOUT div 3
[11:10]	-	Reserved
[14:12]	DDR DLL frange control	0x0: 250MHz to 400MHz 0x1: 400MHz to 600MHz 0x2: 600MHz to 700MHz 0x3: 700MHz to 800MHz 0x4: 800MHz to 900MHz 0x5: 900MHz to 1000MHz 0x6: 1000MHz to 1200MHz 0x7: 1200MHz to 1600MHz
[15]	-	Reserved

Bit	Name	Type	Description	Reset Value	Reset Type
[15:11]	pll1ms	R/W	This signal indicates the M value of the embedded PLL1 to control the frequency output of PLL1.	It depends on the jump setting.	HR
[10:4]	pll1ns	R/W	This signal indicates the N value of the embedded PLL1 to control the frequency output of PLL1.	It depends on the jump setting.	HR
[3:2]	pll1frang	R/W	Control pins for selecting the CKOUT frequency range 0: CKOUT = 62.5MHz ~ 125MHz 1: CKOUT = 125MHz ~ 250MHz 2: CKOUT = 250MHz ~ 500MHz 3: CKOUT = 500MHz ~ 1000MHz	It depends on the jump setting.	HR
1	pll1stable	R/W	Always 1'b1	0x1	HR
0	pll1dis	R/W	PLL1 control 1: Disable 0: Enable	0x0	HR

4.1.4.2.13 PLL2/PLL3 Control Register (Offset = 0x34)

Table 4-14. PLL2/PLL3 Control Register (Offset = 0x34)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:27]	pll3 ms	R/W	This signal indicates the M value of the embedded PLL3 to control the frequency output of PLL3.	0x03	HR
[26:20]	pll3 ns	R/W	This signal indicates the N value of the embedded PLL3 to control the frequency output of PLL3.	0x36	HR
[19:18]	pll3frang	R/W	This signal is used to select the CKOUT frequency range. 0: CKOUT = 62.5MHz ~ 125MHz 1: CKOUT = 125MHz ~ 250MHz 2: CKOUT = 250MHz ~ 500MHz 3: CKOUT = 500MHz ~ 1000MHz	0x3	HR
[17]	pll3stable	R/W	Always 1'b1	0x1	HR
[16]	pll3en	R/W	PLL3 control 1: Enable 0: Disable	0x1	HR
[15:11]	pll2 ms	R/W	This signal indicates the M value of the embedded PLL2 to control the frequency output of PLL2.	Depends on jump setting	HR

Bit	Name	Type	Description	Reset Value	Reset Type
[10:4]	pll2ns	R/W	This signal indicates the N value of the embedded PLL2 to control the frequency output of PLL2.	Depends on jump setting	HR
[3:2]	pll2frang	R/W	Control pins for selecting the CKOUT frequency range 0: CKOUT = 62.5MHz ~ 125MHz 1: CKOUT = 125MHz ~ 250MHz 2: CKOUT = 250MHz ~ 500MHz 3: CKOUT = 500MHz ~ 1000MHz	Depends on jump setting	HR
1	pll2stable	R/W	Always 1'b1	0x1	HR
0	pll2en	R/W	PLL2 control 1: Enable 0: Disable	0x1	HR

4.1.4.2.14 aclk_2_clk Clock Control Register (Offset = 0x38)

Table 4-15. aclk_2_clk Clock Control Register (Offset = 0x38)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:16]	pwm_period	R/W	PWM period If setting to 1, it will be gating 15 clock cycles.	0x0000	HR
[15:3]	-	-	Reserved	-	-
[2:1]	mode_sel	R/W	Mode selection 0: Always on 1: PWM mode 2 or 3: Always off	0x0	HR
0	reg_set	R/W	If setting to 1, it will update the mode_sel and PWM periods (Self-cleared).	0x0	HR

4.1.4.2.15 Driving Capability and Slew Rate and Hold Time Control Register 0 (Offset = 0x40)

Table 4-16. Driving Capability and Slew Rate and Hold Time Control Register 0 (Offset = 0x40)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:28]	cpu_dcsr	R/W	This signal controls the output driving capability and output slew rate. cpu_dcsr[1:0]: Driving capability control (X_NTRST, X_TCK, X_TDI, X_TDO, and X_TMS) 0: 2mA 1: 4mA 2: 6mA 3: 8mA cpu_dcsr[2]: Schmitt-trigger control (X_NTRST,X_TDI, X_TDO, and X_TMS) 0: Normal operation 1: Schmitt-trigger cpu_dcsr[3]: Slew rate control (X_NTRST, X_TCK, X_TDI, X_TDO, and X_TMS) 0: Fast 1: Slow	0x0	HR
[27:24]	uart_dcsr	R/W	This signal controls the output driving capability and output slew rate. uart_dcsr[1:0]: Driving capability control (X_UART0_SIN, X_UART0_SOUT, X_UART2_SIN, and X_UART2_SOUT) 0: 2mA 1: 4mA 2: 6mA 3: 8mA uart_dcsr[2]: Schmitt-trigger control (X_UART0_SOUT and X_UART2_SOUT) 0: Normal operation 1: Schmitt-trigger uart_dcsr[3]: Slew rate control (X_UART0_SIN, X_UART0_SOUT, X_UART2_SIN, and X_UART2_SOUT) 0: Fast 1: Slow	0x0	HR

Bit	Name	Type	Description	Reset Value	Reset Type
[23:20]	cap_dcsr	R/W	This signal controls the output driving capability and output slew rate. cap_dcsr[1:0]: Driving capability control (X_CAP_RST, X_CAP_CLKOUT, X_CAP0_D0, X_CAP0_D1, X_CAP0_D2, and X_CAP0_D3) 0: 2mA 1: 4mA 2: 6mA 3: 8mA cap_dcsr[1:0]: Driving capability control (X_CAP0_D4, X_CAP0_D5, X_CAP0_D6, and X_CAP0_D7) 0: 4mA 1: 8mA 2: 12mA 3: 16mA cap_dcsr[2]: Schmitt-trigger control (X_CAP_RST, X_CAP_CLKOUT, X_CAP0_D0, X_CAP0_D1, X_CAP0_D4, X_CAP0_D5, X_CAP0_D6, and X_CAP0_D7) 0: Normal operation 1: Schmitt-trigger cap_dcsr[3]: Slew rate control (X_CAP_RST, X_CAP0_D1, X_CAP0_D2, X_CAP0_D3, X_CAP0_D4, X_CAP0_D5, X_CAP0_D6, and X_CAP0_D7) 0: Fast 1: Slow	0x0	HR

Bit	Name	Type	Description	Reset Value	Reset Type
[19:16]	isp_dcsr	R/W	This signal controls the output driving capability and output slew rate. isp_dcsr[1:0]: Driving capability control (X_BAYER_CLK) 0: 2mA 1: 4mA 2: 6mA 3: 8mA isp_dcsr[1:0]: Driving capability control (X_BAYER_D4, X_BAYER_D5, X_BAYER_D6, and X_BAYER_D7) 0: 4mA 1: 8mA 2: 12mA 3: 16mA isp_dcsr[2]: Schmitt-trigger control (X_BAYER_D4, X_BAYER_D5, X_BAYER_D6, and X_BAYER_D7) 0: Normal operation 1: Schmitt-trigger isp_dcsr[3]: Slew rate control (X_BAYER_CLK, X_BAYER_D4, X_BAYER_D5, X_BAYER_D6, and X_BAYER_D7) 0: Fast 1: Slow	0x0	HR

Bit	Name	Type	Description	Reset Value	Reset Type
[15:12]	i2c_dcsr	R/W	This signal controls the output driving capability and output slew rate. (X_I2C_SCL and X_I2C_SDA) i2c_dcsr[1:0]: Driving capability control 0: 2mA 1: 4mA 2: 6mA 3: 8mA i2c_dcsr[2]: Schmitt-trigger control 0: Normal operation 1: Schmitt-trigger i2c_dcsr[3]: Slew rate control 0: Fast 1: Slow	0x0	HR
[11:8]	spi_dcsr	R/W	This signal controls the output driving capability and output slew rate. spi_dcsr[1:0]: Driving capability control (X_SPI_FS, X_SPI_SCLK, X_SPI_TX, and X_SPI_RX) 0: 2mA 1: 4mA 2: 6mA 3: 8mA spi_dcsr[2]: Schmitt-trigger control (X_SPI_FS, X_SPI_SCLK, and X_SPI_TX) 0: Normal operation 1: Schmitt-trigger spi_dcsr[3]: Slew rate control (X_SPI_FS, X_SPI_TX, and X_SPI_RX) 0: Fast 1: Slow	0x0	HR

Bit	Name	Type	Description	Reset Value	Reset Type
[7:4]	sdc_dcsr	R/W	This signal controls the output driving capability and output slew rate. sdc_dcsr[1:0]: Driving capability control (X_SD_CD, X_SD_DAT[3:0], X_SD_CLK, and X_SD_CMD_RSP) 0: 4mA 1: 8mA 2: 12mA 3: 16mA sdc_dcsr[2]: Schmitt-trigger control (X_SD_CD, X_SD_DAT[3:0], X_SD_CLK, and X_SD_CMD_RSP) 0: Normal operation 1: Schmitt-trigger sdc_dcsr[3]: Slew rate control (X_SD_CD, X_SD_DAT[3:0], and X_SD_CMD_RSP) 0: Fast 1: Slow	0x0	HR
[3:0]	-	-	Slew rate control 1: Enable 0: Disable [3]: X_CAP0_D[0] [2]: X_GPIO_DAT[3] [1]: X_SSP1_SCLK [0]: X_SPI_SCLK	-	-

4.1.4.2.16 Driving Capability and Slew Rate and Hold Time Control Register 1 (Offset = 0x44)

Table 4-17. Driving Capability and Slew Rate and Hold Time Control Register 1 (Offset = 0x44)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:28]	Slew rate control	R/W	Slew rate control 1: Enable 0: Disable [31]: X_SD_CLK [30]: X_TV_PCLK [29]: X_RMII_CKO [28]: X_CAP_CLKOUT	0x0	HR
[27:25]	-	-	Reserved		HR
[24:16]	Schmitt trigger control	R/W	Schmitt-trigger control 1: Enable 0: Disable [24]: X_RMII_RST [23]: X_SSP1_FS [22]: X_UART2_SIN [21]: X_SPI_RXD [20]: X_CAP0_D[2] [19]: X_CAP0_D[3] [18]: X_BAYER_CLK [17]: X_UART0_SIN [16]: X_TCK	0x000	HR

Bit	Name	Type	Description	Reset Value	Reset Type
[15:12]	gpio_dcsr	R/W	This signal controls the output driving capability and output slew rate. gpio_dcsr[1:0]: Driving capability control (X_GPIO_DAT[3:0]) 0: 4mA 1: 8mA 2: 12mA 3: 16mA gpio_dcsr[2]: Schmitt-trigger control (X_GPIO_DAT[3:0]) 0: Normal operation 1: Schmitt-trigger gpio_dcsr[3]: Slew rate control (X_GPIO_DAT[2:0]) 0: Fast 1: Slow	0x0	HR
[11:8]	pwm_dcsr	R/W	This signal controls the output driving capability and output slew rate. (X_PWM0 and X_PWM1) pwm_dcsr[1:0]: Driving capability control 0: 2mA 1: 4mA 2: 6mA 3: 8mA pwm_dcsr[2]: Schmitt-trigger control 0: Normal operation 1: Schmitt-trigger pwm_dcsr[3]: Slew rate control 0: Fast 1: Slow	0x0	HR

Bit	Name	Type	Description	Reset Value	Reset Type
[7:4]	spi1_dcsr	R/W	This signal controls the output driving capability and output slew rate. spi1_dcsr[1:0]: Driving capability control (X_SSP1_FS, X_SSP1_SCLK, X_SSP1_TX, and X_SSP1_RX) 0: 4mA 1: 8mA 2: 12mA 3: 16mA spi1_dcsr[2]: Schmitt-trigger control (X_SSP1_SCLK, X_SSP1_TX, and X_SSP1_RX) 0: Normal operation 1: Schmitt-trigger spi1_dcsr[3]: Slew rate control (X_SSP1_FS, X_SSP1_TX, and X_SSP1_RX) 0: Fast 1: Slow	0x0	HR

Bit	Name	Type	Description	Reset Value	Reset Type
[3:0]	rmii_dcsr	R/W	This signal controls the output driving capability and output slew rate. rmii_dcsr[1:0]: Driving capability control (X_RMII_RX_CRS_DV, X_RMII_RXD[1:0], X_RMII_RX_ER, X_RMII_TXD[1:0], X_RMII_TX_EN, X_RMII_CKO, X_RMII_MDC, X_RMII_MDIO, X_RMII_PHYLINK, X_RMII_RST, and X_GPIO_DAT[8:4]) 0: 4mA 1: 8mA 2: 12mA 3: 16mA rmii_dcsr[2]: Schmitt-trigger control (X_RMII_RX_CRS_DV, X_RMII_RXD[1:0], X_RMII_RX_ER, X_RMII_TXD[1:0], X_RMII_TX_EN, X_RMII_CKO, X_RMII_MDC, X_RMII_MDIO, X_RMII_PHYLINK, and X_GPIO_DAT[8:4]) 0: Normal operation 1: Schmitt-trigger rmii_dcsr[3]: Slew rate control (X_RMII_RX_CRS_DV, X_RMII_RXD[1:0], X_RMII_RX_ER, X_RMII_TXD[1:0], X_RMII_TX_EN, X_RMII_MDC, X_RMII_MDIO, X_RMII_PHYLINK, X_RMII_RST, and X_GPIO_DAT[7:4]) 0: Fast 1: Slow	0x0	HR

4.1.4.2.17 Bus Bridge Control Register (Offset = 0x48)

Table 4-18. Bus Bridge Control Register (Offset = 0x48)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:12]	-	-	Reserved	-	-
11	CSysReq_u_X2P030	R/W	CSysReq_u_X2P030 0: Request to enter the power-down mode 1: Normal	0x1	HR
[10:4]	-	-	Reserved	-	-
3	CSysReq_u_X2H030_1	-	CSysReq_u_X2H030_1 0: Request to enter the power-down mode 1: Normal	0x1	HR
2	-	-	Reserved	-	-
1	CSysReq_u_X2H030	-	CSysReq_u_X2H030 0: Request to enter the power-down mode 1: Normal	0x1	HR
0	-	-	Reserved	-	-

4.1.4.2.18 Multi-function Port Setting Register 0 (MFPSR0, Offset = 0x50)

Table 4-19. Multi-function Port Setting Register 0 (MFPSR0, Offset = 0x50)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:30]	BAYER_D6	R/W	It uses the “Bayer Pin-mux” table for settings. 0: GPIO_0[21] 1: CAP0_D[10] 2: BAYER_D6 3: SD1_D1	0x0	HR
[29:28]	BAYER_D7	R/W	It uses the “Bayer Pin-mux” table for settings. 0: GPIO_0[20] 1: CAP0_D[11] 2: BAYER_D6 3: SD1_CD	0x0	HR

Bit	Name	Type	Description	Reset Value	Reset Type
[27:26]	DP1 (PLDATA[5])	R/W	It uses the “Serial Combo PHY Pin-mux” table for settings. If setting to 2, the Bayer source selection depends on Offset 0x7C[21:20]. 0: GPIO_0[8] 1: CAP0_D[12] 2: BAYER_D8/CAP1_D[4]	0x0	HR
[25:24]	DN1 (PLDATA[4])	R/W	It uses the “Serial Combo PHY Pin-mux” table for settings. If setting to 2, the Bayer source selection depends on Offset 0x7C[21:20]. 0: GPIO_0[7] 1: CAP0_D[13] 2: BAYER_D9/CAP1_D[5]	0x0	HR
[23:22]	CKP (PLDATA[3])	R/W	It uses the “Serial Combo PHY Pin-mux” table for settings. If setting to 2, the Bayer source selection depends on Offset 0x7C[21:20]. 0: None 1: CAP0_D[14] 2: BAYER_D10/CAP1_D[6]	0x0	HR
[21:20]	CKN (PLDATA[2])	R/W	It uses the “Serial Combo PHY pin-mux” table for settings. If setting to 2, the Bayer source selection depends on Offset 0x7C[21:20]. 0: None 1: CAP0_D[15] 2: BAYER_D11/CAP1_D[7]	0x0	HR
[19:18]	DP0 (PLDATA[1])	R/W	It uses the “Serial Combo PHY Pin-mux” table for settings. If setting to 2, the Bayer source selection depends on Offset 0x7C[21:20]. 0: None 1: CAP0_VS 2: BAYER_VS	0x0	HR

Bit	Name	Type	Description	Reset Value	Reset Type
[17:16]	DN0 (PLDATA[0])	R/W	It uses the “Serial Combo PHY Pin-mux” table for settings. If setting to 2, the Bayer source selection depends on Offset 0x7C[21:20]. 0: CAP0_CLK 1: CAP0_HS 2: BAYER_HS	0x0	HR
[15:14]	BAYER_CLK	R/W	It uses the “Bayer Pin-mux” table for settings. 0: GPIO_0[9] 1: CAP0_CLK 2: BAYER_CLK/CAP1_CLK	0x0	HR
[13:12]	CAP_CLKOUT	R/W	It uses the “Bayer Pin-mux” table for settings. 0: GPIO_0[6] 1: CAP_CLKOUT	0x0	HR
[11:10]	-	-	Reserved	-	-
[9:8]	TDI	R/W	It uses the “ICE Pin-mux” table for settings. 0: TDI 1: GPIO_0[4] 2: PWM7 3: DMIC_CLK	0x0	HR
[7:6]	TDO	R/W	It uses the “ICE Pin-mux” table for settings. 0: TDO 1: GPIO_0[3] 2: PWM6 3: DMIC_DATA	0x0	HR
[5:4]	TMS	R/W	It uses the “ICE Pin-mux” table for settings. 0: TMS 1: GPIO_0[2] 2: PWM5 3: UART2_SOUT	0x0	HR
[3:2]	TCK	R/W	It uses the “ICE Pin-mux” table for settings. 0: TCK 1: GPIO_0[1] 2: PWM4 3: UART2_SIN	0x0	HR

Bit	Name	Type	Description	Reset Value	Reset Type
[1:0]	NTRST	R/W	It uses the “ICE Pin-mux” table settings. 0: NTRST 1: GPIO_0[0]	0x0	HR

4.1.4.2.19 Multi-function Port Setting Register 1 (MFPSR1, Offset = 0x54)

Table 4-20. Multi-function Port Setting Register 1 (MFPSR1, Offset = 0x54)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:26]	-	R/W	Reserved	0x0	HR
[25:24]	I2C_SDA	R/W	It uses the “I2C Pin-mux” table for settings. 0: GPIO_0[19] 1: I2C_SDA 2: PWM3	0x0	HR
[23:22]	I2C_SCL	R/W	It uses the “I2C Pin-mux” table for settings. 0: GPIO_0[18] 1: I2C_SCL 2: PWM2	0x0	HR
[21:20]	CAP0_D[0]	R/W	It uses the “CAP Pin-mux” table for settings. 0: GPIO_0[17] 1: CAP_LO8_D0 2: None 3: SSP1_SCLK	0x0	HR
[19:18]	CAP0_D[1]	R/W	It depends on the “CAP Pin-mux” table for settings. 0: GPIO_0[16] 1: CAP_LO8_D1 2: None 3: SSP1_TXD	0x0	HR
[17:16]	CAP0_D[2]	R/W	It uses the “CAP Pin-mux” table for settings. 0: GPIO_0[15] 1: CAP0_D[2] 2: I2C_SDA 3: SSP1_RXD	0x0	HR

Bit	Name	Type	Description	Reset Value	Reset Type
[15:14]	CAP0_D[3]	R/W	It uses the “CAP Pin-mux” table for settings. 0: GPIO_0[14] 1: CAP0_D[3] 2: I2C_SCL 3: SSP1_FS	0x0	HR
[13:12]	CAP0_D[4]	R/W	It uses the “CAP Pin-mux” table for settings. 0: GPIO_0[13] 1: CAP0_D[4] 2: BAYER_D0	0x0	HR
[11:10]	CAP0_D[5]	R/W	It uses the “CAP Pin-mux” table for setting. 0: GPIO_0[12] 1: CAP0_D[5] 2: BAYER_D1 3: SD1_D2	0x0	HR
[9:8]	CAP0_D[6]	R/W	It uses the “CAP Pin-mux” table for settings. 0: GPIO_0[11] 1: CAP0_D[6] 2: BAYER_D2 3: SD1_D3	0x0	HR
[7:6]	CAP0_D[7]	R/W	It uses the “CAP Pin-mux” table for settings. 0: GPIO_0[10] 1: CAP0_D[7] 2: BAYER_D3 3: SD1_CMD_RSP	0x0	HR
[5:4]	-	R/W	Reserved	0x0	HR
[3:2]	BAYER_D4	R/W	It uses the “BAYER Pin-mux” table for settings. 0: GPIO_0[23] 1: CAP0_D[8] 2: BAYER_D4 3: SD1_CLK	0x0	HR
[1:0]	BAYER_D5	R/W	It uses the “BAYER Pin-mux” table for settings. 0: GPIO_0[22] 1: CAP0_D[9] 2: BAYER_D5 3: SD1_D0	0x0	HR

4.1.4.2.20 Multi-function Port Setting Register 2 (MFPSR2, Offset = 0x58)

Table 4-21. Multi-function Port Setting Register 2 (MFPSR2, Offset = 0x58)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:30]	UART0_SOUT	R/W	It uses the “UART0 Pin-mux” table for settings. 0: GPIO_1[6] 1: UART0_SOUT	0x0	HR
[29:28]	UART0_SIN	R/W	It uses the “UART0 Pin-mux” table for setting. 0: GPIO_1[5] 1: UART0_SIN	0x0	HR
[27:26]	UART2_SOUT	R/W	It uses the “UART2 Pin-mux” table for settings. 0: GPIO_1[4] 1: UART2_SOUT	0x0	HR
[25:24]	UART2_SIN	R/W	It uses the “UART2 Pin-mux” table for settings. 0: GPIO_1[3] 1: UART2_SIN	0x0	HR
[23:22]	SD_CMD_RSP	R/W	It uses the “MEMC Pin-mux” table for settings. 0: GPIO_1[2] 1: SD_CMD_RSP	0x0	HR
[21:20]	SD_CLK	R/W	It uses the “MEMC Pin-mux” table for settings. 0: GPIO_1[1] 1: SD_CLK	0x0	HR
[19:18]	SD_CD	R/W	It uses the “MEMC Pin-mux” table for settings. 0: GPIO_1[0] 1: SD_CD	0x0	HR
[17:16]	GPIO_DAT[3]	R/W	It uses the “GPIO Pin-mux” table for settings. 0: GPIO_0[31] 1: SSP1_SCLK 2: SD1_DAT[2] 3: OTG_30M_dbg	0x0	HR
[15:14]	GPIO_DAT[2]	R/W	It uses the “GPIO Pin-mux” table for settings. 0: GPIO_0[30] 1: SSP1_RXD 2: SD1_DAT[3] 3: OTG_48M_dbg	0x0	HR

Bit	Name	Type	Description	Reset Value	Reset Type
[13:12]	GPIO_DAT[1]	R/W	It uses the “GPIO Pin-mux” table for settings. 0: GPIO_0[29] 1: SSP1_TXD 2: SD1_CMD_RSP	0x0	HR
[11:10]	GPIO_DAT[0]	R/W	It uses the “GPIO Pin-mux” table for settings. 0: GPIO_0[28] 1: SSP1_FS	0x0	HR
[9:8]	SD_DAT[3]	R/W	It uses the “MEMC Pin-mux” table for settings. 0: GPIO_0[27] 1: SD_DAT[3]	0x0	HR
[7:6]	SD_DAT[2]	R/W	It uses the “MEMC Pin-mux” table for settings. 0: GPIO_0[26] 1: SD_DAT[2]	0x0	HR
[5:4]	SD_DAT[1]	R/W	It uses the “MEMC Pin-mux” table for settings. 0: GPIO_0[25] 1: SD_DAT[1]	0x0	HR
[3:2]	SD_DAT[0]	R/W	It uses the “MEMC Pin-mux” table for settings. 0: GPIO_0[24] 1: SD_DAT[0]	0x0	HR
[1:0]	-	R/W	Reserved	0x0	HR

4.1.4.2.21 Multi-function Port Setting Register 3 (MFPSR3, Offset = 0x5C)

Table 4-22. Multi-function Port Setting Register 3 (MFPSR3, Offset = 0x5C)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:30]	GPIO_DAT[7]	R/W	It uses the “GPIO Pin-mux” table for settings. 0: GPIO_1[22] 1: SD1_CLK 2: LC_DATA[15] 3: TV_DATA[15]	0x0	HR

Bit	Name	Type	Description	Reset Value	Reset Type
[29:28]	GPIO_DAT[6]	R/W	It uses the “GPIO pin-mux” table for settings. 0: GPIO_1[21] 1: SD1_DAT[0] 2: LC_DATA[14] 4: TV_DATA[14]	0x0	HR
[27:26]	GPIO_DAT[5]	R/W	It uses the “GPIO Pin-mux” table for settings. 0: GPIO_1[20] 1: SD1_DAT[1] 2: LC_DATA[13] 3: TV_DATA[13]	0x0	HR
[25:24]	GPIO_DAT[4]]	R/W	It uses the “GPIO Pin-mux” table for settings. 0: GPIO_1[19] 1: SD1_CD 2: LC_DATA[12] 4: TV_DATA[12]	0x0	HR
[23:22]	RMII_RST	R/W	It uses the “RMII MAC Pin-mux” table for settings. 0: GPIO_1[18] 1: RMII_CK_IN 2: LC_DATA[11] 3: TV_DATA[11]	0x0	HR
[21:20]	RMII_TX_EN	R/W	It uses the “RMII MAC Pin-mux” table for settings. 0: GPIO_1[17] 1: RMII_TX_EN 2: LC_DATA[10] 3: TV_DATA[10]	0x0	HR
[19:18]	RMII_TXD[1]	R/W	It uses the “RMII MAC Pin-mux” table for settings. 0: GPIO_1[16] 1: RMII_TXD[1] 3: LC_DATA[9] 4: TV_DATA[9]	0x0	HR

Bit	Name	Type	Description	Reset Value	Reset Type
[17:16]	RMII_TXD[0]	R/W	It uses the “RMII MAC Pin-mux” table for settings. 0: GPIO_1[15] 1: RMII_TXD[0] 2: LC_DATA[8] 3: TV_DATA[8]	0x0	HR
[15:14]	RMII_CKO	R/W	It uses the “RMII MAC Pin-mux” table for settings. 0: GPIO_1[14] 1: RMII_CKO 2: LC_DATA[7] 3: TV_DATA[7]	0x0	HR
[13:12]	RMII_PHYLINK	R/W	It uses the “RMII MAC Pin-mux” table for settings. 0: GPIO_1[13] 1: RMII_PHYLINK 2: LC_DATA[6] 3: TV_DATA[6]	0x0	HR
[11:10]	RMII_RX_ER	R/W	It uses the “RMII MAC Pin-mux” table for settings. 0: GPIO_1[12] 1: RMII_RX_ER 2: LC_DATA[2] 3: TV_DATA[2]	0x0	HR
[9:8]	RMII_RXD[1]	R/W	It uses the “RMII MAC Pin-mux” table for settings. 0: GPIO_1[11] 1: RMII_RXD[1] 2: LC_DATA[5] 3: TV_DATA[5]	0x0	HR
[7:6]	RMII_RXD[0]	R/W	It uses the “RMII MAC Pin-mux” table for settings. 0: GPIO_1[10] 1: RMII_RXD[0] 2: LC_DATA[4] 3: TV_DATA[4]	0x0	HR

Bit	Name	Type	Description	Reset Value	Reset Type
[5:4]	RMII_RX_CRS_DV	R/W	It uses the “RMII MAC Pin-mux” table for settings. 0: GPIO_1[9] 1: RMII_RX_CRS_DV 2: LC_DATA[3] 3: TV_DATA[3]	0x0	HR
[3:2]	RMII_MDC	R/W	It uses the “RMII MAC Pin-mux” table for settings. 0: GPIO_1[8] 1: RMII_MDC 2: LC_DATA[1] 3: TV_DATA[1]	0x0	HR
[1:0]	RMII_MDIO	R/W	It uses the “RMII MAC Pin-mux” table for settings. 0: GPIO_1[7] 1: RMII_MDIO 2: LC_DATA[0] 3: TV_DATA[0]	0x0	HR

4.1.4.2.22 Multi-function Port Setting Register 4 (MFPSR4, Offset = 0x60)

Table 4-23. Multi-function Port Setting Register 4 (MFPSR4, Offset = 0x60)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:0]	-	R/W	Reserved	0x0	HR

4.1.4.2.23 Multi-function Port Setting Register 5 (MFPSR5, Offset = 0x64)

Table 4-24. Multi-function Port Setting Register 5 (MFPSR5, Offset = 0x64)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:30]	DDR	R/W	It uses the “DDR Pin-mux” table for settings. 0: map 0 (package ID = 5'h00, 5'h08, 5'h0c and 5'h1c) 1: map 1 (package ID = 5'h04 or 5'h10) 2: map 2 (package ID = 5'h14)	-	-
[29:20]	-	R/W	Reserved	0x0	HR
[19:18]	OTG11_DP	R/W	It uses the “USB 1.1 Pin-mux” table for settings. 0: GPIO_1[31] 1: OTG11_DP	0x0	HR
[17:16]	OTG11_DN	R/W	It uses the “USB 1.1 Pin-mux” table for settings. 0: GPIO_1[30] 1: OTG11_DN	0x0	HR
[15:14]	PMW1	R/W	It uses the “PMW0/1 Pin-mux” table for settings. 0: GPIO_1[29] 1: PWM1 2: DMIC_CLK	0x0	HR
[13:12]	PWM0	R/W	It uses the “PWM0/1 Pin-mux” table for settings. 0: GPIO_1[28] 1: PWM0 2: DMIC_DATA	0x0	HR
[11:10]	-	R/W	Reserved	0x0	HR
[9:8]	TV_PCLK	R/W	It uses the “TV Pin-mux” table for settings. 0: Don’t use 1: None 2: LC_PCLK 3: TV_PCLK Note: If X_SSP1_SCLK is in the SSP1 slave mode, then bit 8 must be logic 1.	0x0	HR

Bit	Name	Type	Description	Reset Value	Reset Type
[7:6]	SSP1_RXD	R/W	It uses the “SSP1 Pin-mux” table for settings. 0: GPIO_1[26] 1: SSP1_RXD 2: LC_VS	0x0	HR
[5:4]	SSP1_TXD	R/W	It uses the “SSP1 Pin-mux” table for settings. 0: GPIO_1[25] 1: SSP1_TXD 2: UART1_SOUT	0x0	HR
[3:2]	SSP1_FS	R/W	It uses the “SSP1 Pin-mux” table for settings. 0: GPIO_1[24] 1: SSP1_FS 2: UART1_SIN	0x0	HR
[1:0]	SSP1_SCLK	R/W	It uses the “SSP1 Pin-mux” table for settings. 0: GPIO_1[27] 1: SSP1_SCLK (If SSP1 is slave mode, then bit [8] must be logic 1.) 2: LC_HS	0x0	HR

Table 4-25 through Table 4-38 describe the detailed pin-mux functions and default states.

Table 4-25. PWM0/1 Pin-mux

Ball Name	Power-On-Reset State	Multi-function Pin and Function Default State				Control Register and Default	Unused PCB Setting and Comment	Comment
		regxx bit[y:x] = 2'b00	regxx bit[y:x] = 2'b01	regxx bit[y:x] = 2'b10	regxx bit[y:x] = 2'b11			
X_PWM0	-	X_GPIO_1[28], I, pull-up	X_PWM0, O	X_DMIC_DATA, I, PU	-	reg64[13:12], default = 2'b00	Floating	-
X_PWM1	-	X_GPIO_1[29], I, pull-up	X_PWM1, O	X_DMIC_CLK, O	-	reg64[15:14], default = 2'b00	Floating	-

Table 4-26. Video-in Interface Pin-mux 0

Ball Name	Power-On Reset State	Multi-function Pin and Function Default State				Control Register and Default	Unused PCB Setting and Comment	Comment
		regxx bit[y:x] = 2'b00	regxx bit[y:x] = 2'b01	regxx bit[y:x] = 2'b10	regxx bit[y:x] = 2'b11			
X_CAP_RST	I, internal pull-down	X_GPIO_0[5], I, pull-down	-	-	-	-	Jumper setting 0, internal pull-down	Jumper[0], internal pull-down
X_CAP_CLKOUT	I, internal pull-up	X_GPIO_0[6], I, pull-up	X_CAP_CLKOUT, O	-	-	reg50[13:12], default = 2'b00	Floating	-
X_BAYER_CLK	I, internal pull-up	X_GPIO_0[9], I, pull-up	X_CAP0_CLK, I	X_BAYER_CLK /X_CAP1_CLK, I	-	reg50[15:14], default = 2'b00	Floating	When X_CAP1_CLK function reg7c[22] = 1'b1

Table 4-27. Video-in Interface Pin-mux 1

Ball Name	Power-On Reset State	Multi-function Pin and Function Default State				Control Register and Default	Unused PCB Setting and Comment	Comment
		regxx bit[y:x] = 2'b00	regxx bit[y:x] = 2'b01	regxx bit[y:x] = 2'b10	regxx bit[y:x] = 2'b11			
X_MPRX_DN0	I	X_CAP0_CLK, I	X_CAP0_HS, I	X_BAYER_HS, I	-	reg50[17:16], default = 2'b00	Floating, based on 9ba00060[3] = 1'b1	Based on 9ba00060[3] = 1'b1, when X_CAP0_CLK function, reg50[15:14] != 2'b10
X_MPRX_DP0	I	-	X_CAP0_VS, I	X_BAYER_VS, I	-	reg50[19:18], default = 2'b00	Floating, based on 9ba00060[3] = 1'b1	Based on 9ba00060[3] = 1'b1
X_MPRX_CKN	I	-	X_CAP0_D[15], I	X_BAYER_D11 /X_CAP1_D[7], I	-	reg50[21:20], default = 2'b00	Floating, based on 9ba00060[3] = 1'b1	Based on 9ba00060[3] = 1'b1
X_MPRX_CKP	I	-	X_CAP0_D[14], I	X_BAYER_D10 /X_CAP1_D[6], I	-	reg50[23:22], default = 2'b00	Floating, based on 9ba00060[3] = 1'b1	Based on 9ba00060[3] = 1'b1
X_MPRX_DN1	I	X_GPIO_0[7], I	X_CAP0_D[13], I	X_BAYER_D9 /X_CAP1_D[5], I	-	reg50[25:24], default = 2'b00	Floating, based on 9ba00060[3] = 1'b1	Based on 9ba00060[3] = 1'b1
X_MPRX_DP1	I	X_GPIO_0[8], I	X_CAP0_D[12], I	X_BAYER_D8 /X_CAP1_D[4], I	-	reg50[27:26], default = 2'b00	Floating, based on 9ba00060[3] = 1'b1	Based on 9ba00060[3] = 1'b1

Table 4-28. Video-in Interface Pin-mux 2

Ball Name	Power-On Reset State	Multi-function Pin and Function Default State				Control Register and Default	Unused PCB Setting and Comment	Comment
		regxx bit[y:x] = 2'b00	regxx bit[y:x] = 2'b01	regxx bit[y:x] = 2'b10	regxx bit[y:x] = 2'b11			
X_BAYER_D7	I, internal pull-up	X_GPIO_0[20], I, pull-up	X_CAP0_D[11], I, pull-up	X_BAYER_D7, X_CAP1_D[3], I, pull-up	X_SD1_CD, I, pull-up	reg50[29:28], default = 2'b00	Floating	-
X_BAYER_D6	I, internal pull-up	X_GPIO_0[21], I, pull-up	X_CAP0_D[10], I, pull-up	X_BAYER_D6, X_CAP1_D[2], I, pull-up	X_SD1_D1, I, pull-up	reg50[31:30], default = 2'b00	Floating	-
X_BAYER_D5	I, internal pull-up	X_GPIO_0[22], I, pull-up	X_CAP0_D[9], I, pull-up	X_BAYER_D5, X_CAP1_D[1], I, pull-up	X_SD1_D0, I, pull-up	reg54[1:0], default = 2'b00	Floating	-
X_BAYER_D4	I, internal pull-up	X_GPIO_0[23], I, pull-up	X_CAP0_D[8], I, pull-up	X_BAYER_D4, X_CAP1_D[0], I, pull-up	X_SD1_CLK, O	reg54[3:2], default = 2'b00	Floating	-

Table 4-29. Video-in Interface Pin-mux 3

Ball Name	Power-On Reset State	Multi-function Pin and Function Default State				Control Register and Default	Unused PCB Setting and Comment	Comment
		regxx bit[y:x] = 2'b00	regxx bit[y:x] = 2'b01	regxx bit[y:x] = 2'b10	regxx bit[y:x] = 2'b11			
X_CAP0_D[7]	I, internal pull-up	X_GPIO_0[10], I, pull-up	X_CAP0_D[7], I, pull-up	X_BAYER_D3, I, pull-up	X_SD1_CMD_RSP, O, 1	reg54[7:6], default = 2'b00	Floating	-
X_CAP0_D[6]	I, internal pull-up	X_GPIO_0[11], I, pull-up	X_CAP0_D[6], I, pull-up	X_BAYER_D2, I, pull-up	X_SD1_D3, I, pull-up	reg54[9:8], default = 2'b00	Floating	-
X_CAP0_D[5]	I, internal pull-up	X_GPIO_0[12], I, pull-up	X_CAP0_D[5], I, pull-up	X_BAYER_D1, I, pull-up	X_SD1_D2, I, pull-up	reg54[11:10], default = 2'b00	Floating	-
X_CAP0_D[4]	I, internal pull-up	X_GPIO_0[13], I, pull-up	X_CAP0_D[4], I, pull-up	X_BAYER_D0, I, pull-up	-	reg54[13:12], default = 2'b00	Floating	-
X_CAP0_D[3]	I, internal pull-up	X_GPIO_0[14], I, pull-up	X_CAP0_D[3], I, pull-up	X_I2C_SCL, I, pull-up	X_SSP1_FS, O, pull-up	reg54[15:14], default = 2'b00	Floating	-
X_CAP0_D[2]	I, internal pull-up	X_GPIO_0[15], I, pull-up	X_CAP0_D[2], I, pull-up	X_I2C_SDA, I, pull-up	X_SSP1_RXD, I, pull-up	reg54[17:16], default = 2'b00	Floating	-
X_CAP0_D[1]	I, internal pull-up	X_GPIO_0[16], I, pull-up	X_CAP0_D[1], I, pull-up	-	X_SSP1_TXD, I, pull-up	reg54[19:18], default = 2'b00	Floating	-
X_CAP0_D[0]	I, internal pull-up	X_GPIO_0[17], I, pull-up	X_CAP0_D[0], I, pull-up	-	X_SSP1_SCLK, O	reg54[21:20], default = 2'b00	Floating	-

Application	Reg. offset	Bit	Value	Comment
MIPI + BT656 in	32'h9ba00060	[3]	1'b0	Mux MIPI PHY to the MIPI/subLVDS PHY mode
	scu reg54	[21:6]	16'h5555	Mux to CAP0 656
	scu reg50	[13:12]	2'b01	
BT656 x2 in	32'h9ba00060	[3]	1'b1	Mux MIPI PHY to the parallel mode
	scu reg50	[17:16]	2'h0	Mux to CAP0_CLK
	scu reg54	[21:6]	16'h5555	Mux to CAP0_D[7:0]
	scu reg50	[15:14]	2'b10	Mux to CAP1_CLK
	scu reg54	[3:0]	4'ha	Mux to CAP1_D[7:0]
	scu reg50	[31:20]	12'haaa	
BT1120/ CCIR601 in	32'h9ba00060	[3]	1'b1	Mux MIPI PHY to the parallel mode
	scu reg50	[15:14]	2'b01	Mux to CAP0_CLK
	scu reg50	[31:16]	16'h5555	Mux to CAP0_HS, CAP0_VS, and CAP0_D[15:10]
	scu reg54	[3:0]	4'h5	Mux to CAP0_D[9:8]
	scu reg54	[21:6]	16'h5555	Mux to CAP0_D[7:0]
BAYER in	32'h9ba00060	[3]	1'b1	Mux MIPI PHY to the parallel mode
	scu reg50	[15:14]	2'b10	Mux to BAYER_CLK
	scu reg50	[31:16]	16'haaaa	Mux to BAYER_HS, BAYER_VS, and BAYER_D[11:6]
	scu reg54	[3:0]	4'ha	Mux to BAYER_D[5:4]
	scu reg54	[13:6]	8'haa	Mux to BAYER_D[3:0]

Table 4-30. I^2C Pin-mux

Ball Name	Power-On Reset State	Multi-function Pin and Function Default State				Control Register and Default	Unused PCB Setting and Comment	Comment
		regxx bit[y:x] = 2'b00	regxx bit[y:x] = 2'b01	regxx bit[y:x] = 2'b10	regxx bit[y:x] = 2'b11			
X_I2C_SCL	I, internal pull-up	X_GPIO_0[18], I, pull-up	X_I2C_SCL, I, pull-up	X_PWM2, O	-	reg54[23:22], default = 2'b00	Floating	-
X_I2C_SDA	I, internal pull-up	X_GPIO_0[19], I, pull-up	X_I2C_SDA, I, pull-up	X_PWM3, O	-	reg54[25:24], default = 2'b00	Floating	-

Table 4-31. MEMC Pin-mux

Ball Name	Power-On Reset State	Multi-function Pin and Function Default State				Control Register and Default	Unused PCB Setting and Comment	Comment
		regxx bit[y:x] = 2'b00	regxx bit[y:x] = 2'b01	regxx bit[y:x] = 2'b10	regxx bit[y:x] = 2'b11			
X_SPI_TXD	I	X_SPI_TXD, I	-	-	-	-	pull-up	-
X_SPI_SCLK	I, jumper setting	X_SPI_SCLK, O, 0	-	-	-	-	Jumper setting 6	Jumper[6]
X_SPI_FS	O, internal pull-up	S_SPI_FS, O, pull-up	-	-	-	-	Floating	-
X_SPI_RXD	I, internal pull-up	X_SPI_RXD, I, pull-up	-	-	-	-	Floating	-
X_SD_CD	I, internal pull-up	X_GPIO_1[0], I, pull-up	X_SD_CD, I, pull-up	-	-	reg58[19:18], default = 2'b00	Floating	-
X_SD_DAT[1]	I, internal pull-up	X_GPIO_0[25], I, pull-up	X_SD_DAT[1], I, pull-up	-	-	reg58[5:4], default = 2'b00	Floating	-
X_SD_DAT[0]	I, internal pull-up	X_GPIO_0[24], I, pull-up	X_SD_DAT[0], I, pull-up	-	-	reg58[3:2], default = 2'b00	Floating	-
X_SD_CLK	I, internal pull-down	X_GPIO_1[1], I, pull-down	X_SD_CLK, O	-	-	reg58[21:20], default = 2'b00	Jumper setting 9, internal pull-down	Jumper[19], internal pull-down
X_SD_CMD_RSP	I, internal pull-up	X_GPIO_1[2], I, pull-up	X_SD_CMD_RSP, O, 1, pull-up	-	-	reg58[23:22], default = 2'b00	Floating	-
X_SD_DAT[3]	I, internal pull-up	X_GPIO_0[27], I, pull-up	X_SD_DAT[3], I, pull-up	-	-	reg58[9:8], default = 2'b00	Floating	-
X_SD_DAT[2]	I, internal pull-up	X_GPIO_0[26], I, pull-up	X_SD_DAT[2], I, pull-up	-	-	reg58[7:6], default = 2'b00	Floating	-

Table 4-32. UART2 Pin-mux

Ball Name	Power-On Reset State	Multi-function Pin and Function Default State				Control Register and Default	Unused PCB Setting and Comment	Comment
		regxx bit[y:x] = 2'b00	regxx bit[y:x] = 2'b01	regxx bit[y:x] = 2'b10	regxx bit[y:x] = 2'b11			
X_UART2_SIN	I, internal pull-up	X_GPIO_1[3], I, pull-up	X_UART2_SIN, I, pull-up	-	-	reg58[25:24], default = 2'b00	Floating	-
X_UART2_SOUT	I, internal pull-up	X_GPIO_1[4], I, pull-up	X_UART2_SOUT, O, 1	-	-	reg58[27:26], default = 2'b00	Jumper setting 8, internal pull-up	Jumper[8], internal pull-up

Table 4-33. OTG 1(OTG 1.1) Pin-mux

Ball Name	Power-On Reset State	Multi-function Pin and Function Default State				Control Register and Default	Unused PCB Setting and Comment	Comment
		regxx bit[y:x] = 2'b00	regxx bit[y:x] = 2'b01	regxx bit[y:x] = 2'b10	regxx bit[y:x] = 2'b11			
X_OTG11_DN	I, internal pull-up	X_GPIO_1[30], I, pull-up	X_OTG11_DN	-	-	reg64[17:16], default = 2'b00	Floating	-
X_OTG11_DP	I, internal pull-up	X_GPIO_1[31], I, pull-up	X_OTG11_DP	-	-	reg64[17:16], default = 2'b00	Floating	-

Table 4-34. SSP1 Pin-mux

Ball Name	Power-On Reset State	Multi-function Pin and Function Default State				Control Register and Default	Unused PCB Setting and Comment	Comment
		regxx bit[y:x] = 2'b00	regxx bit[y:x] = 2'b01	regxx bit[y:x] = 2'b10	regxx bit[y:x] = 2'b11			
X_SSP1_FS	I, internal pull-up	X_GPIO_1[24], I, pull-up	X_SSP1_FS, O, pull-up	X_UART1_SIN, I, pull-up	-	reg64[3:2], default = 2'b00	Floating	-
X_SSP1_TXD	I, internal pull-down	X_GPIO_1[25], I, pull-down	X_SSP1_TXD, I, pull-up	X_UART1_SOUT, O, 1, pull-up	-	reg64[5:4], default = 2'b00	Jumper setting	Jumper[1], internal pull-down
X_SSP1_RXD	I, internal pull-up	X_GPIO_1[26], I, pull-up	X_SSP1_RXD, I, pull-up	X_LC_VS, O	-	reg64[7:6], default = 2'b00	Floating	-
X_SSP1_SCLK	I, internal pull-up	X_GPIO_1[27], I, pull-up	X_SSP1_SCLK, O (SSP1 master mode : scu reg64[1:0] = 2'b01, SSP1 slave mode : scu reg64[1:0] = 2'b01 and reg64[8] = 1'b1)	X_LC_HS, O	-	reg64[1:0], default = 2'b00	Floating	-

Table 4-35. RMII MAC Pin-mux

Ball Name	Power-On Reset State	Multi-function Pin and Function Default State				Control Register and Default	Unused PCB Setting and Comment	Comment
		regxx bit[y:x] = 2'b01= 2'b00	regxx bit[y:x] = 2'b01= 2'b01	regxx bit[y:x] = 2'b01= 2'b10	regxx bit[y:x] = 2'b01= 2'b11			
X_RMII_MDIO	I, internal pull-up	X_GPIO_1[7], I, pull-up	X_RMII_MDIO, I, pull-up	X_LC_DATA[0], O	X_TV_DATA[0], O	reg5c[1:0], default = 2'b00	Floating	-

Ball Name	Power-On Reset State	Multi-function Pin and Function Default State				Control Register and Default	Unused PCB Setting and Comment	Comment
		regxx bit[y:x] = 2'b01= 2'b00	regxx bit[y:x] = 2'b01= 2'b01	regxx bit[y:x] = 2'b01= 2'b10	regxx bit[y:x] = 2'b01= 2'b11			
X_RMII_MDC	I, internal pull-up	X_GPIO_1[8], I, pull-up	X_RMII_MDC, O, 0	X_LC_DATA[1], O	X_TV_DATA[1], O	reg5c[3:2], default = 2'b00	Floating	-
X_RMII_RX_ER	I, internal pull-up	X_GPIO_1[12], I, pull-up	X_RMII_RX_ER, I	X_LC_DATA[2], O	X_TV_DATA[2], O	reg5c[11:10], default = 2'b00	Floating	-
X_RMII_RX_CRS_DV	I, internal pull-up	X_GPIO_1[9], I, pull-up	X_RMII_RX_CRS_DV, I, pull-up	X_LC_DATA[3], O	X_TV_DATA[3], O	reg5c[5:4], default = 2'b00	Floating	-
X_RMII_RXD[0]	I, internal pull-up	X_GPIO_1[10], I, pull-up	X_RMII_RXD[0], I, pull-up	X_LC_DATA[4], O	X_TV_DATA[4], O	reg5c[7:6], default = 2'b00	Floating	-
X_RMII_RXD[1]	I, internal pull-up	X_GPIO_1[11], I, pull-up	X_RMII_RXD[1], I, pull-up	X_LC_DATA[5], O	X_TV_DATA[5], O	reg5c[9:8], default = 2'b00	Floating	-
X_RMII_PHYLINK	I, internal pull-up	X_GPIO_1[13], I, pull-up	X_RMII_PHYLINK, I, pull-up	X_LC_DATA[6], O	X_TV_DATA[6], O	reg5c[13:12], default = 2'b00	Floating	-
X_RMII_CKO	I, internal pull-up	X_GPIO_1[14], I, pull-up	X_RMII_CKO, O	X_LC_DATA[7], O	X_TV_DATA[7], O	reg5c[15:14], default = 2'b00	Floating	-
X_RMII_TXD[0]	I, internal pull-up	X_GPIO_1[15], I, pull-up	X_RMII_TXD[0], O	X_LC_DATA[8], O	X_TV_DATA[8], O	reg5c[17:16], default = 2'b00	Floating	-
X_RMII_TXD[1]	I, internal pull-up	X_GPIO_1[16], I, pull-up	X_RMII_TXD[1], O	X_LC_DATA[9], O	X_TV_DATA[9], O	reg5c[19:18], default = 2'b00	Floating	-
X_RMII_TX_EN	I, internal pull-down	X_GPIO_1[17], I, pull-down	X_RMII_TX_EN, O	X_LC_DATA[10], O	X_TV_DATA[10], O	reg5c[21:20], default = 2'b00	Floating	-
X_RMII_RST	I, internal pull-down	X_GPIO_1[18], I, pull-down	X_RMII_CK_IN, I	X_LC_DATA[11], O	X_TV_DATA[11], O	reg5c[23:22], default = 2'b00	Floating	-

Table 4-36. GPIO Pin-mux

Ball Name	Power-On Reset State	Multi-function Pin and Function Default State				Control Register and Default	Unused PCB Setting and Comment	Comment
		regxx bit[y:x] = 2'b00	regxx bit[y:x] = 2'b01	regxx bit[y:x] = 2'b10	regxx bit[y:x] = 2'b11			
X_GPIO_DAT[4]	I, internal pull-up	X_GPIO_1[19], I, pull-up	X_SD1_CD, I, pull-up	X_LC_DATA[12], O	X_TV_DATA[12], O	reg5c[25:24], default = 2'b00	Floating	-
X_GPIO_DAT[5]	I, internal pull-up	X_GPIO_1[20], I, pull-up	X_SD1_DAT[1], I, pull-up	X_LC_DATA[13], O	X_TV_DATA[13], O	reg5c[27:26], default = 2'b00	Floating	-
X_GPIO_DAT[6]	I, internal pull-up	X_GPIO_1[21], I, pull-up	X_SD1_DAT[0], I, pull-up	X_LC_DATA[14], O	X_TV_DATA[14], O	reg5c[29:28], default = 2'b00	Floating	-
X_GPIO_DAT[7]	I, internal pull-up	X_GPIO_1[22], I, pull-up	X_SD1_CLK, O	X_LC_DATA[15], O	X_TV_DATA[15], O	reg5c[31:30], default = 2'b00	Floating	-
X_TV_PCLK	I, internal pull-up	Don't use, I, pull-up	-	X_LC_PCLK, O	X_TV_PCLK, O	reg64[9:8], default = 2'b00	Floating	-
X_GPIO_DAT[0]	I, internal pull-up	X_GPIO_0[28], I, pull-up	X_SSP1_FS, O, pull-up	-	-	reg58[11:10], default = 2'b00	Floating	-
X_GPIO_DAT[1]	I, internal pull-up	X_GPIO_0[29], I, pull-up	X_SSP1_TXD, I, pull-up	X_SD1_CMD_RSP, O, 1, pull-up	-	reg58[13:12], default = 2'b00	Floating	-
X_GPIO_DAT[2]	I, internal pull-up	X_GPIO_0[30], I, pull-up	X_SSP1_RXD, I, pull-up	X_SD1_DAT[3], I, pull-up	X_OTG_48M_dbg, O	reg5[15:14], default = 2'b00	Floating	-
X_GPIO_DAT[3]	I, internal pull-up	X_GPIO_0[31], I, pull-up	X_SSP1_SCLK, O	X_SD1_DAT[2], I, pull-up	X_OTG_30M_dbg, O	reg58[17:16], default = 2'b00	Floating	-

Table 4-37. ICE Pin-mux

Ball Name	Power-On Reset State	Multi-function Pin and Function Default State				Control Register and Default	Unused PCB Setting and Comment	Comment
		regxx bit[y:x] = 2'b00	regxx bit[y:x] = 2'b01	regxx bit[y:x] = 2'b10	regxx bit[y:x] = 2'b11			
X_NTRST	I, internal pull-down	X_NTRST, I, pull-down	X_GPIO_0[0], I, pull-down	-	-	reg50[1:0], default = 2'b00	Floating	-
X_TDI	I, internal pull-up	X_TDI, I, pull-up	X_GPIO_0[4], I, pull-up	X_PWM7, O	X_DMIC_CLK, O	reg50[9:8], default = 2'b00	Floating	-
X_TMS	I, internal pull-up	X_TMS, I, pull-up	X_GPIO_0[2], I, pull-up	X_PWM5, O	X_UART2_SOUT, O	reg50[5:4], default = 2'b00	Floating	-

Ball Name	Power-On Reset State	Multi-function Pin and Function Default State				Control Register and Default	Unused PCB Setting and Comment	Comment
		regxx bit[y:x] = 2'b00	regxx bit[y:x] = 2'b01	regxx bit[y:x] = 2'b10	regxx bit[y:x] = 2'b11			
X_TCK	I, internal pull-up	X_TCK, I, pull-up	X_GPIO_0[1], I, pull-up	X_PWM4, O	X_UART2_SIN, I, PU	reg50[3:2], default = 2'b00	Floating	-
X_TDO	I, internal pull-up	X_TDO, I, pull-up	X_GPIO_0[3], I, pull-up	X_PWM6, O	X_DMIC_DATA, I, PU	reg50[7:6], default = 2'b00	Floating	-

Table 4-38. UART0 Pin-mux

Ball Name	Power-On Reset State	Multi-function Pin and Function Default State				Control Register and Default	Unused PCB Setting and Comment	Comment
		regxx bit[y:x] = 2'b00	regxx bit[y:x] = 2'b01	regxx bit[y:x] = 2'b10	regxx bit[y:x] = 2'b11			
X_UART0_SIN	I, internal pull-up	X_GPIO_1[5], I, pull-up	X_UART0_SIN, I, pull-up	-	-	reg58[29:28], default = 2'b00	Floating	
X_UART0_SOUT	I, internal pull-up	X_GPIO_1[6], I, pull-down	X_UART0_SOUT, O, 1, pull-up	-	-	reg58[31:30], default = 2'b00	Jumper setting 7, internal pull-down	

4.1.4.2.24 SPI x-bit Count Setting Register (Offset = 0x6C)

Table 4-39. SPI x-bit Count Setting Register (Offset = 0x6C)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:18]	-	-	Reserved	-	-
[17:16]	cap_clko_sscg_ mr	R/W	SSCG (Spread Spectrum Clock Generator) modulation ratio selection for the capture clock output	0x0	HR
[15:3]	-	-	Reserved	-	-
[2:0]	spi_clk_div	R/W	SPI clock divided value and SPI frequency selection SPI = After reg28[10] selection/(spi_clk_div + 1)	0x4	HR

4.1.4.2.25 SDC/MAC/UART/RTC x-bit Count Setting Register (Offset = 0x70)

Table 4-40. SDC/MAC/UART/RTC x-bit Count Setting Register (Offset = 0x70)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:24]	rtc_clk_div	R/W	RTC clock divided value and RTC frequency selection RTC = After reg28[0] selection/(rtc_clk_div + 1)	0xb6	HR
[23:22]	-	-	Reserved	-	-
[21:16]	uart_clk_div	R/W	UART clock divided value and UART frequency selection UART = After reg28[3] selection/(uart_clk_div + 1)	0x09	HR
[15:12]	-	-	Reserved	-	-
[11:8]	rmii_mac_clk_div	R/W	RMII MAC PHY clock divided value and RMII MAC PHY frequency selection RMII MAC PHY clock = After reg28[5:4] selection/(rmii_mac_clk_div + 1)	0x1	HR
[7:4]	-	-	Reserved	0x4	HR
3	-	-	Reserved	-	-
[2:0]	sdc_clk_div	R/W	SDC clock divided value and SDC frequency selection SDC = After reg28[7] selection/(sdc_clk_div + 1)	0x4	HR

4.1.4.2.26 SSP/ADC/ADDA x-bit Count Setting Register (Offset = 0x74)

Table 4-41. SSP/ADC/ADDA x-bit Count Setting Register (Offset = 0x74)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:30]	-	-	Reserved	-	-
[29:24]	adda_clk_div	R/W	ADDA clock divided value and ADDA frequency selection ADDA = After reg28[16:15] selection/(adda_clk_div+1)	0x2c	HR
[23:22]	-	-	Reserved	-	-
[21:16]	adc_clk_div	R/W	ADC clock divided value and ADC frequency selection ADC = OSCH in/(adc_clk_div + 1)	0x04	HR
[15:14]	-	-	Reserved	-	-
[13:8]	ssp1_clk_div	R/W	SSP1 clock divided value and SSP1 frequency selection SSP1: After reg28[14:13] selection/(ssp1_clk_div+1)	0x2c	HR
[7:6]	-	-	Reserved	-	-

Bit	Name	Type	Description	Reset Value	Reset Type
[5:0]	ssp0_clk_div	R/W	SSP0 clock divided value and SSP0 frequency selection SSP0: After reg28[12:11] selection/(ssp0_clk_div + 1)	0x2c	HR

4.1.4.2.27 EXT/Panel Pixel/LCDC Scalar x-bit Count Setting Register (Offset = 0x78)

Table 4-42. EXT/Panel Pixel/LCDC Scalar x-bit Count Setting Register (Offset = 0x78)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:30]	-	-	Reserved	-	-
[29:24]	serial_phy_clk_div	R/W	Serial Combo PHY clock divided value and serial Combo PHY frequency selection It needs to be fixed at 27MHz. Serial Combo PHY = After reg28[6] selection/(serial_phy_clk_div + 1)	0x09	HR
[23:22]	-	-	Reserved	-	-
[21:16]	ext_clk_div	R/W	External clock (Capture clock out) divided value and external frequency selection External = After reg28[2:1] selection/(ext_clk_div + 1)	0x13	HR
[15:14]	-	-	Reserved	-	-
[13:8]	lcd_sca_div	R/W	LCD scaler clock divided value and LCD scaler frequency selection LCD scaler = After reg28[9] selection/(lcd_sca_div + 1)	0x13	HR
[7:6]	-	-	Reserved	-	-
[5:0]	lcd_pclk_div	R/W	LCD pixel clock divided value and LCD pixel frequency selection LCD pixel = After reg28[8] selection/(lcd_pclk_div + 1)	0x13	HR

4.1.4.2.28 System Control Register (Offset = 0x7C)

Table 4-43. System Control Register (Offset = 0x7C)

Bit	Name	Type	Description	Reset Value	Reset Type
31	ddr3_phy_dllrst_n	RO	DDR3 PHY reset status 0: Reset 1: Active	0x1	HR
30	-	-	Reserved	-	-

Bit	Name	Type	Description	Reset Value	Reset Type
29	ssp1_clk_sel	R/W	SSP1 clock source selection 0: ADDA (Audio DA part) 1: External pin (X_SSP1_SCLK)	0x0	HR
28	-	-	Reserved	-	-
27	Audio DA source select	R/W	Audio DA source selection 0: SSP0 1: SSP1	0x0	HR
26	lc_pclk_inv	R/W	LC_PCLK and TV_PCLK inverse 0: Default clock 1: Inverse clock	0x0	HR
25	dphy_csi_clk_select	R/W	dphy_csi_clk_selection 0: dphy_byte_clk (From the serial Combo PHY) 1: ext_clk_div[After reg28[2:1] selection/(ext_clk_div + 1)]	0x0	HR
24	bayer_clk_inv	R/W	bayer_clk_inv 0: Default clock 1: Inverse clock	0x0	HR
23	isp_cap1_clk_sel	R/W	isp_cap1_clk_sel 0: isp_mclk (After reg28[22:21] 1: sensor_clk (After reg7c[24])	0x0	HR
22	sensor_mode_sel	R/W	Select the sensor mode 0: Serial mode 1: Parallel mode	0x0	HR
[21:20]	bayer_src_sel	R/W	Select the Bayer data source 0x0: Serial MIPI 14bit 0x1: Reserved 0x2: Serial subLVDS 14bit 0x3: Parallel 12bit	0x0	HR
19	cap1_data_swap	R/W	cap1_data_swap 0: cap1 data order 1: Swap cap1 data order	-	-
18	sensor_vs_inv	R/W	Inverse sensor V-sync polarity 0: Default V-sync polarity 1: Inverse V-sync	0x0	HR
17	sensor_hs_inv	R/W	Inverse sensor H-sync polarity 0: Default H-sync polarity 1: Inverse H-syan polarity	0x0	HR

Bit	Name	Type	Description	Reset Value	Reset Type
16	bayer_data_swap	R/W	Input parallel Bayer data swap 0: Default Bayer data order 1: Swap Bayer data order	0x0	HR
[15:14]	-	-	Reserved	-	-
13	bayer_14b_in	R/W	bayer_14b_in 0: 12bit Bayer data 1: 14bit Bayer data	0x0	HR
12	cap0_clk_inv	R/W	cap0_clk_inv 0: Default cap0_clk 1: Inverse cap0_clk	0x0	HR
11	-	-	Reserved	-	-
[10:8]	cap0_data_swap	R/W	cap0_data_swap 0x1: cap0 low 8bit internal swap 0x2: cap0 high 8bit swap 0x4: cap0 high/low byte swap	0x0	HR
7	-	-	Reserved	-	-
6	X_RMII_CKO phase select	R/W	X_RMII_CKO phase selection 0: Positive 1: Invert	0x0	HR
5	MAC interface type selection	R/W	MAC interface type selection 0: RMII mode (Only support the RMII mode) 1: RGMII mode	0x0	HR
4	auto watch dog enable	R	Auto watchdog count enable (Set by jumper[8]) 0: Auto count 1: Reset counter Application: For pre-heat enable	-	-
3	-	-	Reserved	-	-
2	-	-	Reserved	-	-
[1:0]	software interrupt	R/W	Software interrupt at INTC port [62:61]	0x0	HR

4.1.4.2.29 DDR3/DDR2 PHY Power Control Setting Register (Offset = 0x80)

Table 4-44. DDR3/DDR2 PHY Power Control Setting Register (Offset = 0x80)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:8]	-	-	Reserved	-	-
7	CSysReq	R/W	System low-power request on the DDR controller 0: Low power request	0x1	HR
[6:5]	-	-	Reserved	-	-
4	ddr3_dllrst_n	R/W	0: DLL of DDR3/DDR2 is reset. 1: DLL of DDR3/DDR2 is not reset.	0x1	HR
3	ddr3_dllpdn	R/W	DLL of DDR3/DDR2 power-down control	0x1	HR
2	-	-	Reserved	-	-
1	ddr3_pllrst_n	R/W	0: PLL of DDR3/DDR2 is reset. 1: PLL of DDR3/DDR2 is not reset.	0x1	HR
0	ddr3_pllpdn	R/W	PLL of DDR3/DDR2 power-down control	0x1	HR

4.1.4.2.30 THDNR Clock Control Register (Offset = 0x88)

Table 4-45. THDNR Clock Control Register (Offset = 0x88)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:16]	pwm_period	R/W	PWM period If setting to 1, it will be gating 15 clock cycles.	0x0000	HR
[15:3]	-	-	Reserved	-	-
[2:1]	mode_sel	R/W	Mode selectiion 0: Always on 1: PWM mode 2 or 3: Always off	0x0	HR
0	reg_set	R/W	If setting to 1, it will update the mode_sel and PWM periods (Self-cleared).	0x0	HR

4.1.4.2.31 H264 Encoder Clock Control Register (Offset = 0x8C)

Table 4-46. H264 Encoder Clock Control Register (Offset = 0x8C)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:16]	pwm_period	R/W	PWM period If setting to 1, it will be gating 15 clock cycles.	0x0000	HR
[15:3]	-	-	Reserved	-	-
[2:1]	mode_sel	R/W	Mode selection 0: Always on 1: PWM mode 2 or 3: Always off	0x0	HR
0	reg_set	R/W	If setting to 1, it will update the mode_sel and PWM periods (Self-cleared).	0x0	HR

4.1.4.2.32 ISP Clock Control Register (Offset = 0x90)

Table 4-47. ISP Clock Control Register (Offset = 0x90)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:16]	pwm_period	R/W	PWM period If setting to 1, it will be gating 15 clock cycles.	0x0000	HR
[15:3]	-	-	Reserved	-	-
[2:1]	mode_sel	R/W	Mode selection 0: Always on 1: PWM mode 2 or 3: Always off	0x0	HR
0	reg_set	R/W	If setting to 1 ,it will update the mode_sel and PWM periods (Self-cleared).	0x0	HR

4.1.4.2.33 GE Clock Control Register (Offset = 0x94)

Table 4-48. GE Clock Control Register (Offset = 0x94)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:16]	pwm_period	R/W	PWM period If setting to 1, it will be gating 15 clock cycles.	0x0000	HR

Bit	Name	Type	Description	Reset Value	Reset Type
[15:3]	-	-	Reserved	-	-
[2:1]	mode_sel	R/W	Mode selectiion 0: Always on 1: PWM mode 2 or 3: Always off	0x0	HR
0	reg_set	R/W	If setting to 1, it will update the mode_sel and PWM periods (Self-cleared).	0x0	HR

4.1.4.2.34 MCP Main Clock Control Register (Offset = 0x98)

Table 4-49. MCP Main Clock Control Register (Offset = 0x98)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:16]	pwm_period	R/W	PWM period If setting to '1', it will gate 15 clock cycles.	0x0000	HR
[15:3]	-	-	Reserved	-	-
[2:1]	mode_sel	R/W	Mode selection 0: Always on 1: PWM mode 2 or 3: Always off	0x0	HR
0	reg_set	R/W	If setting to '1', it will update the mode_sel and PWM periods (Self-cleared).	0x0	HR

4.1.4.2.35 DI Main Clock Control Register (Offset = 0x9C)

Table 4-50. DI Main Clock Control Register (Offset = 0x9C)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:16]	pwm_period	R/W	PWM period If setting to '1', it will be gate 15 clock cycles.	0x0000	HR
[15:3]	-	-	Reserved	-	-
[2:1]	mode_sel	R/W	Mode selection 0: Always on 1: PWM mode 2 or 3: Always off	0x0	HR

Bit	Name	Type	Description	Reset Value	Reset Type
0	reg_set	R/W	If setting to '1', it will update the mode_sel and PWM periods (Self-cleared).	0x0	HR

4.1.4.2.36 IP Software Reset Control Register 0 (Offset = 0xA0)

Table 4-51. IP Software Reset Control Register 0 (Offset = 0xA0)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:26]	-	-	Reserved	-	-
25	CSIRX	R/W	MIPI controller reset 0: Reset is active. 1: Reset is not active.	0x1	HR
24	DPHY	R/W	DPHY(Serial Combo PHY for MIPI or subLVDS) reset 0: Reset is active. 1: Reset is not active.	0x1	HR
23	subLVDS	R/W	subLVDS controller reset 0: Reset is active. 1: Reset is not active.	0x0	HR
[22:19]	-	-	Reserved	-	-
18	RMII MAC	R/W	RMII MAC reset 0: Reset is active. 1: Reset is not active.	0x1	HR
17	AES	R/W	AES reset 0: Reset is active. 1: Reset is not active.	0x1	HR
16	MCP	R/W	MPEG4/JPEG codec reset 0: Reset is active. 1: Reset is not active.	0x1	HR
15	-	-	Revered	0x1	HR
14	IVS	R/W	IVS (hclk) reset 0: Reset is active. 1: Reset is not active.	0x1	HR
13	-	-	Reserved	-	-
12	OTG 1	R/W	OTG 1 (OTG 1.1) controller reset 0: Reset is active. 1: Reset is not active.	0x1	HR

Bit	Name	Type	Description	Reset Value	Reset Type
11	OTG 0	R/W	OTG 0(OTG 2.0) controller reset 0: Reset is active. 1: Reset is not active.	0x1	HR
10	LCDC	R/W	RLC decoder and LCD controller reset 0: Reset is active. 1: Reset is not active.	0x1	HR
9	rlc_enc	R/W	RLC encoder reset 0: Reset is active. 1: Reset is not active.	0x1	HR
8	-	-	Reserved	-	-
7	SCALER	R/W	Scaler reset 0: Reset is active. 1: Reset is not active.	0x1	HR
6	-	-	Reserved	-	-
5	DI	R/W	De-interlace reset 0: Reset is active. 1: Reset is not active.	0x1	HR
4	THDNR	R/W	3D de-noise reset 0: Reset is active. 1: Reset is not active.	0x1	HR
3	ISP	R/W	ISP reset 0: Reset is active. 1: Reset is not active.	0x1	HR
2	-	-	Reserved	-	-
1	H264_enc	R/W	H.264 encoder reset 0: Reset is active. 1: Reset is not active.	0x1	HR
0	VCAP	R/W	capture input module reset 0: Reset is active. 1: Reset is not active.	0x1	HR

4.1.4.2.37 IP Software Reset Control Register 1 (Offset = 0xA4)

Table 4-52. IP Software Reset Control Register 1 (Offset = 0xA4)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:5]	-	-	Reserved	-	-
4	DDRx main clock	R/W	DDR3/DDR2 controller to reset main clock domain 0: Reset is active. 1: Reset is not active.	0x1	HR
3	DDRx CH3	R/W	DDR3/DDR2 controller to reset channel 3 0: Reset is active. 1: Reset is not active.	0x1	HR
2	DDRx CH2	R/W	DDR3/DDR2 controller to reset channel 2 0: Reset is active. 1: Reset is not active.	0x1	HR
1	DDRx CH1	R/W	DDR3/DDR2 controller to reset channel 1 0: Reset is active. 1: Reset is not active.	0x1	HR
0	DDRx CH0	R/W	DDR3/DDR2 controller to reset channel 0 0: Reset is active. 1: Reset is not active.	0x1	HR

4.1.4.2.38 Software Interrupt Set/Clear Control Register (Offset = 0xA8)

Table 4-53. Software Interrupt Set/Clear Control Register (Offset = 0xA8)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:0]	soft_int_set_clr	R/W	Software interrupt set/clear control Write pulse: [31:16]: Clear interrupt[91:76] [15:0]: Set interrupt[91:76] Read: [31:16]: 0 [15:0]: Status of interrupt[91:76]	0x0	HR

4.1.4.2.39 Hardcore Clock Off Control Register (Offset = 0xAC)

Table 4-54. Hardcore Clock Off Control Register (Offset = 0xAC)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:16]	-	-	Reserved	-	-
16	EXT_SSCG	R/W	External SSCG clock 0: On 1: Off	0x1	HR
[15:10]	-	-	Reserved	-	-
9	ISP	R/W	ISP pixel clock 0: On 1: Off	0x1	HR
8	subLVDS	R/W	subLVDS pixel clock 0: On 1: Off	0x1	HR
7	-	-	Reserved	-	-
6	MIPI	R/W	MIPI pixel clock 0: On 1: Off	0x1	HR
5	X_RMII_CKO	R/W	X_RMII_CKO (RMII MAC PHY clock out) 0: On 1: Off	0x1	HR
4	External clock	R/W	External clock (Capture clock out) 0: On 1: Off	0x1	HR
3	SAR ADC	R/W	SAR ADDA clock 0: On 1: Off	0x1	HR
2	ADDA	R/W	ADDA clock 0: On 1: Off	0x1	HR
1	DPHY	R/W	DPHY (Serial Combo PHY) 0: On 1: Off	0x1	HR
0	OTG 0 PHY	R/W	OTG 0 PHY clock 0: On 1: Off	0x1	HR

4.1.4.2.40 AXI Module Clock Off Control Register (AXIMCLKOFF, Offset = 0xB0)

Table 4-55. AXI Module Clock Off Control Register (AXIMCLKOFF, Offset = 0xB0)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:18]	-	-	Reserved	-	-
17	VCAP	R/W	Capture clock 0: On 1: Off	0x1	HR
16	ISP	R/W	ISP clock 0: On 1: Off	0x1	HR
15	LCDC	R/W	RLC decoder and LCD controller clock 0: On 1: Off	0x1	HR
14	SCALER	R/W	Scalar clock 0: On 1: Off	0x1	HR
13	H264_enc	R/W	H.264 encoder clock 0: On 1: Off	0x1	HR
12	rlc encoder	R/W	RLC encoder clock 0: On 1: Off	0x1	HR
11	THDNR	R/W	3D de-noise clock 0: On 1: Off	0x1	HR
10	DI	R/W	De-interlace clock 0: On 1: Off	0x1	HR
9	IVS	R/W	IVS clock 0: On 1: Off	0x1	HR
8	INTC	R/W	Interrupt controller clock 0: On 1: Off	0x1	HR

Bit	Name	Type	Description	Reset Value	Reset Type
7	DDR3/DDR2	R/W	DDR controller clock 0: On 1: Off	0x0	HR
[6:3]	-	-	Reserved	-	-
2	X2P_0	R/W	AXI-to-APB 0 clock 0: On 1: Off	0x0	HR
1	X2H_1	R/W	AXI-to-AHB 1 clock 0: On 1: Off	0x0	HR
0	X2H_0	R/W	AXI-to-AHB 0 clock 0: On 1: Off	0x0	HR

4.1.4.2.41 AHB Module Clock Off Control Register (AHBMCLKOFF, Offset = 0xB4)

Table 4-56. AHB Module Clock Off Control Register (AHBMCLKOFF, Offset = 0xB4)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:23]	-	-	Reserved	-	-
22	SDC_1	R/W	SDC 1 controller clock 0: On 1: Off	0x1	HR
21	ISP	R/W	ISP register programming clock 0: On 1: Off	0x1	HR
20	TVE	R/W	TV encoder register programming clock 0: On 1: Off	0x1	HR
19	H264_enc	R/W	H.264 encoder register programming clock 0: On 1: Off	0x1	HR
18	DDRx CH1	R/W	DDR3 controller channel1 clock 0: On 1: Off	0x1	HR

Bit	Name	Type	Description	Reset Value	Reset Type
17	ROMIF	R/W	ROM clock 0: On 1: Off	0x0	HR
16	OTG_1	R/W	OTG 1 (OTG 1.1) controller clock 0: On 1: Off	0x1	HR
15	SPI	R/W	SPI controller clock 0: On 1: Off	0x1	HR
14	SDC	R/W	SDC controller clock 0: On 1: Off	0x1	HR
13	ge	R/W	Graphic engine controller clock 0: On 1: Off	0x1	HR
12	MCP	R/W	MPEG4/JPEG codec register programming clock 0: On 1: Off	0x1	HR
11	RMII MAC	R/W	RMII MAC controller clock 0: On 1: Off	0x0	HR
10	AES	R/W	Encryption controller clock 0: On 1: Off	0x1	HR
9	OTG_0	R/W	OTG 0 (OTG 2.0) controller clock 0: On 1: Off	0x1	HR
8	DMA	R/W	DMA clock 0: On 1: Off	0x1	HR
[7:3]	-	-	Reserved	-	-
2	HCLK_1	R/W	AHB 1 clock 0: On 1: Off	0x0	HR
1	X2H_1	R/W	AXI-to-AHB 1 clock 0: On 1: Off	0x0	HR

Bit	Name	Type	Description	Reset Value	Reset Type
0	X2H_0	R/W	AXI-to-AHB 0 clock 0: On 1: Off	0x0	HR

4.1.4.2.42 APB Module Clock Off Control Register 0 (APBMCLKOFF0, Offset = 0xB8)

Table 4-57. APB Module Clock Off Control Register 0 (APBMCLKOFF0, Offset = 0xB8)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:21]	-	-	Reserved	-	-
20	APBBRG	R/W	APB bridge clock 0: On 1: Off	0x0	HR
19	-	-	Reserved	-	-
18	-	-	Reserved	-	-
17	GPIO_1	R/W	GPIO 1 clock 0: On 1: Off	0x1	HR
16	GPIO_0	R/W	GPIO 0 clock 0: On 1: Off	0x1	HR
15	-	-	Reserved	-	-
14	RTC	R/W	RTC clock 0: On 1: Off	0x1	HR
13	TMR_2	R/W	Timer 2 clock 0: On 1: Off	0x1	HR
12	TMR_1	R/W	Timer 1 clock 0: On 1: Off	0x1	HR
11	TMR_0	R/W	Timer 0 clock 0: On 1: Off	0x1	HR

Bit	Name	Type	Description	Reset Value	Reset Type
10	WDT	R/W	WatchDog timer clock 0: On 1: Off	0x1	HR
9	PWM	R/W	PWM clock 0: On 1: Off	0x1	HR
8	adc_wrapper	R/W	ADC wrapper clock 0: On 1: Off	0x1	HR
7	IIC	R/W	I^2C clock 0: On 1: Off	0x1	HR
6	adda_wrapper	R/W	ADDA wrapper clock 0: On 1: Off	0x1	HR
5	SSP_1	R/W	SSP 1 clock 0: On 1: Off	0x1	HR
4	SSP_0	R/W	SSP 0 clock 0: On 1: Off	0x1	HR
3	irdet	R/W	IRDA detector clock 0: On 1: Off	0x1	HR
2	UART_2	R/W	UART 2 clock 0: On 1: Off	0x1	HR
1	UART_1	R/W	UART 1 clock 0: On 1: Off	0x1	HR
0	UART_0	R/W	UART 0 clock 0: On 1: Off	0x1	HR

4.1.4.2.43 APB Module Clock Off Control Register 1 (APBMCLKOFF1, Offset = 0xBC)

Table 4-58. APB Module Clock Off Control Register 1 (APBMCLKOFF1, Offset = 0xBC)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:21]	-	-	Reserved	-	-
20	subLVDS	R/W	subLVDS register programming clock 0: On 1: Off	0x1	HR
19	-	-	Reserved	-	-
18	CSIRX	R/W	CSIRX register programming clock 0: MIPI is On. 1: MIPI is Off.	0x1	HR
17	LCDC	R/W	LCDC register programming clock 0: On 1: Off	0x1	HR
16	rlc_dec	R/W	RLC decoder register programming clock 0: On 1: Off	0x1	HR
15	rlc_enc	R/W	RLC encoder register programming clock 0: On 1: Off	0x1	HR
14	DI	R/W	DI (De-interlace) register programming clock 0: On 1: Off	0x1	HR
13	THDNR	R/W	3D de-noise register programming clock 0: On 1: Off	0x1	HR
12	IVS	R/W	IVS register programming clock 0: On 1: Off	0x1	HR
[11:9]	-	-	Reserved	-	-
8	DDRx	R/W	DDR3/DDR2 control register programming clock 0: On 1: Off	0x0	HR
7	-	-	Reserved	-	-

Bit	Name	Type	Description	Reset Value	Reset Type
6	AXIC_2	R/W	AXI controller 2 register programming clock 0: On 1: Off	0x0	HR
5	AXIC_1	R/W	AXI controller 1 register programming clock 0: on 1: off	0x0	HR
4	AXIC_0	R/W	AXI controller 0 register programming clock 0: On 1: Off	0x0	HR
3	-	-	Reserved	-	-
2	X2P_0	R/W	AXI-to-APB 0 register programming clock 0: On 1: Off	0x0	HR
1	X2H_1	R/W	AXI-to-AHB 1 register programming clock 0: On 1: Off	0x0	HR
0	X2H_0	R/W	AXI-to-AHB 0 register programming clock 0: On 1: Off	0x0	HR

4.1.4.2.44 OTG 0 PHY/Controller and OTG1 Controller Control Register 0 (OTGPHY0, Offset = 0xC0)

Table 4-59. OTG 0 PHY/Controller and OTG 1 Controller Control Register 0 (OTGPHY0, Offset = 0xC0)

Bit	Name	Type	Description	Reset Value	Reset Type
31	OTG_RS2	R/W	PLL loop filter resistor setting used for adjusting BW	0x0	HR
30	OTG_RS1	R/W	PLL loop filter resistor setting used for adjusting BW	0x0	HR
29	OTG_IS2	R/W	PLL charge pump current setting used for adjusting BW	0x0	HR
28	OTG_IS1	R/W	PLL charge pump current setting used for adjusting BW	0x0	HR
27	OTG_TXSW1	R/W	Output swing-level setting	0x0	HR
26	OTG_TXSW0	R/W	Output swing-level setting	0x0	HR

<table>
<tr><th>Bit</th><th>Name</th><th>Type</th><th>Description</th><th>Reset Value</th><th>Reset Type</th></tr>
<tr><td>25</td><td>OTG_PCR2</td><td>R/W</td><td>High-speed transmitter current-level setting
<table>
<tr><th>PCR2</th><th>PCR1</th><th>Current</th></tr>
<tr><td>0</td><td>0</td><td>100µA</td></tr>
<tr><td>0</td><td>1</td><td>90µA</td></tr>
<tr><td>1</td><td>0</td><td>120µA</td></tr>
<tr><td>1</td><td>1</td><td>110µA</td></tr>
</table></td><td>0x1</td><td>HR</td></tr>
<tr><td>24</td><td>OTG_PCR1</td><td>R/W</td><td>High-speed transmitter current-level setting</td><td>0x1</td><td>HR</td></tr>
<tr><td>23</td><td>OTG_PREEMPS2</td><td>R/W</td><td>3bit control signal of the pre-emphasis strength
{OTG_PREEMPS2, OTG_PREEMPS1, OTG_PREEMPS0}
0x0: None
0x7: Strongest</td><td>0x0</td><td>HR</td></tr>
<tr><td>22</td><td>OTG_PREEMPS1</td><td>R/W</td><td>3bit control signal of the pre-emphasis strength</td><td>0x0</td><td>HR</td></tr>
<tr><td>21</td><td>OTG_PREEMPS0</td><td>R/W</td><td>3bit control signal of the pre-emphasis strength</td><td>0x0</td><td>HR</td></tr>
<tr><td>20</td><td>OTG_PREEMPS_EN</td><td>R/W</td><td>Pre-emphasis function control
0: Disable
1: Enable</td><td>0x0</td><td>HR</td></tr>
<tr><td>19</td><td>OTG_PHY_OUTCLKSEL</td><td>R/W</td><td>0: Select the crystal as the clock source
1: Select the CORECLKIN signal from customer as the clock source</td><td>0x1</td><td>HR</td></tr>
<tr><td>[18:15]</td><td>-</td><td>-</td><td>Reserved</td><td>-</td><td>-</td></tr>
<tr><td>14</td><td>OTG_SQREF2</td><td>R/W</td><td>Control signal of the reference voltage in the squelch circuit
<table>
<tr><th>SQREF2</th><th>SQREF1</th><th>SQREF0</th><th>Voltage</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>115mV</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>77mV</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>145mV</td></tr>
</table></td><td>0x0</td><td>HR</td></tr>
<tr><td>13</td><td>OTG_SQREF1</td><td>R/W</td><td>Control signal of the reference voltage in the squelch circuit</td><td>0x1</td><td>HR</td></tr>
<tr><td>12</td><td>OTG_SQREF0</td><td>R/W</td><td>Control signal of the reference voltage in the squelch circuit</td><td>0x1</td><td>HR</td></tr>
<tr><td>11</td><td>-</td><td>-</td><td>Reserved</td><td>-</td><td>-</td></tr>
</table>

Bit	Name	Type	Description	Reset Value	Reset Type
10	OTG_HD2	R/W	Control signal of the reference voltage in the host disconnect detect circuit HD2 / HD1 / HD0 / Voltage 0 / 0 / 0 / 574mV 0 / 1 / 1 / 629mV 1 / 0 / 0 / 503mV 1 / 1 / 1 / 557mV	0x0	HR
9	OTG_HD1	R/W	Control signal of the reference voltage in the host disconnect detect circuit	0x0	HR
8	OTG_HD0	R/W	Control signal of the reference voltage in the host disconnect detect circuit	0x0	HR
7	-	-	Reserved	-	-
6	OTG_PONRST	R/W	0: Digital power is not ready. 1: Digital power is ready.	0x0	HR
5	OTG_PHY_OSCOUTEN	R/W	CLKOSCO clock output selection 0: CLKOSCO is disabled. 1: CLKOSCO is enabled.	0x0	HR
4	OTG_PHY_PLLALIV	R/W	When PLLALIV = '1', the suspended PLL will be alive, while SUSCLK120/SUSCLK480 and CLK48M will generate the clock.	0x1	HR
3	OTG_VBUS	R/W	OTG 0 controller VBUS input	0x1	HR
2	OTG_IDDIG	R/W	OTG 0 controller IDDIG	0x1	HR
1	OTG_1_VBUS	R/W	OTG 1 controller VBUS input	0x0	HR
0	OTG_1_IDDIG	R/W	OTG 1 controller IDDIG	0x0	HR

4.1.4.2.45 OTG 0 PHY Control Register 1 (OTGPHY0, Offset = 0xC4)

Table 4-60. OTG 0 PHY Control Register 1 (OTGPHY1, Offset = 0xC4)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:21]	-	-	Reserved	-	-
20	otg_u_vctload_n	R/W	Vctload signal Active low	0x0	HR
[19:15]	otg_u_vctl[4:0]	R/W	Vendor control Vendor-defined 5bit parallel input bus	0x00	HR
[14:13]	otg_u_xcvrsel[1:0]	R/W	This signal selects the LS, FS, or HS transceiver. 0x0: Enable the High-Speed transceiver 0x1: Enable the Full-Speed transceiver 0x2: Enable the Low-Speed transceiver 0x3: Send an LS packet on the FS bus or receive an LS packet	0x0	HR
12	otg_u_termsel	R/W	This signal selects the FS or HS termination 0: Enable the HS termination 1: Enable the FS termination	0x0	HR
11	otg_u_susp_n	R/W	Suspend signal Active low	0x0	HR
[10:9]	otg_u_opmode[1:0]	R/W	This signal selects a mode from various operation modes. 0x0: Normal operation 0x1: Non-driving 0x2: Disable bitstuffing and NRZI encoding 0x3: Normal operation without automatic generation of the SYNC and EOP patterns	0x0	HR
8	otg_u_dppuldn	R/W	This signal controls the 15kΩ pull-down resistor on the DP line. 0: Pull-down resistor is not connected to the DP line. 1: Pull-down resistor is connected to the DP line.	0x0	HR
7	otg_u_dmpuldn	R/W	This signal controls the 15kΩ pull-down resistor on the DM line. 0: Pull-down resistor is not connected to the DM line. 1: Pull-down resistor is connected to the DM line.	0x0	HR

Bit	Name	Type	Description	Reset Value	Reset Type
6	otg_u_txbitstfen	R/W	This signal indicates that the data on DATAIN[7:0] need to be bitstuffed or not. 0: Bitstuffing is disabled. 1: Bitstuffing is enabled.	0x0	HR
5	otg_u_txbitstfenh	R/W	This signal indicates that the data on DATAIN[15:8] need to be bitstuffed or not. 0: Bitstuffing is disabled. 1: Bitstuffing is enabled. This signal is only required when the 16bit mode is selected.	0x0	HR
4	otg_u_fslsserial	R/W	0: FS and LS packets are sent by using the parallel interface. 1: FS and LS packets are sent by using the serial interface.	0x0	HR
3	otg_u_fslstxen_n	R/W	This signal enables the active-low output.	0x0	HR
2	otg_u_fslstxdata	R/W	Differential data at the D+/D- output	0x0	HR
1	otg_u_fslstxse0	R/W	This signal forces single-ended data to zero.	0x0	HR
0	control enable	R/W	0: Control is not enabled. 1: Control is enabled.	0x0	HR

4.1.4.2.46 Software Register 0 (Offset = 0xD0)

Table 4-61. Software Register 0 (Offset = 0xD0)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:0]	Software register 0	R/W	Reserved for software register 0	0x0	HR

4.1.4.2.47 Software Register 1 (Offset = 0xD4)

Table 4-62. Software Register 1 (Offset = 0xD4)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:0]	Software register 1	R/W	Reserved for software register 1	0x0	HR

4.1.4.2.48 Software Register 2 (Offset = 0xD8)

Table 4-63. Software Register 2 (Offset = 0xD8)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:0]	Software register 2	R/W	Reserved for software register 2	0x0	HR

4.1.4.2.49 Software Register 3 (Offset = 0xDC)

Table 4-64. Software Register 3 (Offset = 0xDC)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:0]	Software register 3	R/W	Reserved for software register 3	0x0	HR

4.2 Interrupt Controller

4.2.1 Features

- Supports 16 software-generation interrupts
- Supports 93 shared peripheral interrupts
- Provides edge-triggered and level-triggered interrupt sources in positive or negative directions
- Independently programmable “ON” or “OFF” de-bounce circuit for each interrupt source
- Provides interrupt request signal to host controller to be configured to active high or active low

4.2.2 Interrupt Routing

The active-high IRQIN input signals indicate that a peripheral device requests an interrupt to the interrupt controller. The interrupt source assignment of spi[92:0] is listed in Table 4-65.

Table 4-65. Interrupt Routing (For cpun_irq_out_r)

Bit	Device	Condition
0	DPHY	Active high, level-triggered Default is 1'b1 when D-PHY is powered off.
1	DMAC	Active high, level-triggered
2	-	-
3	RMII MAC	Active high, level-triggered
4	GE (Graphic Engine)	Active high, level-triggered
5	X2P_0	Active high, level-triggered
6	SSP_0	Active high, level-triggered
7	-	-
8	SCU	Active high, level-triggered
9	OTG_0	Active high, level-triggered
10	OTG_1	Active high, level-triggered
11	SSP_1	Active high, level-triggered
12	SCALAR	Active high, level-triggered
13	GPIO_0	Active high, level-triggered
14	TIMER_0	Active high, level-triggered
15	SDC_0	Active high, level-triggered
16	WDT	Active high, level-triggered
17	SDC_1	Active high, level-triggered
18	I2C	Active high, level-triggered
19	AES	Active high, level-triggered
20	-	-
21	UART_0	Active high, level-triggered
22	UART_1	Active high, level-triggered
23	-	-
24	LCDC	Active high, level-triggered
25	UART_2	Active high, level-triggered
26	DDRC	Active high, level-triggered

Bit	Device	Condition
27	-	-
28	-	-
29	H.264 encoder	Active high, level-triggered
30	-	-
31	MPEG4/JPEG codec	Active high, level-triggered
32	VCAP (Video capture)	Active high, level-triggered
33	-	-
34	-	-
35	SAR_ADC	Active high, level-triggered
36	TIMER_1	Active high, level-triggered
37	IC_0	Active high, level-triggered
38	-	-
39	rlc enc	Active high, level-triggered
40	-	-
41	DI (De-interlace)	Active high, level-triggered
42	THDNR (3D de-noise)	Active high, level-triggered
43	ISP	Active high, level-triggered
44	-	-
45	-	-
46	subLVDS	Active high, level-triggered
47	IVS	Active high, level-triggered
48	IR	Active high, level-triggered
49	PWM	Active high, level-triggered
50	RTC INT	Active high, level-triggered
51	RTC ALARM	Active high, level-triggered
52	CSIRX	Active high, level-triggered
53	TIMER_2	Active high, level-triggered
54	SPI	Active high, level-triggered
55	-	-
56	-	-
57	-	-
58	-	-
59	-	-
60	APBBRG_0	Active high, level-triggered

Bit	Device	Condition
61	reg7c[0] (SCU)	Active high, level-triggered
62	reg7c[1] (SCU)	Active high, level-triggered
63	GPIO_1	Active high, level-triggered
64	timer 0 tm1_intr	Active high, level-triggered
65	timer 0 tm2_intr	Active high, level-triggered
66	timer 0 tm3_intr	Active high, level-triggered
67	timer 1 tm1_intr	Active high, level-triggered
68	timer 1 tm2_intr	Active high, level-triggered
69	timer 1 tm3_intr	Active high, level-triggered
70	timer 2 tm1_intr	Active high, level-triggered
71	timer 2 tm2_intr	Active high, level-triggered
72	timer 2 tm3_intr	Active high, level-triggered
73	-	-
74	-	-
75	-	-
76	set rega8[0]/clr rega8[16] (scu)	Active high, level-triggered
77	set rega8[1]/clr rega8[17] (scu)	Active high, level-triggered
78	set rega8[2]/clr rega8[18] (scu)	Active high, level-triggered
79	set rega8[3]/clr rega8[19] (scu)	Active high, level-triggered
80	set rega8[4]/clr rega8[20] (scu)	Active high, level-triggered
81	set rega8[5]/clr rega8[21] (scu)	Active high, level-triggered
82	set rega8[6]/clr rega8[22] (scu)	Active high, level-triggered
83	set rega8[7]/clr rega8[23] (scu)	Active high, level-triggered
84	set rega8[8]/clr rega8[24] (scu)	Active high, level-triggered
85	set rega8[9]/clr rega8[25] (scu)	Active high, level-triggered
86	set rega8[10]/clr rega8[26] (scu)	Active high, level-triggered
87	set rega8[11]/clr rega8[27] (scu)	Active high, level-triggered
88	set rega8[12]/clr rega8[28] (scu)	Active high, level-triggered
89	set rega8[13]/clr rega8[29] (scu)	Active high, level-triggered
90	set rega8[14]/clr rega8[30] (scu)	Active high, level-triggered
91	set rega8[15]/clr rega8[31] (scu)	Active high, level-triggered
92	-	-

4.2.3 Programming Model

The following abbreviations are used for the register definitions:

- R/W: Read/Write
- WO: Write Only
- RO: Read Only

4.2.3.1 Register Summary of Interrupt Controller

The register summary of the interrupt controller is listed in Table 4-66.

Table 4-66. Register Summary of Interrupt Controller

Address	R/W Type	Description	Reset Value
0x00	RO	SPI[31:0] source register	0x0000_0000
0x04	R/W	SPI[31:0] enable register	0x0000_0000
0x08	WO	SPI[31:0] interrupt clear register	-
0x0C	R/W	SPI[31:0] trigger mode register	0x0000_0000
0x10	R/W	SPI[31:0] trigger level register	0x0000_0000
0x14	RO	SPI[31:0] pending register	0x0000_0000
0x20	RO	SPI[63:32] source register	0x0000_0000
0x24	R/W	SPI[63:32] enable register	0x0000_0000
0x28	WO	SPI[63:32] interrupt clear register	-
0x2C	R/W	SPI[63:32] trigger mode register	0x0000_0000
0x30	R/W	SPI[63:32] trigger level register	0x0000_0000
0x34	RO	SPI[63:32] pending register	0x0000_0000
0x58	R/W	SPI[31:0] de-bounce register	It depends on the configuration.
0x5C	R/W	SPI[63:32] de-bounce register	It depends on the configuration.
0x60	RO	SPI[92:64] source register	0x0000_0000
0x64	R/W	SPI[92:64] enable register	0x0000_0000
0x68	WO	SPI[92:64] interrupt clear register	-
0x6C	R/W	SPI[92:64] trigger mode register	0x0000_0000
0x70	R/W	SPI[92:64] trigger level register	0x0000_0000
0x74	RO	SPI[92:64] pending register	0x0000_0000
0x78	R/W	SPI[92:64] de-bounce register	It depends on the configuration.

Address	R/W Type	Description	Reset Value
0x600	RO	SPI[31:0] active bit register	0x0000_0000
0x604	RO	SPI[63:32] active bit register	0x0000_0000
0x608	RO	SPI[95:64] active bit register	0x0000_0000
0x660 ~ 66C	RW	SGI[15:0] interrupt priority register	0x0000_0000
0x680	WC	SGI control register	-
0x684	RW	Priority mask register	0xF
0x688	RW	Binary point register	0x7
0x68C	RO	Interrupt acknowledge register	0x1FF
0x690	WO	End of interrupt register	-
0x694	RO	Run priority register	0xF
0x698	RO	Highest pending interrupt register	0x1FF
0x69C	RW	Interrupt control register	0x0000_0006
0x6A0	RW	CPU0 match register	It depends on the configuration.
0x6A4	RW	CPU1 match register	It depends on the configuration.
0x6D0	R/W	CPU0 match mask register	0x0
0x6D4	R/W	CPU1 match mask register	0x0
0x700	RO	Feature1 register	It depends on the configuration.
0x708	RO	Revision register	-

4.2.3.2 Register Definitions

The following subsections provide the detailed descriptions of the control registers.

4.2.3.2.1 SPI Source Register

Table 4-67. SPI Source Register

Bit	Name	Type	Description
[31:0]	spi_src_reg	RO	This register records the status of the SPI[477:0] interrupt detection on the distributor interface

The base address offset of the SPI source register is shown in Figure 4-5. The interrupt controller supports 93 SPIs and it will accordingly reduce the number of the registers.

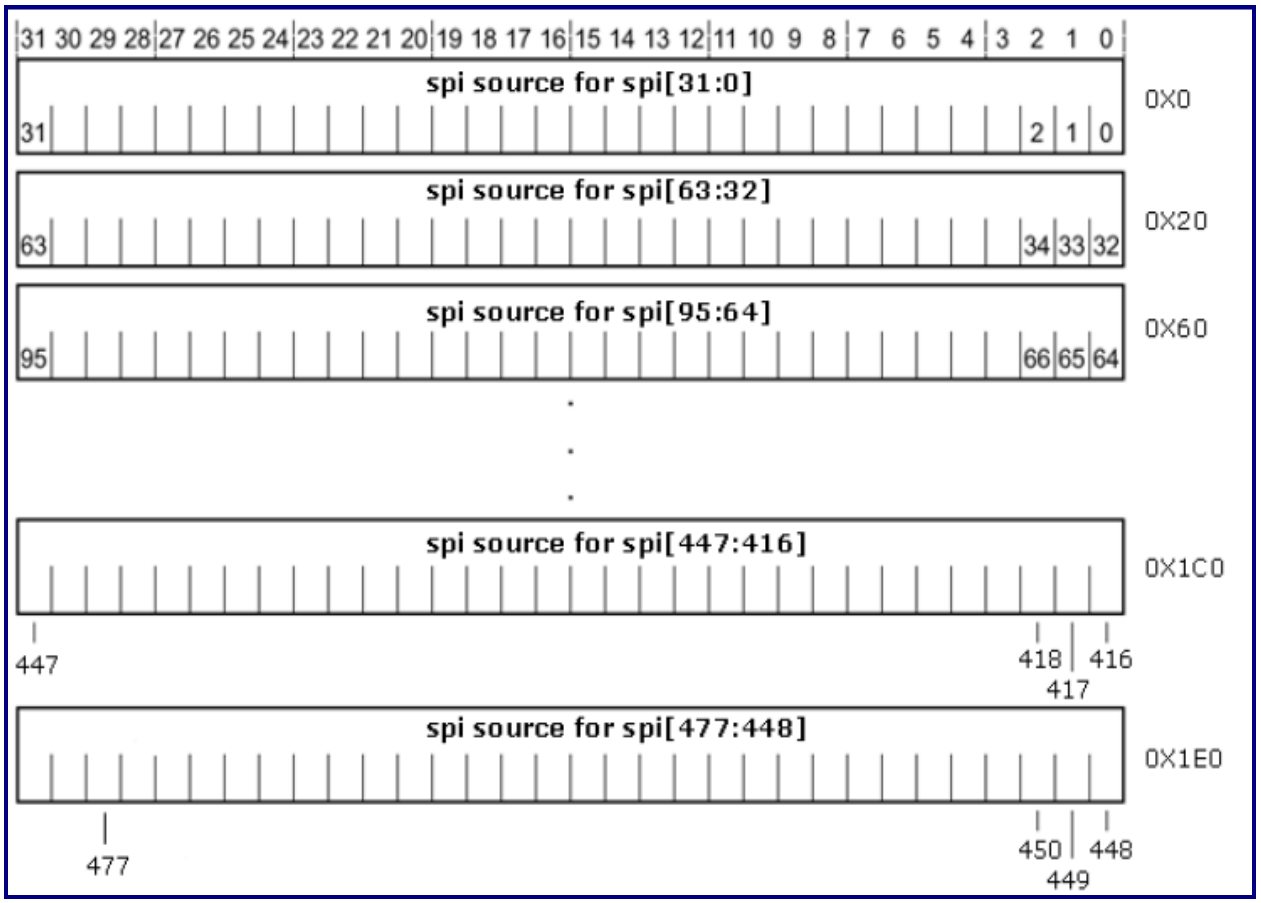


Figure 4-5. Base Address Offset of SPI Source Register

4.2.3.2.2 SPI Enable Register

Table 4-68. SPI Enable Register

Bit	Name	Type	Description
[31:0]	spi_en_reg	RW	Enable/Disable the interrupt source of SPI 0: Disable the interrupt source 1: Enable the interrupt source

The base address offset of the SPI enable register is shown in Figure 4-6. The interrupt controller supports 93 SPIs and it will accordingly reduce the number of the registers.

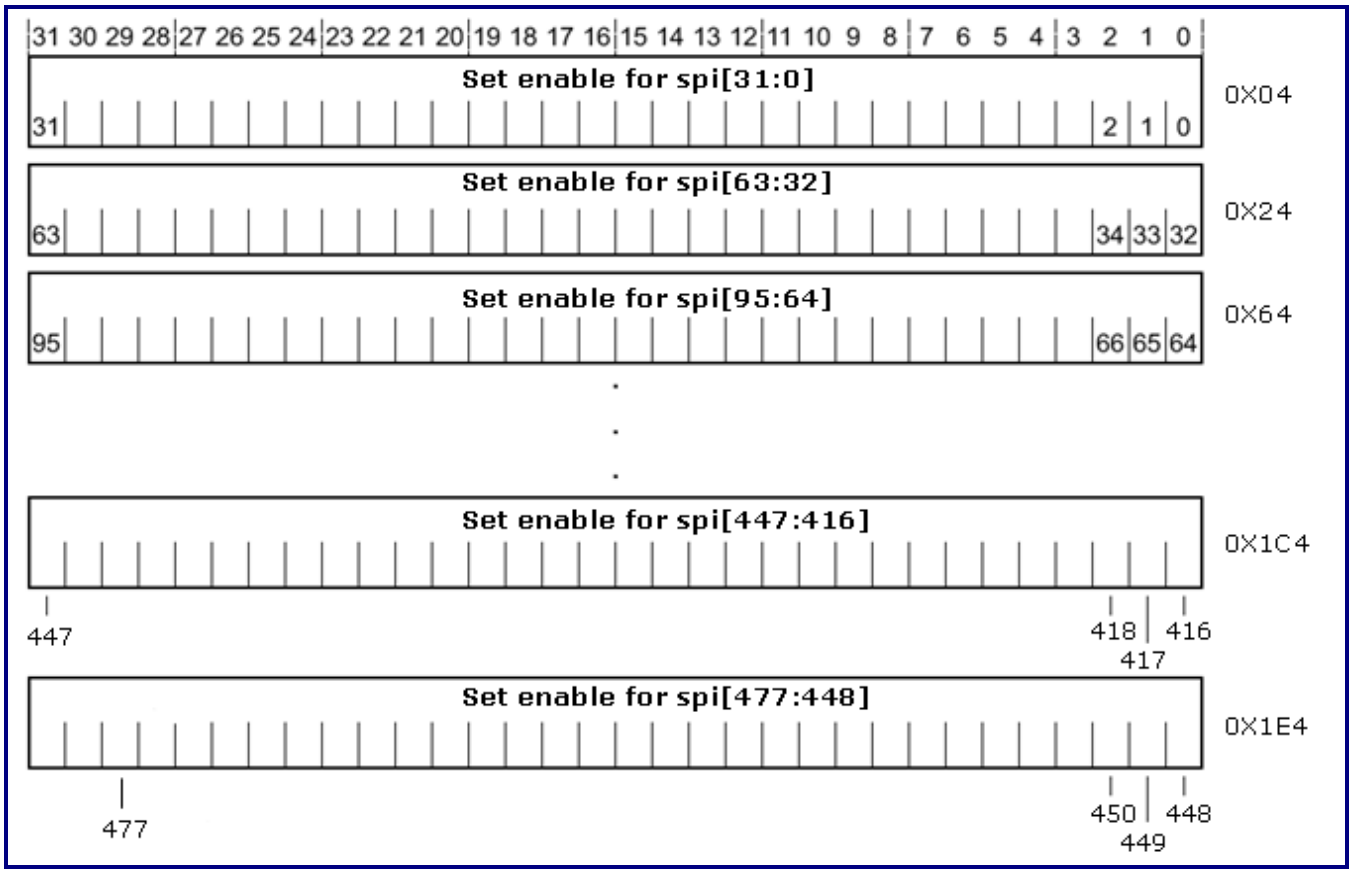


Figure 4-6. Base Address Offset of SPI Enable Register

4.2.3.2.3 SPI Interrupt Clear Register

Table 4-69. SPI Interrupt Clear Register

Bit	Name	Type	Description
[31:0]	spi_clr_reg	WO	When set to '1', the interrupt status, spi_src_reg, will automatically be cleared to '0'. Note: This register can only take effect when the input interrupt sources are in the edge-trigger mode.

The base address offset of the SPI interrupt clear register is shown in Figure 4-7. The interrupt controller supports 93 SPIs and it will accordingly reduce the number of the registers.

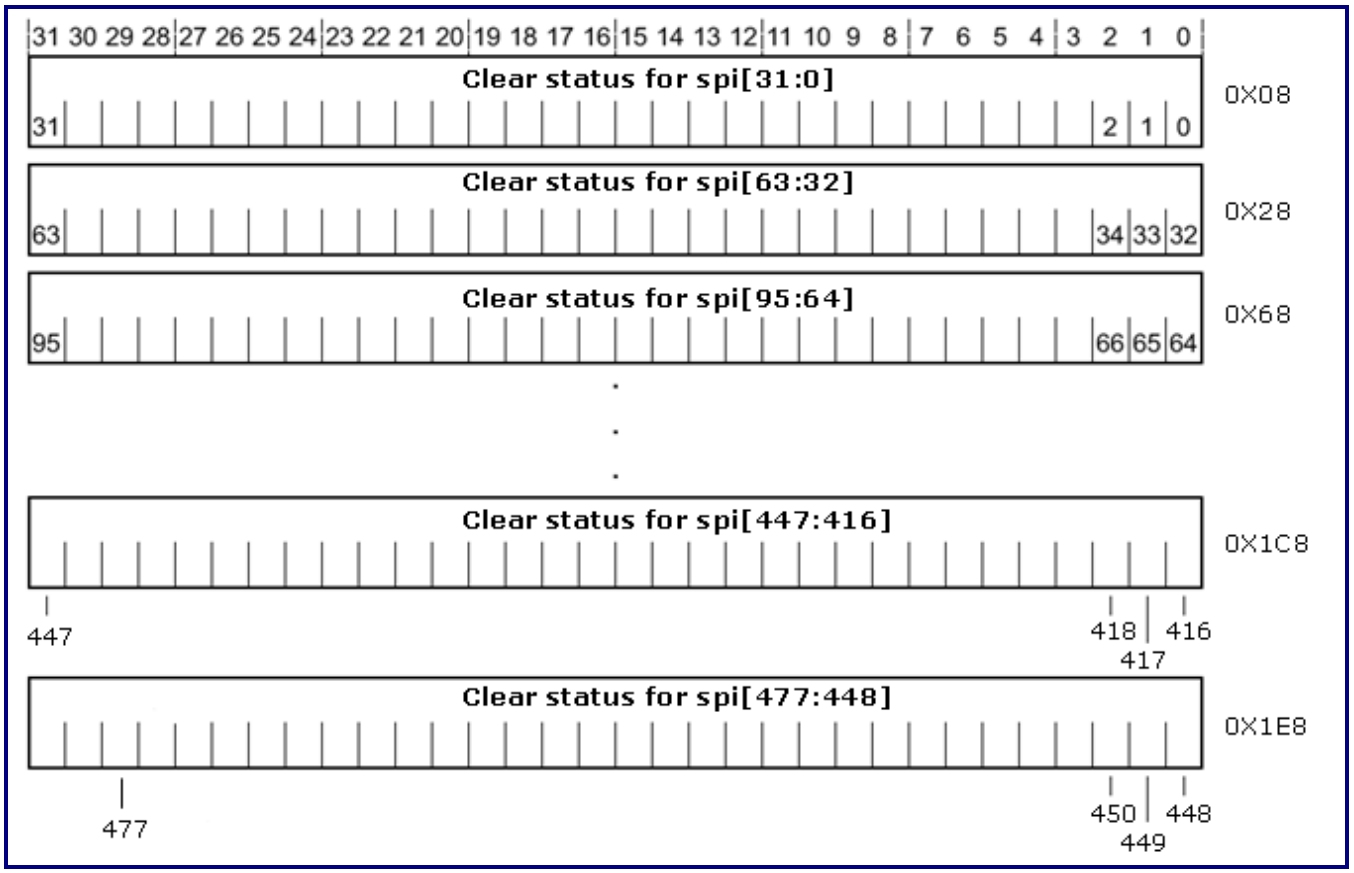


Figure 4-7. Base Address Offset of SPI Interrupt Clear Register

4.2.3.2.4 SPI Trigger Mode Register

Table 4-70. SPI Trigger Mode Register

Bit	Name	Type	Description
[31:0]	spi_mode_reg	RW	This register selects the type of the interrupt source of SPI. 0: Level-trigger mode 1: Edge-trigger mode

The base address offset of the SPI-trigger mode register is shown in Figure 4-8. The interrupt controller supports 93 SPIs and it will accordingly reduce the number of the registers.

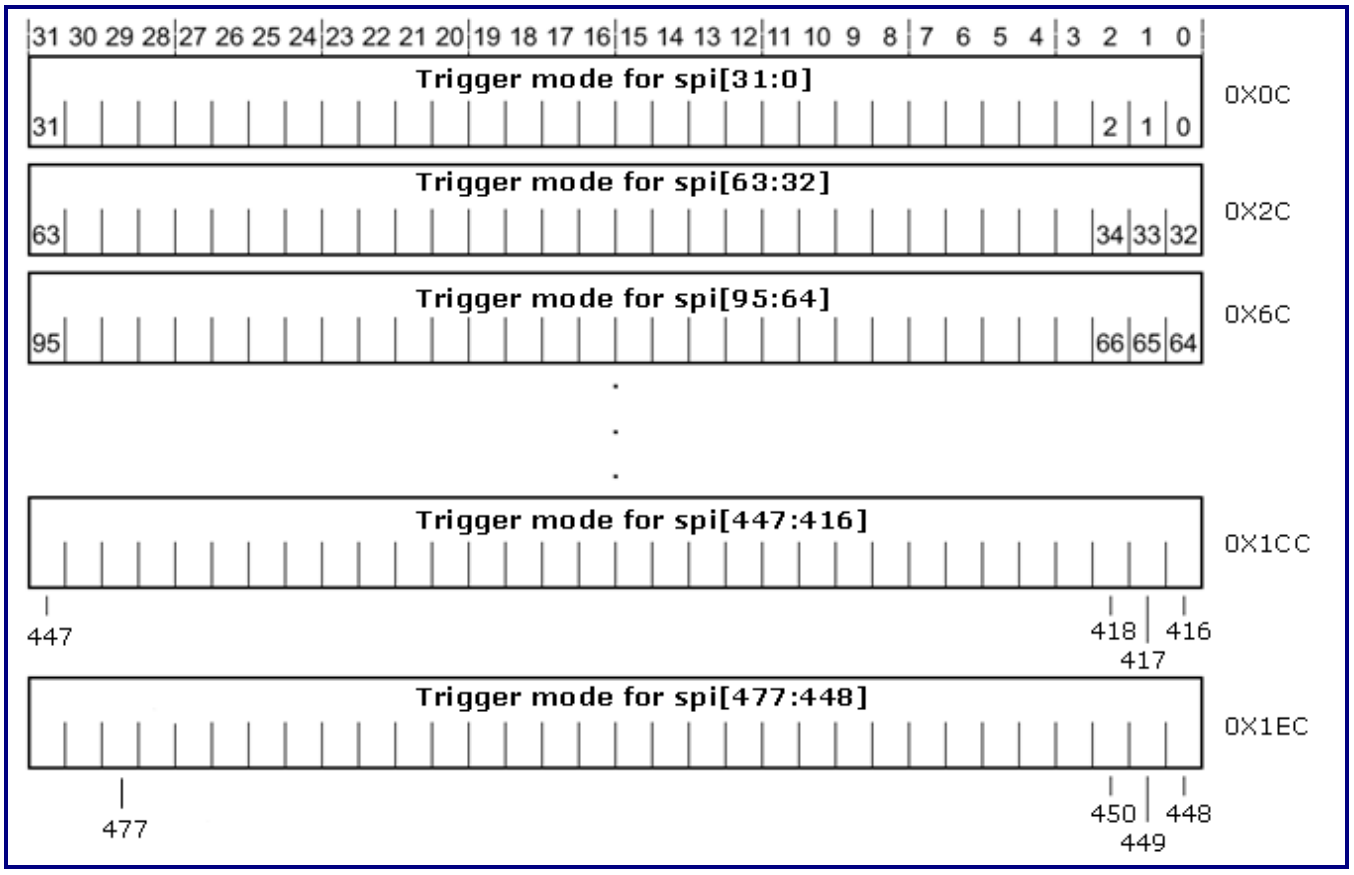


Figure 4-8. Base Address Offset of SPI-trigger Mode Register

4.2.3.2.5 SPI Trigger Level Register

Table 4-71. SPI Trigger Level Register

Bit	Name	Type	Description
[31:0]	spi_level_reg	RW	This register selects the trigger level of the SPI interrupt source. 0: Active-high level-trigger or rising-edge trigger 1: Active-low level-trigger or falling-edge trigger

The base address offset of the SPI trigger level register is shown in Figure 4-9. The interrupt controller supports 93 SPIs and it will accordingly reduce the number of the registers.

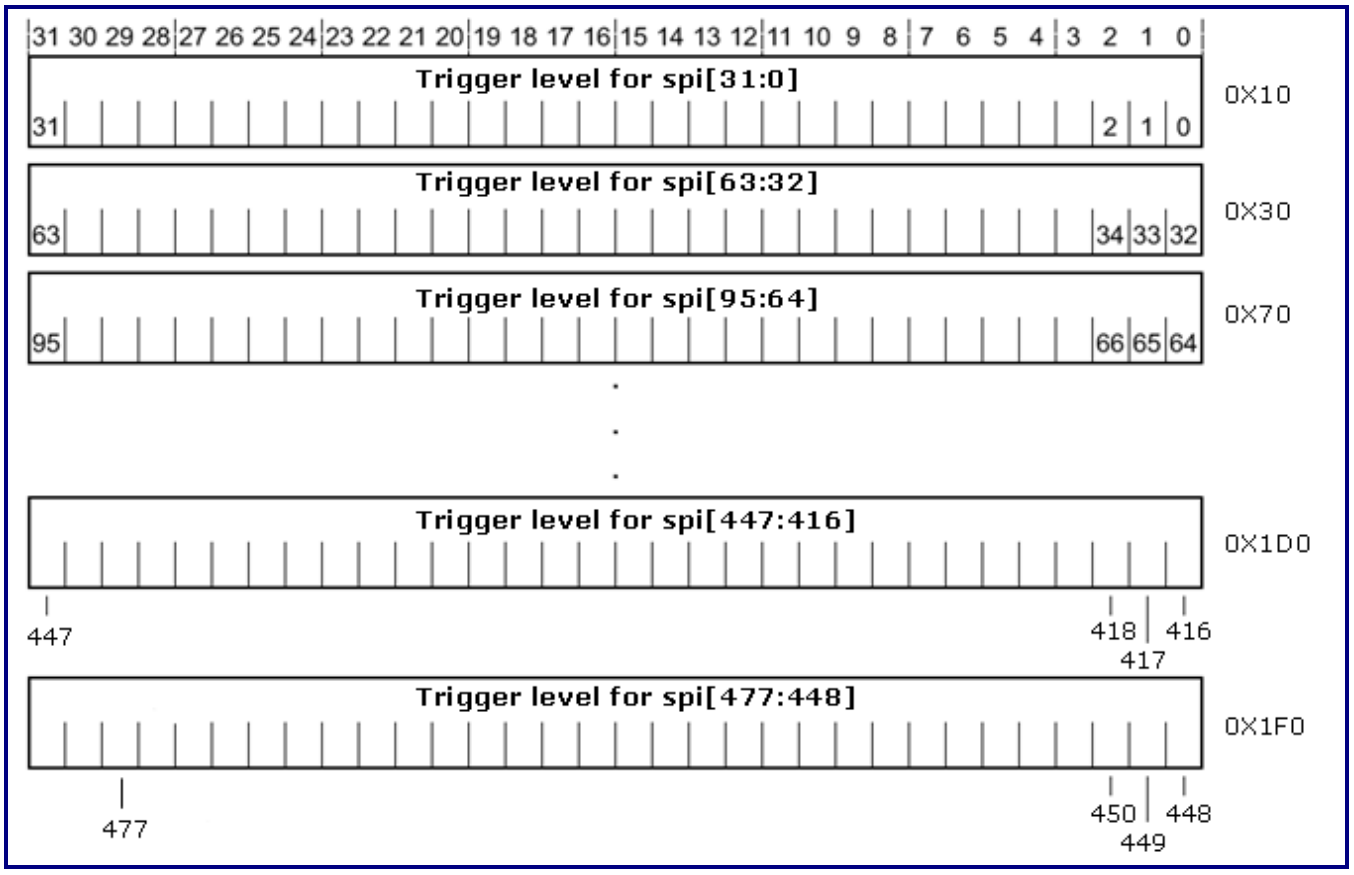


Figure 4-9. Base Address Offset of SPI Trigger Level Register

4.2.3.2.6 SPI Pending Register

Table 4-72. SPI Pending Register

Bit	Name	Type	Description
[31:0]	spi_pending_reg	RO	This register records the SPI status after masking spi_en_reg. 0: Interrupt is not asserted. 1: Interrupt is asserted.

The base address offset of the SPI pending register is shown in Figure 4-10. The interrupt controller supports 93 SPIs and it will accordingly reduce the number of the registers.

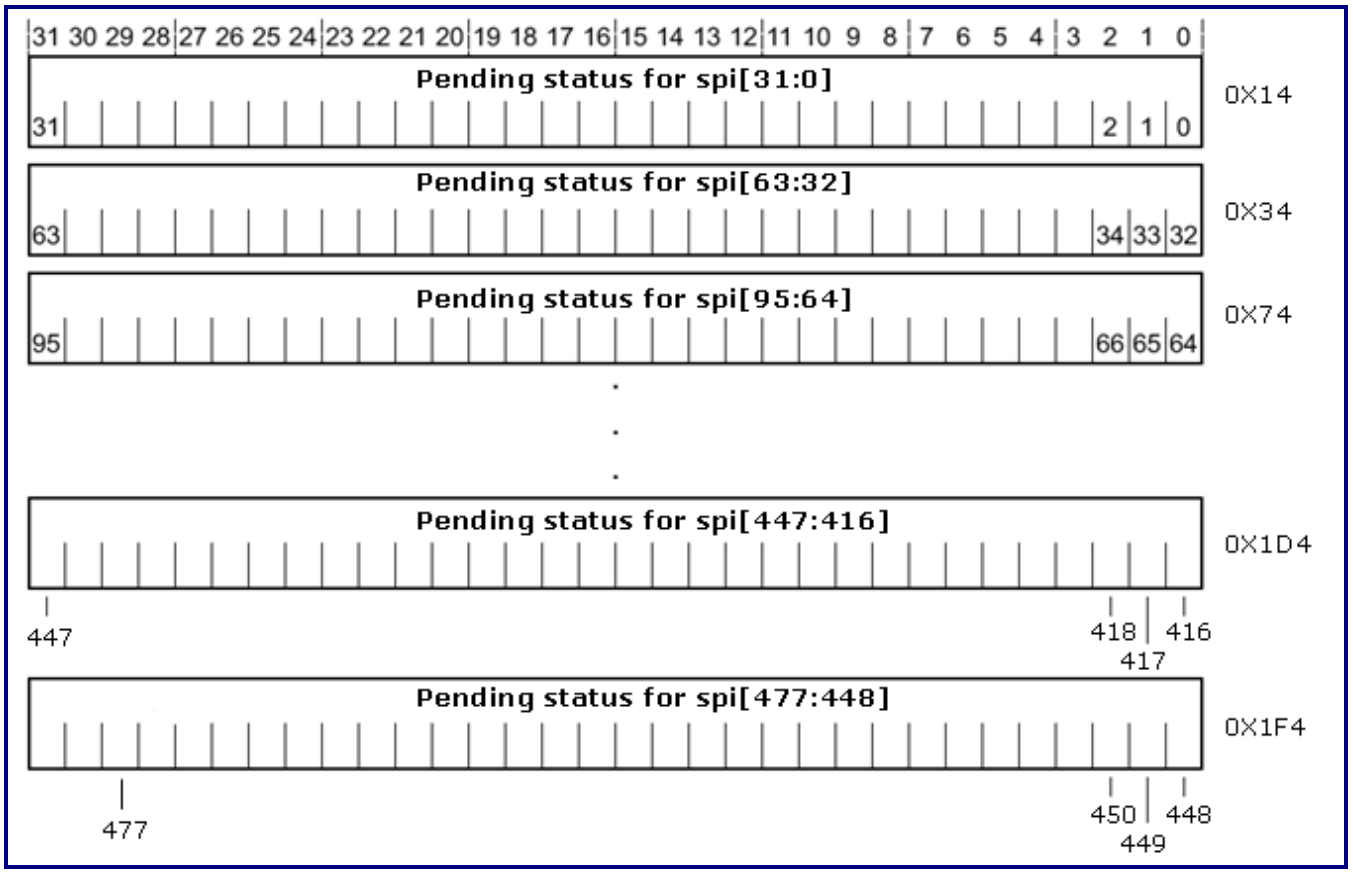


Figure 4-10. Base Address Offset of SPI Pending Register

4.2.3.2.7 SPI De-bounce Register

Table 4-73. SPI De-bounce Register

Bit	Name	Type	Description
[31:0]	spi_debounce_reg	RW	The interrupt controller supports one-cycle de-bounce design for the interrupt input to prevent glitches. This register is used to control the SPI interrupt de-bounce function. 0: Disable 1: Enable

The base address offset of the SPI de-bounce register is shown in Figure 4-11. The interrupt controller supports 93 SPIs and it will accordingly reduce the number of the registers.

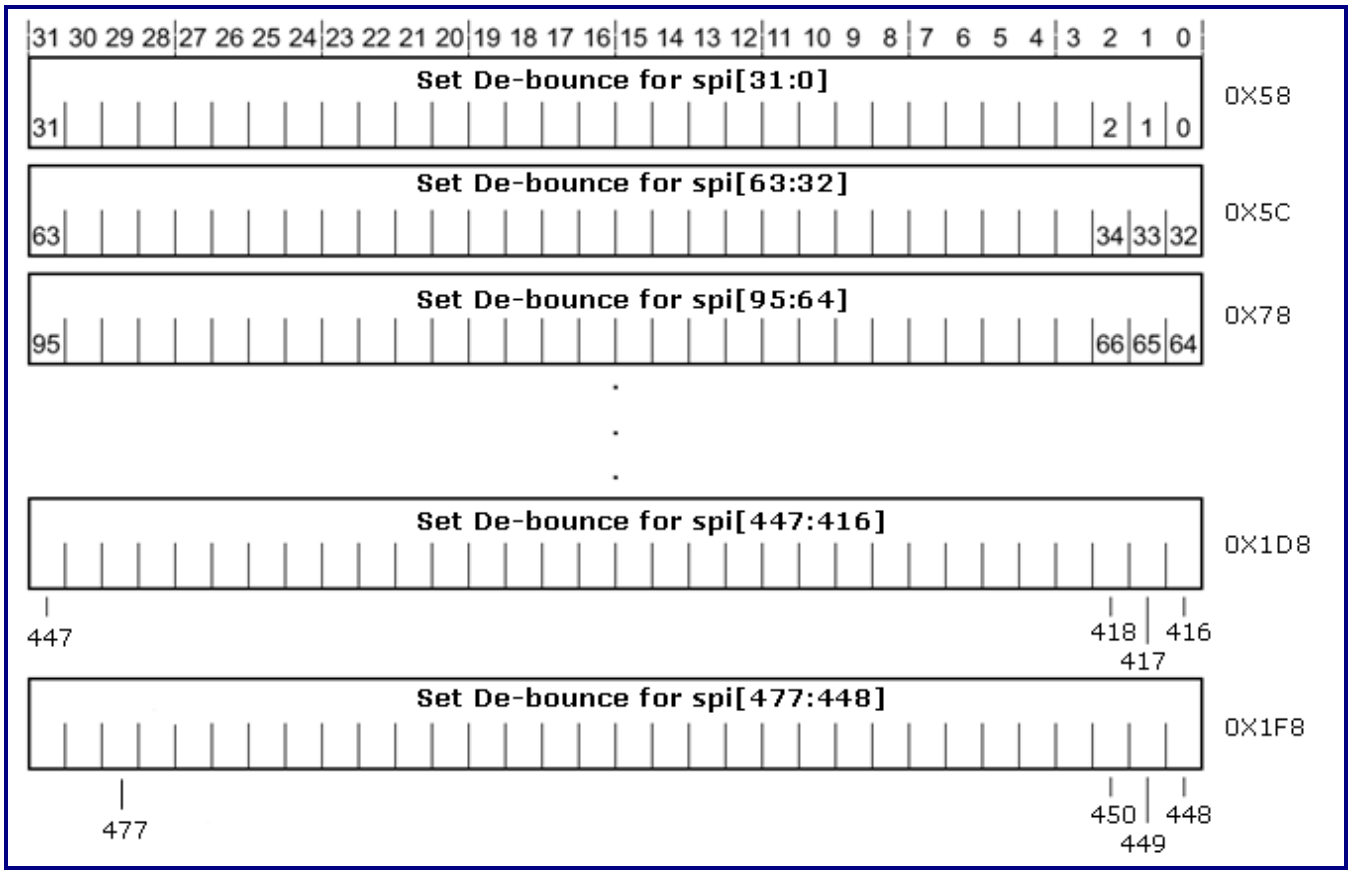


Figure 4-11. Base Address Offset of SPI De-bounce Register

4.2.3.2.8 SPI Active Bit Register

Table 4-74. SPI Active Bit Register

Bit	Name	Type	Description
[31:0]	spi_active_reg	RO	The active bit register is used to determine which interrupt is currently active (Bit read as '1') on at least one CPU.

The base address offset of the SPI active bit register is shown in Figure 4-12. The interrupt controller supports 93 SPIs and it will accordingly reduce the number of the registers.

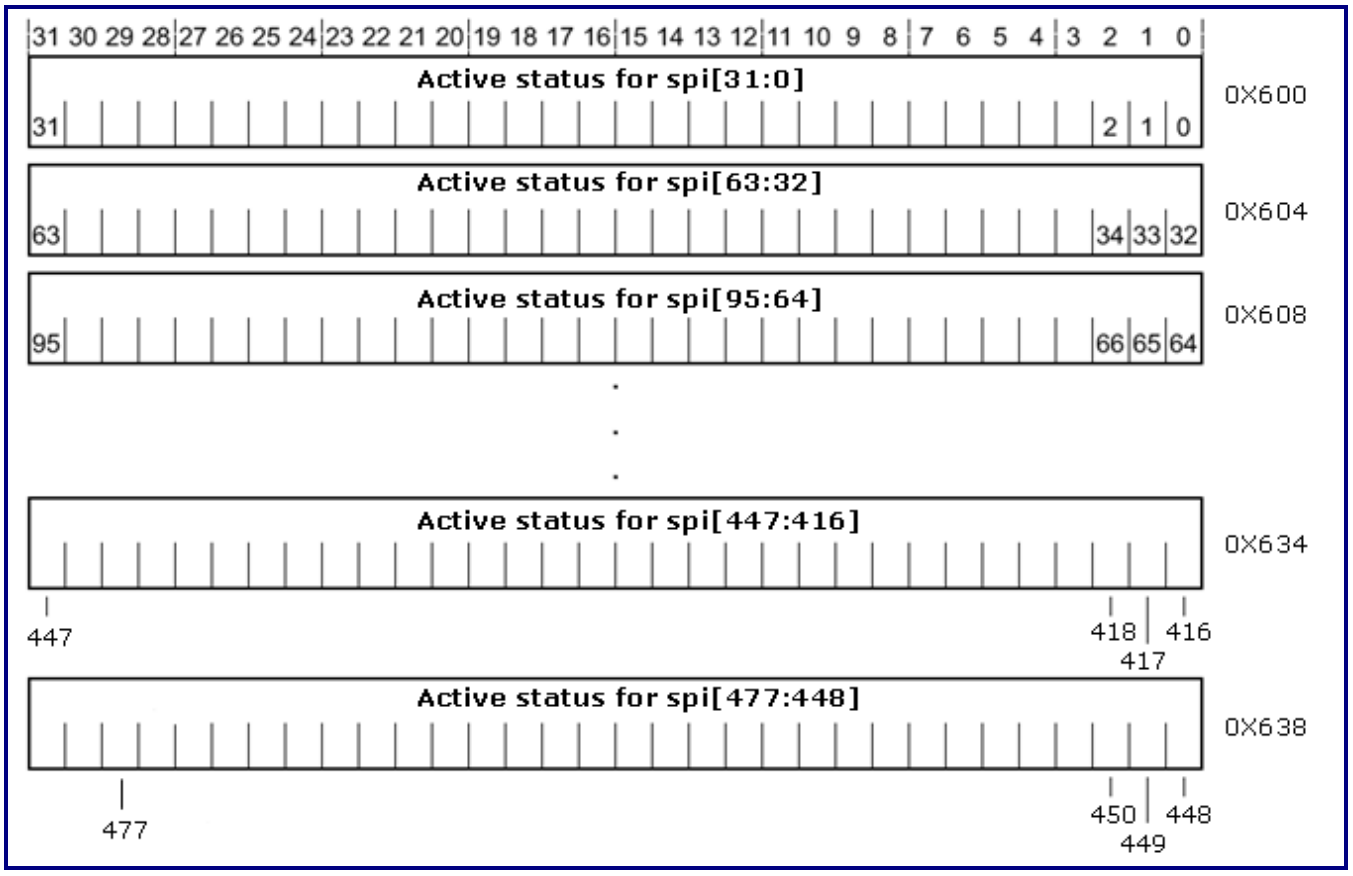


Figure 4-12. Base Address Offset of SPI Active Register

4.2.3.2.9 SPI Interrupt Priority Register

Table 4-75. SPI Interrupt Priority Register

Bit	Name	Type	Description
[31:0]	spi_priority_reg	RW	This register stores the priority level of individual interrupt. The highest priority is '0', and the lowest priority is 0xF.

The base address offset of the SPI interrupt priority register is shown in Figure 4-13. This register holds the 4bit priority level for four interrupts. Bits[31:28], bits[23:20], bits[15:12], and bits[7:4] are reserved and read as zero. The interrupt controller supports 93 SPIs and it will accordingly reduce the number of the registers.

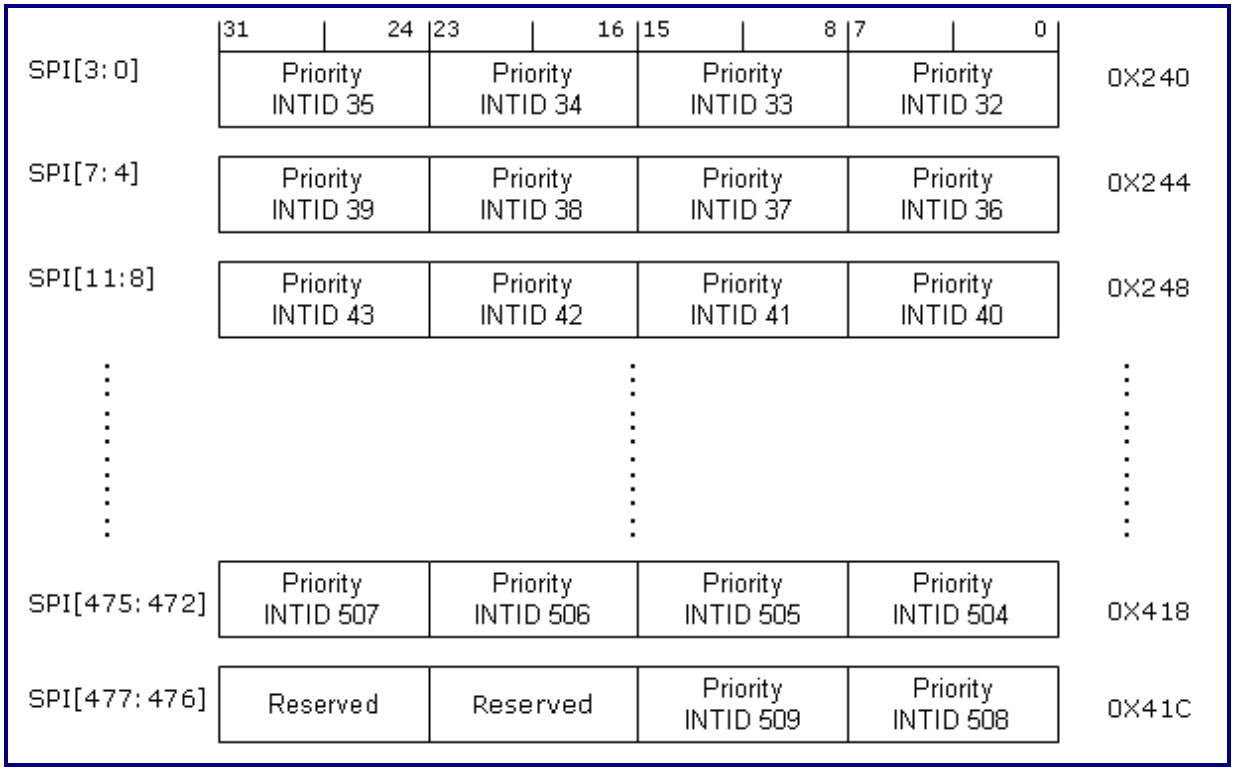


Figure 4-13. Base Address Offset of SPI Interrupt Priority Register

4.2.3.2.10 SPI Target Register

The content of the SPI target register will be rearranged in the CPUID order, as shown in Table 4-76.

Table 4-76. SPI Target Register

Bit	Name	Type	Description
[31:0]	spi_target_reg	RW	This register determines which CPU can receive an interrupt. The interrupt controller supports 93 SPIs. Each bit holds 32bit target for 32 interrupts to one CPU. Each bit in the target registers refers to one SPI. For example, if the value of the register offset 0x420 equals to 0x03, it indicates that the interrupts 32 and 33 will be sent to CPU0. The interrupt target register can be ignored for the software-trigged interrupts.

The base address offset of the SPI target register is shown in Figure 4-14. The interrupt controller only supports 93 SPIs. It will accordingly reduce the number of the registers.

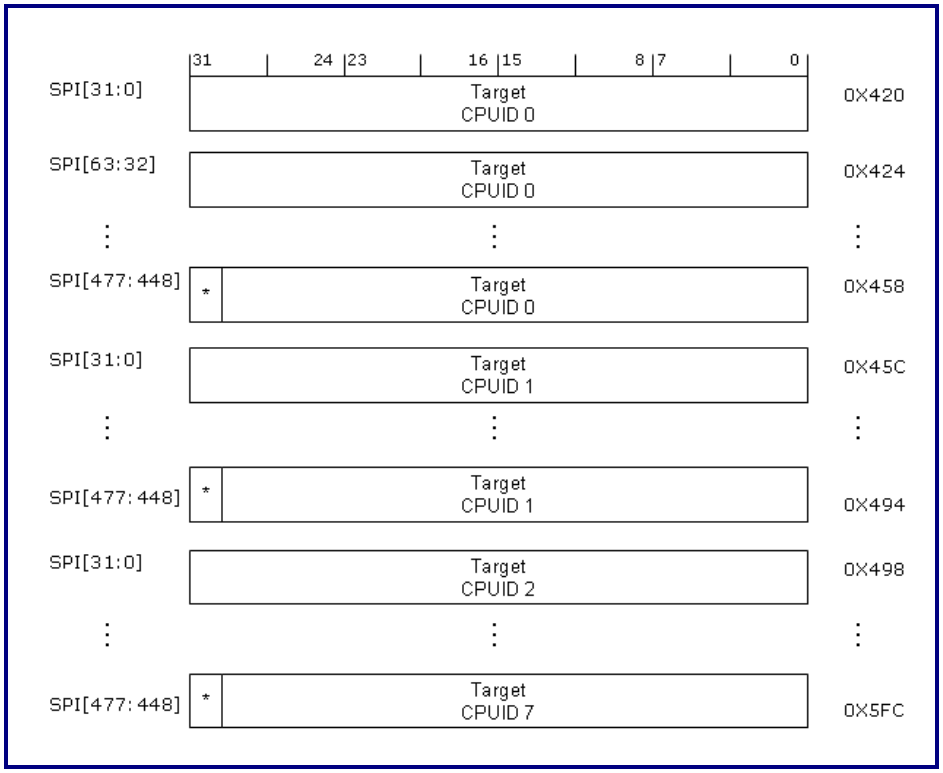


Figure 4-14. Base Address Offset of SPI Target Register

4.2.3.2.11 SGI Interrupt Priority Register

Table 4-77. SGI Interrupt Priority Register

Bit	Name	Type	Description
[31:0]	cpun_sgi_priority_reg	RW	This register stores the priority of individual interrupt. The highest priority is '0', and the lowest priority is 0xF. Where "n" indicates the number of the CPU interfaces. "n" is 7.

The base address offset of the SGI interrupt priority register is shown in Figure 4-15. The SGI interrupt ID is from 0 to 15. The interrupt controller has seven CPU interfaces, the distributor will provide banked registers at the address offset from 0x660 to 0x66C. The distributor uses the ID tag, received on ARId or AWId by using the settings of cpu***n***_match_reg, to select the appropriate register. The interrupt controller supports less than 16 SGIs and it will accordingly reduce the number of the registers.

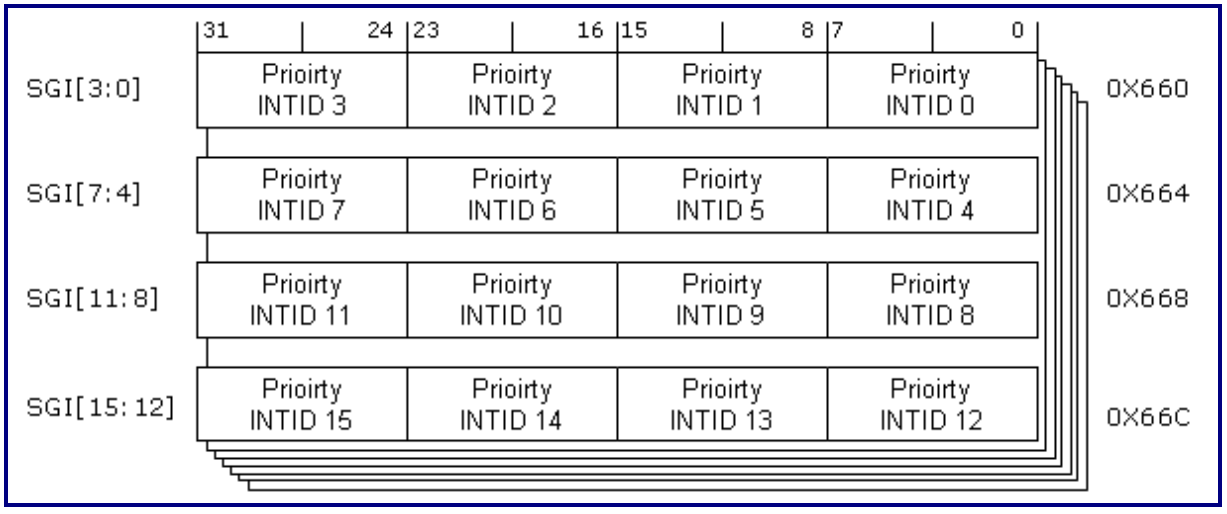


Figure 4-15. Base Address Offset of SGI Interrupt Priority Register

4.2.3.2.12 SGI Control Register

Table 4-78. SGI Control Register

Bit	Name	Type	Description
[31:17]	-	-	Reserved
[16:9]	target_list	WC	This field determines which CPU can receive an interrupt. 0: Interrupt will not trigger the relative CPU. 1: Interrupt will trigger the relative CPU. Bit[9] is mapped to CPU0; bit[16] is mapped to CPU7, and so on.
[8:0]	INTID	WC	This field is used to trigger an interrupt (Identified with the related INTID) to a list of CPUs. If INTID is bigger than the number of the supported interrupts, it has no effect.

When users trigger an interrupt ID ≥ 32 (SPI), target_list will overwrite the setting in **spi_target_reg**.

4.2.3.2.13 CPU Match Register

Table 4-79. CPU Match Register

Bit	Name	Type	Description
[31:N+1]	-	-	Reserved
[N:0]	cpun_match_reg	RW	This register determines the CPUn ID tag for accessing the CPUn bank register, where “n” indicates the number of the CPU interfaces. The valid value is ranging from 0 to 7.

"N" is configured by using the AXI command ID width. The base address offset of the CPU match register is shown in Figure 4-16. The interrupt controller supports seven CPU interfaces and it will accordingly reduce the number of the registers.

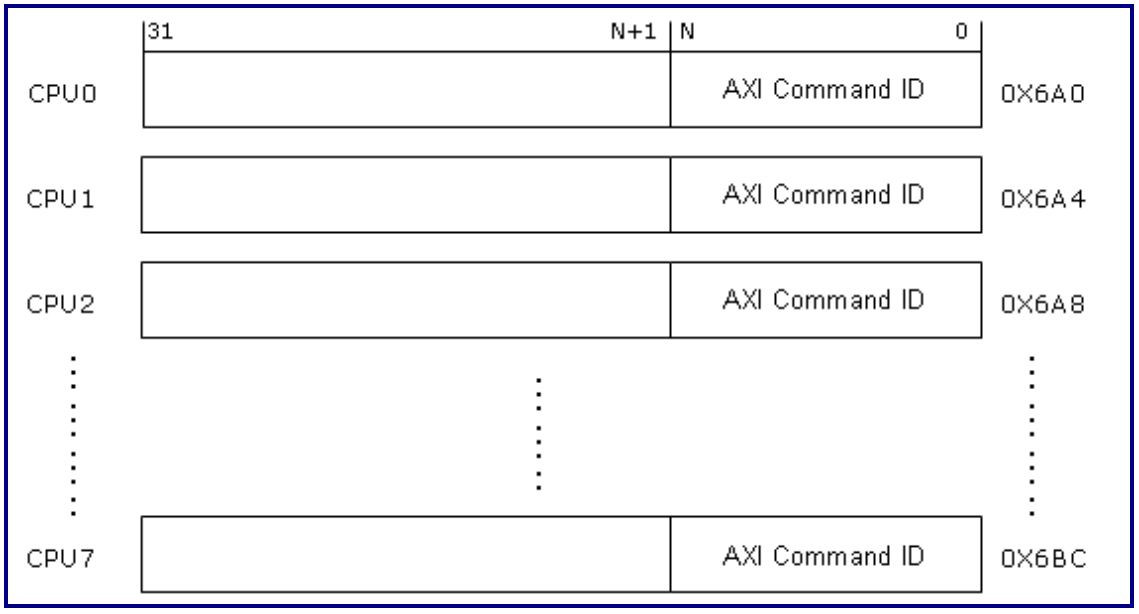


Figure 4-16. Base Address Offset of CPU Match Register

4.2.3.2.14 CPU Match Mask Register

Table 4-80. CPU Match Mask Register

Bit	Name	Type	Description
[31:N + 1]	-	-	Reserved
[N:0]	cpun_match_mask_reg	RW	This register compares the mask bit to the CPUID tag. Where "N" indicates the number of the CPU interfaces. The valid value is ranging from 0 to 7. If users want to mask the comparison, please set the related bit to '1'.

"N" is configured by using the AXI ID width command. The base address offset of the CPU match mask register is shown in Figure 4-17. The interrupt controller supports seven CPU interfaces and it will accordingly reduce the number of the registers.

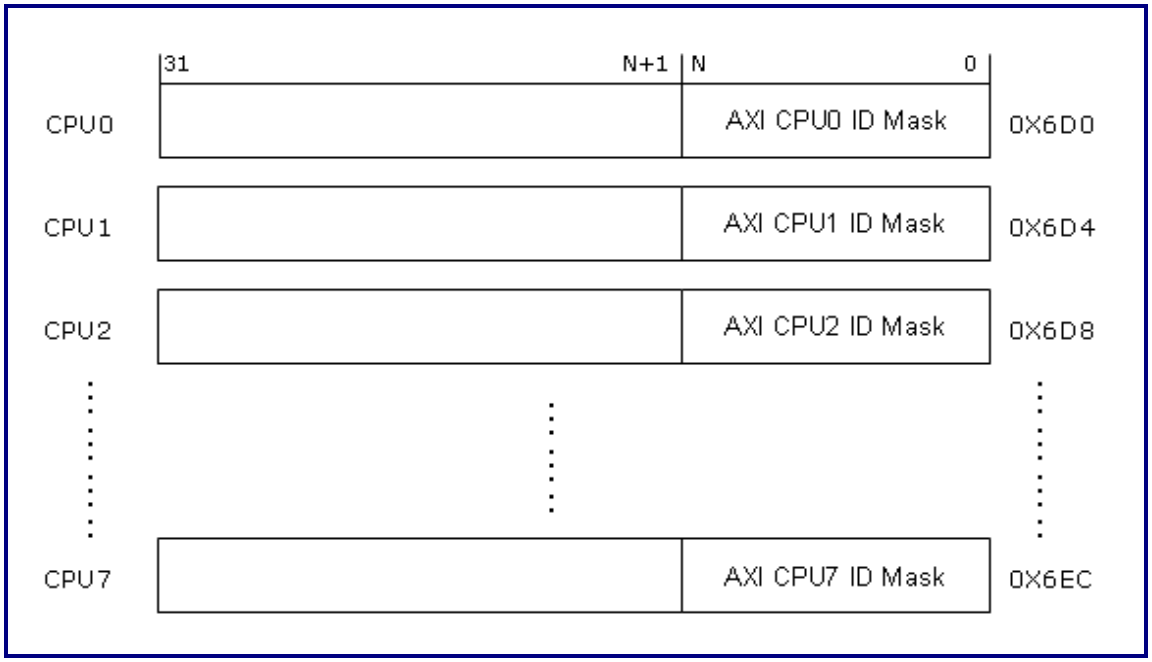


Figure 4-17. Base Address Offset of CPU Match Mask Register

4.2.3.2.15 Feature1 Register

Table 4-81 lists and describes the current features of the interrupt controller. The hardware configuration features are shown in this register.

Table 4-81. Feature1 Register

Bit	Name	Type	Description
[31:20]	-	-	Reserved
[19:16]	CPU_NUM	RO	Number of CPUs
[15:11]	SGI_NUM	RO	Number of SGIs
[10:2]	SPI_NUM	RO	Number of SPIs

Bit	Name	Type	Description
1	LGI	RO	Legacy interrupt detection function control 0: Disable 1: Enable
0	BW_EN	RO	Backward compatible function control 0: Disable 1: Enable

4.2.3.2.16 Revision Register

Table 4-82 lists and describes the current revision number of the interrupt controller.

Table 4-82. Revision Register

Bit	Name	Type	Description
[31:24]	-	-	Reserved
[23:16]	MAJOR_REV	RO	Major revision number
[15:8]	MINOR_REV	RO	Minor revision number
[7:0]	REL_REV	RO	Release number

4.2.4 Functional Description

4.2.4.1 AXI Slave Port

The interrupt controller contains one AXI slave port for accessing the register. Because it does not have the command FIFO, it can only deal with the commands one by one. If the read and write commands arrive at the same time, the read command will be served first if the prior command is also a read command.

4.2.4.2 Interrupt Detection

According to the register setting, the interrupt controller supports four interrupt detection types for SGI and SPI: High-level, low-level, rising-edge, and falling-edge. In order to prevent the signal noise, the interrupt controller supports the one-cycle de-bounce function for each interrupt.

4.2.4.3 Priority Arbitration

Each interrupt (SGI/SPI) has its priority level. Address 0x0 has the highest priority while address 0xF has the lowest priority. The interrupt controller will find the interrupt with the highest priority from all asserted interrupts. The priority arbitration is based on the following rules:

- Interrupt is asserted.
- Interrupt is chosen with the highest priority level.
- If the priority level is the same, choose the interrupt with the lowest interrupt ID.

The interrupt controller supports 16 SGIs and 93 SPIs. At the first arbitration cycle, the controller finds the highest interrupt priority among every 16 interrupts. At the second arbitration cycle, the controller continuously finds the highest interrupt priority from the results of the first arbitration cycle. Therefore, users can obtain the highest priority level and the interrupt ID. Figure 4-18 illustrates an example of the priority arbitration between SPI[0] and SPI[1] .

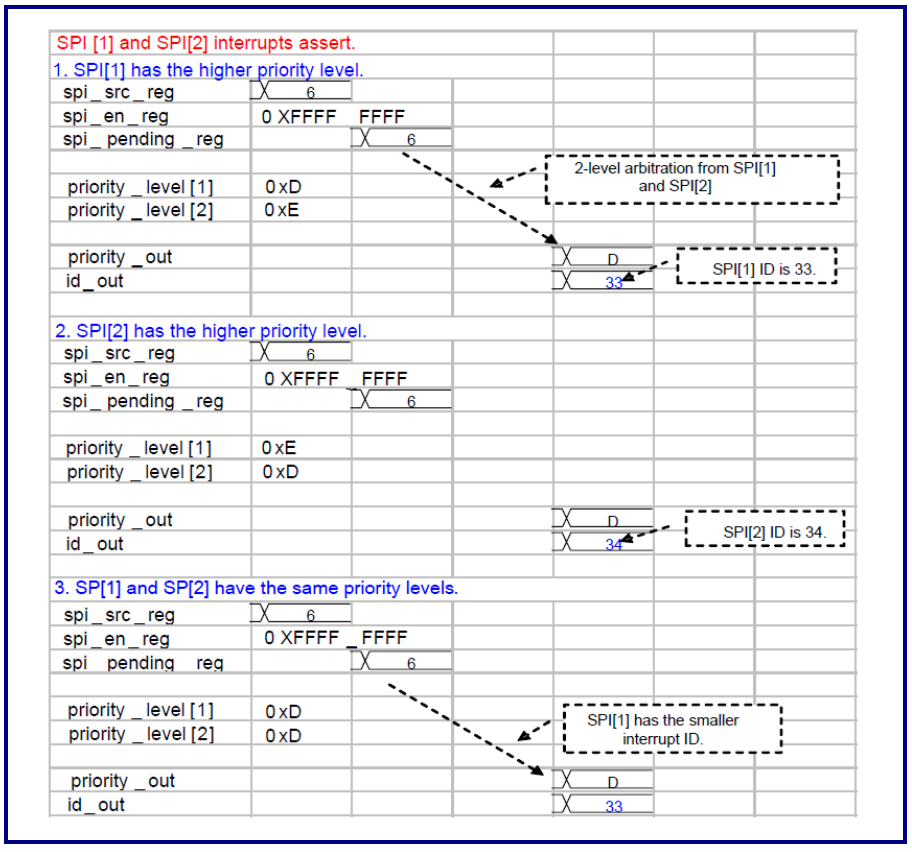


Figure 4-18. Interrupt Arbitration between SPI[0] and SPI[1]

4.2.4.4 Stack Array

The stack array is used to record the interrupt ID and run the priority level for the interrupt pre-emption. For example, if CPU currently serves an interrupt with the interrupt priority of 0xE, another interrupt with the priority level of 0xC will be asserted before completing this interrupt. Because the new coming interrupt has the higher priority level, 0xC, cpu***n***_irq_out will be asserted again and the interrupt controller will need to push the currently running priority level, 0xE, and save the interrupt ID into the stack array. When CPU finishes the 0xC interrupt service and writes 0xC into the EOI register, the running register and ACK register will be updated by popping from the stack array.

4.2.5 Initialization

4.2.5.1 Backward Compatible Scheme

Users need to follow the procedure listed below for clearing the corresponding interrupt status:

1. Set "Trigger Mode Register" and "Trigger Level Register" for SPI to ensure that the input trigger type is correct.
2. Set "SPI De-bounce Register" to enable the de-bounce function
3. Set "SPI Enable Register" to enable the corresponding bits
4. When cpu***n***_irq_out_r is active, read "SPI Pending Register" to decide which interrupt to be served.
5. After serving the interrupt, clear the requested interrupt in the peripheral (In the level-trigger mode) or set "SPI Interrupt Clear Register" (In the edge-trigger mode) to clear the corresponding interrupt status.

4.3 Timer

4.3.1 General Description

The timer provides three independent sets of sub-timers. Each sub-timer can use the internal system clock (PCLK) or external clock (EXTCLK) to increase or decrease the counting. Two match registers are provided for each sub-timer. When the value of the match registers equals to any value of the sub-timers, the timer interrupt will be immediately triggered. Issuing the timer interrupt will be decided by the register setting when an overflow occurs.

4.3.2 Features

- Includes three independent 32bit timer programming models
- Supports internal or external clock source selection
- Issued interrupt upon overflow and time-up
- Contains two match registers in each timer
- Supports incremental and decremental models

4.3.3 Programming Model

4.3.3.1 Summary of Timer Registers

Table 4-83 shows the offset, type, width, reset value, name, and configuration option of each programming timer register.

Table 4-83. Summary of Timer Registers

Offset	Type	Width	Reset	Name	Configuration	Description
0x00	R/W	32	0x—	Tm1Counter	None	Timer1 counter
0x04	R/W	32	0x—	Tm1Load	None	Timer1 auto reload value
0x08	R/W	32	0x—	Tm1Match1	None	Timer1 match value
0x0C	R/W	32	0x—	Tm1Match2	None	Timer1 match value
0x10	R/W	32	0x—	Tm2Counter	TM2	Timer2 counter
0x14	R/W	32	0x—	Tm2Load	TM2	Timer2 auto reload value
0x18	R/W	32	0x—	Tm2Match1	TM2	Timer2 match value
0x1C	R/W	32	0x—	Tm2Match2	TM2	Timer2 match value
0x20	R/W	32	0x—	Tm3Counter	TM3	Timer3 counter
0x24	R/W	32	0x—	Tm3Load	TM3	Timer3 auto reload value
0x28	R/W	32	0x—	Tm3Match1	TM3	Timer3 match value
0x2C	R/W	32	0x—	Tm3Match2	TM3	Timer3 match value
0x30	R/W	12	0x0	TmCR	None	Timer1, Timer2, Timger3 control registers
0x34	R/W	9	0x0	IntrState	None	Interrupt State of timer
0x38	R/W	9	0x0	IntrMask	None	Interrupt Mask of timer
0x3C	R	32	0x—	TmRevision	None	Timer revision number

4.3.3.2 Register Description

The following subsections describe the timer registers in details.

4.3.3.2.1 Tm1Counter, Tm2Counter, Tm3Counter

Tm1Counter, Tm2Counter, and Tm3Counter are the counter registers of Timer1, Timer2, and Timer3, respectively. When the timer is disabled, Tm(1 ~ 3)Counter will hold the value.

Table 4-84. Tm1Counter, Tm2Counter, Tm3Counter Registers

Bit	Name	Type	Description
32	Tm1Counter	Read/Write	Tm1Counter will be enabled if the TM1 option is configured.
	Tm2Counter		Tm2Counter will be enabled if the TM2 option is configured.
	Tm3Counter		Tm3Counter will be enabled if the TM3 option is configured.

4.3.3.2.2 Tm1Load, Tm2Load, Tm3Load

Tm1Load, Tm2Load, and Tm3Load are the auto-reload registers for Timer1, Timer2, and Timer3, respectively. When the Timer(1~ 3) overflow occurs, the value of Tm(1 ~ 3)Load will be loaded into the Tm(1 ~ 3)Counter register. This value is used to set the period between two overflows or underflows.

Note: An overflow indicates that the counter has exceeded the supported limits. That is, the counter counts to higher than 0XFFFF_FFFF or counts to lower than 0X0000_0000.

Table 4-85. Tm1Load, Tm2Load, Tm3Load Registers

Bit	Name	Type	Description
32	Tm1Load	Read/Write	Tm1Counter will be enabled if the TM1 option is configured.
	Tm2Load		Tm2Counter will be enabled if the TM2 option is configured.
	Tm3Load		Tm3Counter will be enabled if the TM3 option is configured.

4.3.3.2.3 Tm1Match1, Tm2Match1, Tm3Match1

Tm1Match1, Tm2Match1, and Tm3Match1 are the match registers of Timer1, Timer2, and Timer3, respectively. When the value of Tm(1 ~ 3)Counter equals to the value of Tm(1 ~ 3)Match1 and the Tm(1 ~ 3) enable bit is set, tm(1 ~ 3)_intr will be triggered.

Table 4-86. Tm1Match1, Tm2Match1, Tm3Match1 Registers

Bit	Name	Type	Description
32	Tm1Match1	Read/Write	Tm1Counter will be enabled if the TM1 option is configured.
	Tm2Match1		Tm2Counter will be enabled if the TM2 option is configured.
	Tm3Match1		Tm3Counter will be enabled if the TM3 option is configured.

4.3.3.2.4 Tm1Match2, Tm2Match2, Tm3Match2

Tm1Match2, Tm2Match2, and Tm3Match2 are the match registers of Timer1, Timer2, and Timer3, respectively. When the value of Tm(1 ~ 3)Counter equals to the values of Tm(1 ~ 3)Match2 and Tm(1 ~ 3) enable bits, tm(1~3)_intr will be triggered. The counter will keep counting and no reload will occur.

Table 4-87. Tm1Match2, Tm2Match2, Tm3Match2 Registers

Bit	Name	Type	Description
32	Tm1Match2	Read/Write	Tm1Counter will be enabled if the TM1 option is configured.
	Tm2Match2		Tm2Counter will be enabled if the TM2 option is configured.
	Tm3Match2		Tm3Counter will be enabled if the TM3 option is configured.

4.3.3.2.5 TmCR

TmCR is the control register of the timer that controls the timer enable/disable function, clock source selection, and overflow mode. When Tm1Enable is set to '0', tm1_intr will never be triggered and the value of Tm1Counter will be held. When Tm1Clock in TmCR is set to '0', Tm1Counter will use PCLK as the clock source; otherwise, Tm1Counter will use EXT1CLK as the clock source. When Tm1OFEnable is set, tm1_intr will be triggered once Timer1 overflow occurs. Otherwise, tm1_intr will only be triggered when the value of Tm1Counter equals to the value of Tm1Match1 or Tm1Match2.

Table 4-88 is the bit assignments of the TmCR resister.

Table 4-88. TmCR Register

Bit	Name	Description
0	Tm1Enable	Timer1 enable 0: Disable 1: Enable
1	Tm1Clock	Timer1 clock source selection 0: PCLK 1: EXT1CLK
2	Tm1OFEnable	Timer1 overflow interrupt enable 0: Disable 1: Enable

Bit	Name	Description
3	Tm2Enable	Timer2 enable 0: Disable 1: Enable
4	Tm2Clock	Timer2 clock source selection 0: PCLK 1: EXT2CLK
5	Tm2OFEnable	Timer2 overflow interrupt enable 0: Disable 1: Enable
6	Tm3Enable	Timer3 enable 0: Disable 1: Enable
7	Tm3Clock	Timer3 clock source selection 0: PCLK 1: EXT3CLK
8	Tm3OFEnable	Timer3 overflow interrupt enable 0: Disable 1: Enable
9	Tm1UpDown	Timer1 up or down counting 0: Down counting When Tm1Counter decreases to ‘0’ from the initial value, it will underflow and Tm1Counter will auto-reload the value of Tm1Load to Tm1Counter. 1: Up counting When Tm1Counter increases to 0xffff from the initial value, it will overflow and Tm1Counter will auto-reload the value of Tm1Load to Tm1Counter.
10	Tm2UpDown	Timer2 up or down counting 0: Down counting When Tm2Counter decreases to ‘0’ from the initial value, it will underflow and Tm2Counter will auto-reload the value of Tm2Load to Tm2Counter. 1: Up counting When Tm2Counter increases to 0xffff from the initial value, it will overflow and Tm2Counter will auto-reload the value of Tm2Load to Tm2Counter.

Bit	Name	Description
11	Tm3UpDown	Timer3 up or down counting 0: Down counting When Tm3Counter decreases to '0' from the initial value, it will underflow and Tm1Counter will auto-reload the value of Tm3Load to Tm3Counter. 1: Up counting When Tm3Counter increases to 0xffff from the initial value, it will overflow and Tm3Counter will auto-reload the value of Tm3Load to Tm3Counter.

4.3.3.2.6 IntrState

The IntrState register is the interrupt state register of timer. When tmr_intr is asserted, CPU must check which interrupt has occurred by reading the IntrState register. Each bit of the IntrState register must indicate a corresponding interrupt that has occurred. All bits of IntrState must be cleared by firmware. If some bits of pwdata are set, the related bits of IntrState will be cleared when users write to the IntrState address.

Table 4-89. IntrState Register

Bit	Name	Description
0	Tm1Match1	Tm1Match1 interrupt 0: No effect 1: Tmr1Counter value equals to the value in the Tm1Match1 register.
1	Tm1Match2	Tm1Match2 interrupt 0: No effect 1: Tmr1Counter value equals to the value in the Tm1Match2 register.
2	Tm1Overflow	Tm1Overflow interrupt 0: No effect 1: Tmr1Counter overflow
3	Tm2Match1	Tm2Match1 interrupt 0: No effect 1: Tmr2Counter value equals to the value in the Tm2Match1 register.
4	Tm2Match2	Tm2Match2 interrupt 0: No effect 1: Tmr2Counter value equals to the value in the Tm2Match2 register.
5	Tm2Overflow	Tm2Overflow interrupt 0: No effect 1: Tmr2Counter overflow

Bit	Name	Description
6	Tm3Match1	Tm3Match1 interrupt 0: No effect 1: Tmr3Counter value equals to the value in the Tm3Match1 register.
7	Tm3Match2	Tm3Match2 interrupt 0: No effect 1: Tmr3Counter value equals to the value in theTm3Match2 register.
8	Tm3Overflow	Tm3Overflow interrupt 0: No effect 1: Tmr3Counter overflow

4.3.3.2.7 IntrMask

The IntrMask register is an interrupt mask register. If one bit of the IntrMask register is set, the corresponding bit of the IntrState register will be masked and will never assert tmr_intr even though the corresponding bit of the ntrState register is logic one.

Table 4-90. IntrMask Register

Bit	Name	Description
0	MTm1Match1	Mask the Tm1Match1 interrupt 0: No effect 1: Tm1Match1 in the IntrState register will be masked.
1	MTm1Match2	Mask the Tm1Match2 interrupt 0: No effect 1: Tmr1Match2 in the IntrState register will be masked.
2	MTm1Overflow	Mask the Tm1Overflow interrupt 0: No effect 1: Tmr1Overflow in the IntrState register will be masked.
3	MTm2Match1	Mask the Tm2Match1 interrupt 0: No effect 1: Tm2Match1 in the IntrState register will be masked.
4	MTm2Match2	Mask the Tm2Match2 interrupt 0: No effect 1: Tmr2Match2 in the IntrState register will be masked.
5	MTm2Overflow	Mask the Tm2Overflow interrupt 0: No effect 1: Tmr2Overflow in the IntrState register will be masked.

Bit	Name	Description
6	MTm3Match1	Mask the Tm3Match1 interrupt 0: No effect 1: Tm3Match1 in the IntrState register will be masked.
7	MTm3Match2	Mask the Tm3Match2 interrupt 0: No effect 1: Tmr3Match2 in the IntrState register will be masked.
8	MTm3Overflow	Mask the Tm3Overflow interrupt 0: No effect 1: Tmr3Overflow in the IntrState register will be masked.

4.3.3.2.8 TmRevision

The TmRevision register contains the revision information.

Table 4-91. TmRevision Register

Bit	Name	Type	Description
32	TmRevision	Read	Revision number

4.4 WatchDog Timer

4.4.1 General Description

WatchDog Timer (WDT) is used to prevent the system from infinite looping if software is trapped in deadlock. In the normal operation, users can restart WDT at regular intervals before the counter counts down to 0. If the counter does reach 0, WDT will generate one signal or a combination of signals, such as the system reset, system interrupt, or external interrupt to reset the system or interrupt the system, or interrupt an external device.

4.4.2 Features

- During timeout, outputs have the combination of the following signals:
  - System reset
  - System interrupt
- Includes 32bit down counter
- Supports internal or external clock source selection
- Variable time-out periods of reset
- Includes access protection

4.4.3 Programming Model

4.4.3.1 Summary of WatchDog Timer Registers

Table 4-92 lists and describes the summary of the WatchDog timer registers.

Table 4-92. Summary of WatchDog Timer Registers

Offset	Type	Width	Reset Value	Name	Config.	Description
0x00	R	32	0x3EF1480	WdCounter	None	WatchDog timer counter register
0x04	R/W	32	0x3EF1480	WdLoad	None	WatchDog timer counter auto-reload register The auto-reload register is set to 0x3EF1480 as default.
0x08	W	16	0x0000	WdRestart	None	WatchDog timer counter restart register When writing 0x5AB9 to this register, the WatchDog timer will auto-reload WdLoad to Wdcounter and restart counting.
0x0C	R/W	5	0x0	WdCR	None	WatchDog timer control register
0x10	R	1	0x0	WdStatus	None	WatchDog timer status register This register will be set when counter reaches 0. 0: Counter does not reach 0. 1: Counter reaches 0.
0x14	W	1	0x0	WdClear	None	WatchDog timer is cleared. Writing 1 or 0 to this register will clear WdStatus.

Offset	Type	Width	Reset Value	Name	Config.	Description
0x18	R/W	8	0xFF	WdIntrlen	None	WatchDog timer interrupt length This register controls the length of wd_rst, and wd_intr. The default value is 0xFF.
0x1C	R	32	0x—	WdRevision	None	Revision number

4.4.3.2 Register Description

The following subsections describe the detailed information of the WatchDog timer registers.

4.4.3.2.1 WdCounter (Offset = 0x00)

Table 4-93. WdCounter Register (Offset = 0x00)

Bit	Name	Type	Description
32	WdCounter	R	The WdCounter register contains the current counter value. When reset, the WdCounter register is set to 0x3EF1480. After programmers write 0x5AB9 to WdRestart, the value of WdLoad will be loaded into WdCounter. When the WdCounter register starts to decrease WdCR[0] only once, the WatchDog timer enable bit, will be set. If the WatchDog timer is disabled, WdCounter will hold the value. If WdCR[4] is set, the register will be driven by an external clock, and WdCounter will decrease at the EXTCLK frequency. This register is read only.

4.4.3.2.2 WdLoad (Offset = 0x04)

Table 4-94. WdLoad Register (Offset = 0x04)

Bit	Name	Type	Description
32	WdLoad	R/W	The WdLoad register contains the value which will be loaded into WdCounter. When reset or restarted, the value of WdLoad will be automatically loaded into the WdCounter register. The reset value of WdLoad is 0x3EF1480.

4.4.3.1 WdRestart (Offset = 0x08)

Table 4-95. WdRestart Register (Offset = 0x08)

Bit	Name	Type	Description
32	WdRestart	W	The WdRestart register is used to avoid the unexpected counting. If the programmer writes 0x5AB9 to this register, the WatchDog timer counter will load WdLoad into the WdCounter register and the WatchDog timer counter will restart to decrease. After finishing the write cycle, WdRestart will automatically be reset to 0.

4.4.3.1.1 WdCR (Offset = 0x0C)

WdCR is the WatchDog timer control register. Table 4-96 lists the definitions of enable bits, reset bits, interrupt bits, external enable bits, and clock source bits.

Table 4-96. WdCR Register (Offset = 0x0C)

Bit	Name	Reset	Description
0	WdEnable	0x0	WatchDog timer enable 0: Disable 1: Enable
1	WdRst	0x0	WatchDog timer system reset enable 0: Disable 1: Enable
2	WdIntr	0x0	WatchDog timer system interrupt enable 0: Disable 1: Enable
3	WdExt	0x0	WatchDog timer external signal enable 0: Disable 1: Enable
4	WdClock	0x0	WatchDog timer clock source 0: PCLK 1: EXTCLK

4.4.3.1.2 WdStatus (Offset = 0x10)

Table 4-97. WdStatus Register (Offset = 0x10)

Bit	Name	Type	Description
1	WdStatus	R	The WdStatus register records to see if the WatchDog timer reaches 0 or not. It is read only.

4.4.3.1.3 WdClear (Offset = 0x14)

Table 4-98. WdClear Register (Offset = 0x14)

Bit	Name	Type	Description
1	WdClear	W	When writing 1 to this register, WdStatus will be cleared.

4.4.3.1.4 WdIntrlen (Offset = 0x18)

Table 4-99. WdIntrlen Register (Offset = 0x18)

Bit	Name	Type	Description
8	WdIntrlen	R/W	The WdIntrlen register decides the duration of the assertion of wd_rst, and wd_intr signals. The default value is 0xFF, which means that the default assertion duration of wd_rst, and wd_intr is 256 clock cycles.

4.4.3.1.5 WdRevision (Offset = 0x1C)

Table 4-100. WdRevision Register (Offset = 0x1C)

Bit	Name	Type	Description
32	WdRevision	R	Revision number

Chapter 5

Encryption

This chapter contains the following section:

5.1 AES-DES Cipher Coprocessor

5.1.1 General Description

The AES-DES cipher coprocessor provides an efficient hardware implementation of the DES/Triple-DES/ AES algorithm for high-performance encryption and decryption, which can be applied to a variety of applications.

In the DES/Triple-DES configuration, AES-DES supports four block cipher modes, including ECB, CBC, CFB, and OFB. In the AES configuration, it supports five block cipher modes, including ECB, CBC, CTR, CFB, and OFB.

5.1.1.1 Terminology

Term	Description
AES	Advanced Encryption Standard
DES	Data Encryption Standard
Triple_DES	Triple Data Encryption Standard
Block Size	Block size in this document represents the data counts processed during one cipher operation cycle
Cipher Operation Cycle	Cycle times between input data and output data of security engine
CBC	Cipher Block Chaining Mode
ECB	Electronic Codebook Mode
CTR	Counter Mode
CFB	8bit Cipher Feedback Mode
OFB	Output Feedback Mode
INFIFO	Input FIFO
OUTFIFO	Output FIFO

5.1.2 Features

- DES/Triple-DES encryption/decryption compliant with NIST standard
- AES 128bit/192bit/256bit encryption/decryption compliant with NIST standard
- Supports block cipher mode
  - DES/Triple-DES
    - ECB mode
    - CBC mode
    - CFB mode
    - OFB mode
  - AES
    - ECB mode
    - CBC mode
    - CFB mode
    - OFB mode
    - CTR mode

5.1.3 Functional Description

5.1.3.1 DMA Engine

The DMA block supports the data transfer from the source address to the destination address. Before enabling DMA, users have to program the related registers, such as the source address, destination address, and total transfer size. When the DMA engine completes the total transfer size, an interrupt will be triggered. Users can set the DMAStop bit of the DMACtrl register to disable the DMA function. When the DMAStop bit is set, FIFO will be reset, and the DMA stop interrupt will be triggered.

5.1.3.2 Data FIFO

The data FIFO is composed of input FIFO (INFIFO) and output FIFO (OUTFIFO). DMA reads data from the AHB bus and pushes it into INFIFO. DMA pops data from OUTFIFO and writes it to the AHB bus. When the security engine is active, it pops the original data from INFIFO and pushes the result data into OUTFIFO. The status of the data FIFO can be read from the FIFO status register.

5.1.3.3 Security Engine

The security engine supports three cipher algorithms: AES, DES, and Triple-DES, and up to five operation modes for these algorithms: ECB, CBC, CTR, CFB, and OFB. For AES, the engine provides five operation modes (ECB, CBC, CTR, CFB, and OFB) and three key sizes (128bits, 192bits, and 256bits), which are determined by the register settings. For DES and Triple DES, it provides four operation modes (ECB, CBC, CFB, and OFB). The key size of DES is 64bits. The key size of Triple-DES is 192bits. Users can program the EncrypControl register to set to specific value for specific application. When DMA is enabled and the data in FIFO is ready, the security engine will encrypt or decrypt data from INFIFO and return the resulting data to OUTFIFO. For each encryption/decryption cycle, one block size of data will be processed. The operation will be repeated until the total transfers are completed. The block size for each mode is shown below.

Table 5-1. Block Size of Each Mode

Algorithm	AES	DES	AES/DES
Mode	ECB/CBC/OFB/CTR	ECB/CBC/OFB	CFB
Block size	16bytes	8bytes	1byte

Table 5-2 lists the clock cycles of one cipher operation with different algorithms.

Table 5-2. Cipher Operation Cycle

Algorithm	AES	DES	Triple-DES
Clock cycle	12	16	48

5.1.4 Register Descriptions

5.1.4.1 Summary of Control Registers

Table 5-3. Summary of Control Registers

Address (Offset)	Type	Name	Description	Reset Value
0x00	R/W	EncryptControl	Encryption/Decryption control register	00000000
0x04	-	-	Reserved	-
0x08	R	FIFOStatus	IN/OUT FIFO status information	00000005

Address (Offset)	Type	Name	Description	Reset Value
0x0C	R	PErrStatus	Parity error information	00FFFFFF
0x10	R/W	Key0	0th 32bits of the key stream	00000000
0x14	R/W	Key1	1st 32bits of the key stream	00000000
0x18	R/W	Key2	2nd 32bits of the key stream	00000000
0x1C	R/W	Key3	3rd 32bits of the key stream	00000000
0x20	R/W	Key4	4th 32bits of the key stream	00000000
0x24	R/W	Key5	5th 32bits of the key stream	00000000
0x28	R/W	Key6	6th 32bits of the key stream	00000000
0x2C	R/W	Key7	7th 32bits of the key stream	00000000
0x30	R/W	IV0	0th 32bits of the initial vector stream	00000000
0x34	R/W	IV1	1st 32bits of the initial vector stream	00000000
0x38	R/W	IV2	2nd 32bits of the initial vector stream	00000000
0x3C	R/W	IV3	3rd 32bits of the initial vector stream	00000000
0x40	-	-	Reserved	-
0x44	-	-	Reserved	-
0x48	R/W	DMASrc	DMA source address	00000000
0x4C	R/W	DMADes	DMA destination address	00000000
0x50	R/W	DMATrasSize	DMA total transfer size	00000000
0x54	R/W	DMACtrl	DMA control register	00000000
0x58	R/W	FIFOThold	FIFO threshold	00000010
0x5C	R/W	IntrEnable	Interrupt enable bit	00000000
0x60	R	IntrSrc	Interrupt source register	00000000
0x64	R	MaskedIntrStatus	Masked interrupt status	00000000
0x68	W	IntrClr	Interrupt clear register	00000000
0x6C	-	-	Reserved	-
0x70	R	REVISION	Revision register	-
0x74	R	FEATURE	Feature register	-
0x78	-	-	Reserved	-
0x7C	-	-	Reserved	-
0x80	R	LAST_IV0	Last block IV output for the next block	-
0x84	R	LAST_IV1	Last block IV output for the next block	-
0x88	R	LAST_IV2	Last block IV output for the next block	-
0x8C	R	LAST_IV3	Last block IV output for the next block	-

5.1.4.2 Register Descriptions

The following subsections describe the APB Bridge registers in details.

5.1.4.2.1 Encryption Control Register (EncryptControl) (Offset = 0x00)

This register stores information to control the functions of the security engine.

Table 5-4. Encryption Control Register (EncryptControl) (Offset = 0x00)

Bit	Name	Type	Description
[31:9]	-	-	Reserved
8	pchk	R/W	Parity check used only in the DES mode 1: Enable the parity check 0: Disable the parity check
7	first	R/W	This bit specifies the first data block to load the initial vector. When operating in the CBC, CTR, CFB, or OFB mode, users should specify the first block to load the initial vector for the cipher operation. 1: Specify the first block of the cipher operation 0: Non-first block This bit automatically returns to 0 when finishing one cipher operation.
[6:4]	emode	R/W	Operation mode selection 000: ECB mode 001: CBC mode 010: CTR mode 100: CFB mode 101: OFB mode
[3:1]	method	R/W	Encryption algorithm selection 3'b000: DES 001: Triple-DES 100: AES-128 (Key size = 128bits) 101: AES-192 (Key size = 192bits) 110: AES-256 (Key size = 256bits)
0	decrypt	R/W	Decryption or encryption stage 1: Decryption stage 0: Encryption stage

5.1.4.2.2 FIFO Status Register (FIFOStatus) (Offset = 0x08)

This register stores the FIFO information, which reflects the current entities in INFIFO and OUTFIFO.

Table 5-5. FIFO Status Register (Offset = 0x08)

Bit	Name	Type	Description
[31:24]	of_entity	R	OUT FIFO entity count
[23:16]	If_entity	R	IN FIFO entity count
[15:4]	-	-	Reserved
3	Offull	R	OUT FIFO is full.
2	ofempty	R	OUT FIFO is empty.
1	Iffull	R	IN FIFO is full.
0	ifempty	R	IN FIFO is empty.

5.1.4.2.3 Parity Error Register (PErrStatus) (Offset = 0x0C)

This register stores the results of the parity check.

Table 5-6. Parity Error Register (PErrStatus) (Offset = 0x0C)

Bit	Name	Type	Description
31	perr	R	Parity error
[30:24]	-	-	Reserved
[23:0]	perrb	R	Parity error bits of each byte of the DES Keys

5.1.4.2.4 Security Key N Registers (KeyN, Offset = 0x10 ~ 0x2C)

These registers (0x10 ~ 0x2C) store the security key stream information for the cipher operation. For AES-128, the key stream is 128bits (Key0 ~ Key3). For AES-192, the key stream is 192bits (Key0 ~ Key5). For AES-256, the key stream is 256bits (Key0 ~ Key7). For DES, the key stream is 64bits (Key0 ~Key1). For Triple-DES, the key stream is three 64bits, ({Key0, Key1}, {Key2, Key3}, and {Key4, Key5}).

Table 5-7. Security Key N Register (KeyN, Offset = 0x10 ~ 0x2C)

Bit	Name	Type	Description
[31:0]	KeyN	R/W	N^{th} 32bits of key stream

5.1.4.2.5 Initial Vector N Registers (IVN, Offset = 0x30 ~ 0x3C)

These registers (0x30 ~ 0x3C) store the initial vector information for the cipher operation. For AES, the initial vector is 128bits (IV0 ~ IV3). For DES/Triple-DES, the initial vector is 64bits (IV0 ~ IV1).

Table 5-8. Initial Vector N Registers (IVN, Offset = 0x30 ~ 0x3C)

Bit	Name	Type	Description
[31:0]	IVN	R/W	N^{th} 32bits of the initial vector stream

5.1.4.2.6 DMA Source Address Register (DMASrc, Offset = 0x48)

This register stores information about the DMA source address from which DMA reads data to INFIFO for the cipher operation.

Table 5-9. DMA Source Address Register (DMASrc, Offset = 0x48)

Bit	Name	Type	Description
[31:0]	DMASrc	-	Source address of DMA

5.1.4.2.7 DMA Destination Address Register (DMADes, Offset = 0x4C)

This register stores information about the DMA destination address to which DMA writes data from OUTFIFO for the cipher operation.

Table 5-10. DMA Destination Address Register (DMADes, Offset = 0x4C)

Bit	Name	Type	Description
[31:0]	DMADes	-	Destination address of DMA

5.1.4.2.8 DMA Transfer Size Register (DMATrasSize, Offset = 0x50)

This register stores information about the total transfer size that DMA will transfer. The unit of the value is byte.

Table 5-11. DMA Transfer Size Register (DMATrasSize, Offset = 0x50)

Bit	Name	Type	Description
[31:12]	-	-	Reserved
[11:0]	TranSize	R/W	Total bytes of a DMA transfer in a range from 0 to 4095 The bytes must be the multiples of the block size, which is determined by the modes of the cipher operation specified in EncryptControl. The block size for all modes of AES, except the CFB mode, is 16bytes (128bits). The block size for all modes of DES, except the CFB mode, is 8bytes (64bits). The block size of the CFB mode of AES and DES is 1byte (8bits).

5.1.4.2.9 DMA Control Register (DMACtrl, Offset = 0x54)

The information in this register is used to control the DMA functions. When writing '1' to the DmaStop bit, the encryption/decryption action will be immediately stopped, and the interrupt (StopIntr, Register 0x60[2]) will be asserted. The hardware will automatically set the DmaStop bit to '0' when users clear the interrupt (ClrStopIntr, Register 0x68[2]).

Table 5-12. DMA Control Register (DMACtrl, Offset = 0x54)

Bit	Name	Type	Description
[31:3]	-	-	Reserved
2	DmaStop	W	1: Stop DMA 0: No effect
		R	1: DMA stop operation 0: No DMA stop operation
1	-	-	Reserved
0	DmaEn	W	1: Enable DMA 0: No effect
		R	1: Enable DMA 0: Disable DMA

5.1.4.2.10 FIFO Threshold Register (FIFOThold, Offset = 0x58)

This register stores information about the FIFO threshold used as the condition to activate the read or write transfers of DMA.

Table 5-13. FIFO Threshold Register (FIFOThold, Offset = 0x58)

Bit	Name	Type	Description
[31:16]	-	-	Reserved
[15:8]	OUTFIFOThold	R/W	Watermark of the OUTFIFO data count activates the engine to pop data from OUTFIFO and writes data to the memory. Unit: Byte
[7:0]	INFIFOThold	R/W	Watermark of the INFIFO data count activates the engine to read data from the memory and pushes data to INFIFO. Unit: Byte

5.1.4.2.11 Interrupt Enable Register (IntrEnable, Offset = 0x5C)

This register stores information to enable or disable the interrupt source.

Table 5-14. Interrupt Enable Register (IntrEnable, Offset = 0x5C)

Bit	Name	Type	Description
[31:3]	-	-	Reserved
2	StopIntrEn	R/W	0: Disable the DMA stop interrupt source 1: Enable the DMA stop interrupt source
1	ErrIntrEn	R/W	0: Disable the DMA receive hresp error interrupt source 1: Enable the DMA receive hresp error interrupt source
0	DoneIntrEn	R/W	0: Disable the DMA transfer Done interrupt source 1: Enable the DMA transfer Done interrupt source

5.1.4.2.12 Interrupt Source Register (IntrSrc, Offset = 0x60)

This register stores information about the interrupt source.

Table 5-15. Interrupt Status Register (IntrSrc, Offset = 0x60)

Bit	Name	Type	Description
[31:3]	-	-	Reserved

Bit	Name	Type	Description
2	StopIntr	R	DMA stop interrupt source before mask
1	ErrIntr	R	DMA receive hresp error interrupt source before mask
0	DoneIntr	R	DMA transfer done interrupt source before mask

5.1.4.2.13 Masked Interrupt Status (MaskedIntrStatus, Offset = 0x64)

This register stores information about the interrupt after mask.

Table 5-16. Masked Interrupt Status Register (MaskedIntrStatus, Offset = 0x64)

Bit	Name	Type	Description
[31:3]	-	-	Reserved
2	StopIntrMsk	R	DMA stop interrupt after mask
1	ErrIntrMsk	R	DMA receive hresp error after mask
0	DoneIntrMsk	R	DMA transfer done Interrupt after mask

5.1.4.2.14 Interrupt Clear Register (IntrClr, Offset = 0x68)

This register is used to clear the interrupt.

Table 5-17. Interrupt Clear Register (IntrClr, Offset = 0x68)

Bit	Name	Type	Description
[31:3]	-	-	Reserved
2	ClrStopIntr	W	0: No effect 1: Clear the DMA stop interrupt
1	ClrErrIntr	W	0: No effect 1: Clear the DMA receive hresp error interrupt
0	ClrDoneIntr	W	0: No effect 1: Clear the DMA transfer done interrupt

5.1.4.2.15 Revision Register (REVISION, Offset = 0x70)

This register stores the revision information.

Table 5-18. Revision Register (REVISION, Offset = 0x70)

Bit	Name	Type	Description
[31:3]	REVISION	R	Revision number

5.1.4.2.16 Feature Register (FEATURE, Offset = 0x74)

This register stores information of hardware feature.

Table 5-19. Feature Register (FEATURE, Offset = 0x74)

Bit	Name	Type	Description
[31:8]	-	-	Reserved
[7:0]	FIFO_DEPTH	R	IN/OUT FIFO depth Unit: Byte

5.1.4.2.17 Last Initial Vector N Registers (Last_IV N, Offset = 0x80 ~ 0x8C)

These registers (0x80 ~ 0x8C) store information about the initial vectors of the last block size for the cipher operation. For AES, the initial vector is 128bits (IV0 ~ IV3). For DES/Triple-DES, the initial vector is
64bits (IV0 ~ IV1). For some modes, such as the ECB mode or IV2 ~ IV3 for DES/Triple-DES, the register value is '0.'

Table 5-20. Last Initial Vector *N* Registers (Last_IV N, Offset = 0x80 ~ 0x8C)

Bit	Name	Type	Description
[31:8]	Last_IV *N*	R	N^{th} 32bits of the last initial vector stream

5.1.5 Programming Sequence

The programming sequence is as follows:

1. Set the EncryptControl register for the encryption/decryption operations
2. Set Initial Vector IV to a mode of a cipher algorithm, such as CBC, CTR, CFB, or OFB. There are four 32bit registers (IV0 ~ IV3) for the initial vector setting. For the AES algorithm, IV is 16bytes (IV0 ~ IV3). For the DES algorithm, IV is 8bytes (IV0 ~ IV1). One IVn register is 4bytes of the initial vector. The byte sequence is in the most-significant-bit-first order. The initial vectors for different cipher algorithms are shown below:
   DES/Triple-DES: IV[63:0] = {IV0, IV1}
   AES: IV[127:0] = {IV0, IV1, IV2, IV3}

Table 5-21. Byte Sequence of Initial Vector

IV Register	IV0				IV1				IV2				IV3	
Byte sequence	0	1	2	3	4	5	6	7	8	9	10	11	12	…
IV[63:0]	Initial vector													

3. Set the key values for different cipher algorithms, such as AES, DES, and Triple-DES. There are eight 32bit registers (Key0 ~ Key7) for the key value setting. For the AES-128 algorithm, the key value is 16bytes (Key0 ~ Key3). For the AES-192 algorithm, the key value is 24bytes (Key0 ~ Key5). For the AES-256 algorithm, the key value is 32bytes (Key0 ~ Key7). For the DES algorithm, the key value is 8bytes (Key0 ~ Key1). For the triple-DES algorithm, there are three 8bytes ({Key0, Key1}, {Key2, Key3}, {Key4, Key5}) key values. The byte sequence is in the most-significant-bit-first order.

The following tables list the key values of different cipher algorithms:

AES-128 Algorithm

AES-128Key[127:0] = {Key0, Key1, Key2, Key3}

AES-128Key[127:96] = Key0

AES-128Key[95:64] = Key1

AES-128Key[63:0] = Key2

AES-128Key[31:0] = Key3

Table 5-22. AES-128 Key Stream of Byte Sequence

Key Register	IV0				IV1				IV2				IV3	
Byte sequence	0	1	2	3	4	5	6	7	8	9	10	11	12	…
key[127:0]	AES-128 key value													

AES-192 Algorithm

AES-192Key[191:0] = {Key0, Key1, Key2, Key3, Key4, Key5}

AES-192Key[191:160] = Key0

AES-192Key[159:128] = Key1

AES-192Key[127:96] = Key2

AES-192Key[95:64] = Key3

AES-192Key[63:0] = Key4

AES-192Key[31:0] = Key5

Table 5-23. AES-192 Key Stream of Byte Sequence

Key Register	Key0				Key1				Key2				Key5	
Byte sequence	0	1	2	3	4	5	6	7	8	9	10	…	…	23
Key[191:0]	AES-192 key value													

AES-256 Algorithm

AES-256Key[255:0] = {Key0, Key1, Key2, Key3, Key4, Key5, Key6, Key7}

AES-256Key[255:224] = Key0

AES-256Key[223:192] = Key1

AES-256Key[191:160] = Key2

AES-256Key[159:128] = Key3

AES-256Key[127:96] = Key4

AES-256Key[95:64] = Key5

AES-256Key[63:0] = Key6

AES-256Key[31:0] = Key7

Table 5-24. AES-256 Key Stream of Byte Sequence

Key Register	Key0				Key1				Key2					Key7
Byte sequence	0	1	2	3	4	5	6	7	8	9	10	…	…	31
Key[255:0]	AES-256 key value													

DES Algorithm

DESkey[63:0] = {Key0, Key1}

DESkey[63:32] = Key0

DESkey[31:0] = Key1

Table 5-25. DES Key Stream of Byte Sequence

Key Register	Key0				Key1				Key2				Key3	…
Byte sequence	0	1	2	3	4	5	6	7	8	9	10	11	12	…
key[63:0]	DES key value								Reserved					

Triple-DES Algorithm

Triple-DESKey1[63:0] = {Key0, Key1}

Triple-DESKey1[63:32] = Key0

Triple-DESkey1[31:0] = Key1

Triple-DESKey2[63:0] = {Key2, Key3}

Triple-DESKey2[63:32] = Key2

Triple-DESKey2[31:0] = Key3

Triple-DESKey3[63:0] = {Key4, Key5}

Triple-DESKey3[63:32] = Key4

Triple-DESKey3[31:0] = Key5

Table 5-26. Triple DES Key Stream of Byte Sequence

Key Register	Key0				Key1				Key2				Key5	
Byte sequence	0	1	2	3	4	5	6	7	8	9	10	…	…	23
Key[191:0]	Triple-DES Key1								Triple-DES Key2				Triple-DES Key3	

To locate the memory space of the input and output data blocks, users should prepare the input data block in specific memory space. The cipher or plain text is represented in the byte sequence order, such as b0, b1, b2, b3, and b4. The index indicates an address offset. For example, 'b0' indicates data b0 resided in offset 0 and b1 resided in offset 1. The endianness in this design is little endian. In order for the system to function correctly, all modules accessing the memory should be of the same endianness.

Set DMA related registers

- Set the DMASrc register for the source address, DMADes register for the destination address, and DMATrasSize register for the total data transfer size from the source to destination. The value of DMATrasSize register represents the byte count and should be multiple of the block size that is determined by the cipher algorithm.
- Set the FIFOThold register to specify the FIFO threshold for DMA to activate the read/write transfers from the memory. Users can set two watermarks in the FIFOThold register as the condition to activate the read or write transfers. The read transfer meets the condition that the entities of INFIFO are less than the value of INFIFOThold or one block size. The write transfer meets the condition that the entities of OUTFIFO are more than the value of OUTFIFOThold or one block size.
- Set the IntrEnable register to control the interrupt signal.
- Set the DmaEn bit of DMACtrl to '1' to activate the DMA engine and the security engine.

After the total transfer size is complete, interrupt will be triggered and DMA will be stopped.

Chapter 6

Memory Interface

This chapter contains the following sections:

6.1 DDR3/DDR2 Memory Controller

6.1.1 General Description

The DDR3/DDR2 memory controller supports various DDR3/DDR2 SDRAMs. This controller uses a burst length of eight for the DDR3/DDR2 memory to accelerate the read and write speeds. The memory controller prefetches the sequential read data for the burst read commands to enhance the data transfer rate.

6.1.2 Features

- Supports maximum of 256Mbytes for each rank
- Uses burst length of eight for DDR3/DDR2 memory to speed-up read and write operations
- Enters DDR3/DDR2 SDRAM self-refresh mode by using hardware handshake signal or software configuration
- Prefetches sequential read data for AHB/AXI burst read commands
- Supports MA tables corresponding to different sizes and types
- Provides two arbitration strategies for channel arbitrations:
  - Two-level round-robin arbitration with channel grant counts
  - Read-write grouping arbitration with R/W commands grant counts
- Supports hardware SPLIT responses for AHB slave interface
- Supports read-only/write-only/normal modes for AHB slave interface
- Supports exclusive accesses with burst length of 0 (SINGLE) for AXI slave interface

6.1.3 Functional Description

6.1.3.1 Operation Modes

The DDR controller provides two operation modes: The normal mode and the self-refresh mode. When the system is powered on and a hardware reset is terminated, it will be initialized by programming the control register to enter the normal mode. In the normal mode, all operations run at the full speed. In order to save power, the self-refresh mode is implemented to minimize the power dissipation of the memory module. The self-refresh mode can be entered by hardware handshaking or software programming. To properly operate the self-refresh mode, users should follow the steps below. The AHB/AXI commands should be stopped before issuing the self-refresh request to ensure a proper operation.

6.1.3.2 Refresh Function

The auto-refresh command prevents the data loss in DDR3/DDR2 SDRAM. The period of a refresh command can be programmed depending on the speed of the system clock and the DDR3/DDR2 SDRAM specification. This value is usually documented in the DDR3/DDR2 SDRAM specification as a specified timing interval in which all rows (row_num) should be completely refreshed to prevent data loss. The average refresh interval can be 7.8μs (Commercial) or 3.9μs (Industrial). The DDR controller allows the posted refresh commands until the threshold is achieved. Bits[15:13] of the register offset 0x00 are the threshold. The recommended value is 4. "manual_refresh" is a hardware signal to trigger the auto-refresh command despite of archiving the post-refresh command threshold or not.

6.1.3.3 Automatic Power-down

Users can program the automatic power-down enable register (Offset = 0x2C, bit 12) to save power at the system idle state. The controller can detect the empty AHB/AXI command queue and wait until the counter forces DDRx SDRAM to enter the active power-down mode (CKE pin is pulled low). In the power-down mode, when the auto-refresh timer is up, the DDR controller will automatically pull the CKE pin high and issue an auto-refresh command. Once the auto-refresh command is issued, if there is no other command to be issued, the automatic power-down state will be re-entered until a valid AHB/AXI command is issued from the master.

6.1.3.4 Channel Arbitration

The channel arbiter is responsible for selecting a granted channel which issues the commands to the memory controller. The arbitration policies are described in the following subsections. There are two arbitration modes in the arbiter:

(1) Two-level round-robin arbitration with channel grant counts.

(2) Read-Write grouping arbitration with R/W group grant counts and channel grant counts.

6.1.3.4.1 Two-level Round-robin Arbitration with Channel Grant Counts

All channels are divided into two levels: The high-priority level and the low-priority level. Users can specify the high priority channels by setting the register (Offset = 0x30). For example, five channels are enabled: Channel 2 and channel 4 are specified as the high-priority channels; other channels (0, 1, and 3) are specified as the low-priority channels. The low-priority group will be a member of the high-priority group. The grant sequences will be:

CH2 → CH4 → CH0 → CH2 → CH4 → CH1 → CH2 → CH4 → CH3 → CH2 → CH4 → CH0 ……

Each channel has one 4bit grant window count (Offset = 0x34 and Offset = 0x38). When the granted channel gains the grant and issues one command to the memory controller, its grant window count will be decreased by one. Once the grant window count of the granted channel is decreased to zero, the arbiter will perform the re-arbitration. However, if the granted channel de-asserts its request before the grant window count becomes zero, the re-arbitration will be performed.

6.1.3.4.2 Read-Write Grouping Arbitration with R/W Group Counts and Channel Grant Counts

Besides the two-level round-robin algorithm, an advanced mechanism, "Read/Write group", is used to reduce the SDRAM "Write-to-Read" (WtR) wait cycles. The key concept is to merge the same command type (Read with read or write with write) to reduce the occurrence of Write to Read turnaround probabilities. The "Read-Write group" arbitration can be enabled by programming the Channel Arbitration Setup Register (Offset = 0x30, bit 31). If the Read-Write group mechanism is enabled, all channels will be classified into the read or write arbitration group. **Please note that if the "Read-Write group" arbitration is enabled, all channels should be set to the same priority level, either all low-priority level or all high-priority level.**

Group Re-arbitration

A programmable counter, "group_grant_count" (Offset = 0x30), defines the maximum command count to be issued for a read or write group. Maximum group_grant_count*2 commands will be serviced once a read or write group is granted. Each time a channel issues a command to the memory controller, the allowed command counts will be decreased by one. When "group_grant_count" becomes zero, the group-level re-arbitration will be performed. Consequently, the arbiter will select an arbitration group as new active group and reload the setting of "group_grant_count" for counting.

Channel Re-arbitration

Similar to the two-level round robin arbitration, when the grant window count of a channel becomes zero, the channel re-arbitration will be performed. The arbiter will select a channel belongs to the same Read or Write attributes with the current R/W granted group if "group_grant_count" is not decreased to zero . For example, if channel 0 and channel 2 belong to the Read group, channel 1 and channel 3 belong to the Write group. For group_grant_count = 8, ch0, ch1, ch2, ch3 grant_count = 2, the arbitration may look like this:

Ch0 Read → Ch2 Read → Ch0 Read → Ch2 Read → Ch1 Write → Ch3 Write → Ch1 Write → Ch3 write

4 read 4 read 4 read 4 read 4 write 4 write 4 write 4 write

6.1.4 Programming Model

6.1.4.1 Register Descriptions

The DDR controller registers are shown in Table 6-1. The following subsections describe the DDR controller registers in details.

Table 6-1. Summary of DDR Controller Registers

Offset	Type	Description	Reset Value
0x00	R/W	Memory controller configuration	0x08000000
0x04	R/W	Memory controller state control	0x0003C000
0x08	R/W	Mode register set values for the Mode Register (MR) and Extended Mode Register (EMR)	0x00000000
0x0C	R/W	Mode register set value register of EMR2 and EMR3	0x00000000
0x10	R/W	External rank0/rank1 register	0x00000000
0x14	R/W	Timing parameter 0	0x00000000

Offset	Type	Description	Reset Value
0x18	R/W	Timing parameter 1	0x00000000
0x1C	R/W	Timing parameter 2	0x00000000
0x20	R/W	Data block control register	0x00002F41
0x24	R/W	DLL delay tuning of the DDR PHY read path	0x00000033
0x28	RO/R/W	COMPBLK control	0x00000fbf
0x2C	R/W	Automatic power-down	0x00000000
0x30	R/W	Channel arbitration setup	0x10000000
0x34	R/W	Channel arbiter grant count - A	0x07070707
0x38	R/W	Reserved	
0x3C	R/W	DDR PHY Write/read data access timing control	0x00000000
0x40	R/W	Command flush control	0x00000000
0x44	R/W	Command flush status	0x00000000
0x48	R/W	AHB split control	0x00000000
0x4C	R/W	Update control	0x00000000
0x50	R	Reserved	
0x54	R	Reserved	
0x58	R	Reserved	
0x5C	R/W	PHY Power Control Register	0x00000007
0x60	RO/R/W	Write-leveling control register(Only for the DDR3 mode)	0x00030006
0x64	R/W	Byte 7 ~ byte 4 of the write-leveling control register(Only for the DDR3 mode)	0x00000000
0x68	R/W	Byte 3 ~ byte 0 of the write-leveling control register(Only for the DDR3 mode)	0x00000000
0x6C	R/W	MISC control register(Only for the DDR3 mode)	0x00000000
0x70	R/W	Read-leveling control register(Only for the DDR3 mode)	0x00000003
0x74	R/W	msdly byte control register	0x00000011
0x78	R/W	wdllsel control register	0x00000000
0x7C	R/W	Traffic monitor clock cycle register	0x00000000
0x80	R	Command count for channel 0	0x00000000
0x84	R	Command count for channel 1	0x00000000
0x88	R	Command count for channel 2	0x00000000
0x8C	R	Command count for channel 3	0x00000000
0x90	-	Reserved	-
0x94	-	Reserved	-
0x98	-	Reserved	-
0x9C	-	Reserved	-

Offset	Type	Description	Reset Value
0xA0	R/W	AHB INCR read prefetch length 1	0x03030303
0xA4	R/W	Reserved	0x03030303
0xA8	R/W	Initialization waiting cycle count 1	0x00003E80
0xAC	R/W	Initialization waiting cycle count 2	0x00061A80
0xB0	R/W	QoS control register	0x00000000
0xB4	R/W	QoS command count register A	0x00000000
0xB8	R/W	Reserved	0x00000000
0xBC	R/W	QoS command count register C	0x00000000
0xC0	R/W	Reserved	0x00000000
0xC4	R/W	Channel arbitration setup register B	0x10000000
0xC8	R/W	Channel arbitration setup register C	0x07070707
0xCC	R/W	Reserved	0x07070707
0xD0	R/W	Debug address	0x00000000
0xD4	R/W	Debug address mask	0x00000000
0xD8	R/W	Debug write data	0x00000000
0xDC	R/W	Debug write data mask	0x00000000
0xE0	R/W	Debug master control	0x00000000
0xE4	R/W	Debug access control	0x00000000
0xE8	R/W	Debug policy control	0x00000000
0xEC	R/W	Debug control	0x00000000
0xF0	R	Debug address status	0x00000000
0xF4	R	Debug write data status	0x00000000
0xF8	R	Debug master status	0x00000000
0xFC	R	Debug access status	0x00000000
0x100 ~ 0x12C	-	Reserved	-

6.1.4.2 Memory Controller Configuration Register (Offset = 0x00)

Table 6-2 lists and describes the memory controller configuration register of the DDR controller function register. GDS is used to adjust the read data timing of the DDR controller.

Table 6-2. Memory Controller Configuration Register (Offset = 0x00)

Bit	Name	Type	Description
[31:29]	-	-	Reserved
28	manual refresh enable	R/W	The controller will check the hardware input signal, manual_refresh, to issue one auto-refresh command and then the successive auto-refresh command will be issued based on the tREFI setting. 0: Manual refresh is disabled despite of the hardware manual_refresh signal. 1: Controller will issue one auto-refresh command after receiving the manual_refresh signal.
27	DisAPBErrResp	R/W	This bit disables the APB error response register The APB register interface will respond to PSlVERR under one of the following conditions: • Write/Read to the non-existed registers • Write to the read-only registers 0: Enable the APB error response 1: Disable the APB error response The default value is '1'.
[26:23]	WaitCycle	R/W	When one command is pushed into the reorder engine, it starts to count the cycles until the command is popped out to scheduler. If the cycle count of the command exceeds the wait cycle, this command will have the highest priority for popped out to scheduler. 0: Wait cycle equals to 64 cycles. 1: Wait cycle equals to 2 * 64 cycles. 2: Wait cycle equals to 3 * 64 cycles. … 15: Wait cycle equals to 16 * 64 cycles.

Bit	Name	Type	Description
[22:21]	Cont_Cmd_Limit	R/W	Continuous Command Limitation in scheduler If there is a pending read command in scheduler, there will be at most n write commands that can be issued. If there is a pending write command in scheduler, there will be at most n read commands that can be issued. If Cont_Cmd_Limit = 0, n = 32 If Cont_Cmd_Limit = 1, n = 64 If Cont_Cmd_Limit = 2, n = 128 If Cont_Cmd_Limit = 3, n = 256
20	Dis_Reorder	R/W	DDR command reorder engine can check the bank and row information to reduce the bank conflict performance impact 0: Enable the DDR command reorder 1: Disable the DDR command reorder
19	LPDDR_mode	R/W	Reserved (Always 0, does not support LPDDR) [19:16] = {LPDDR_mode, byone, DDR_mode} 0000: DDR3 1:2 mode 0011: DDR2 1:2 mode 0110: DDR1 1:1 mode 0111: DDR2 1:1 mode 1110: LPDDR1 1:1 mode 1011: LPDDR2 1:2 mode
18	byone	R/W	MClk versus DRAM clock frequency ratio 1: DRAM clock equals to the MClk clock frequency. 0: DRAM clock is twice of the MClk clock frequency. Always 0, it does not support the 1:1 mode.
[17:16]	DDR_mode	R/W	DDR_mode 00: DDR3 mode 01: Reserved 10: Reserved 11: DDR2 mode

Bit	Name	Type	Description
[15:13]	Post-refresh command counts threshold	R/W	Post-refresh command count threshold The DDR controller will post a refresh command until the counts of the posted refresh commands exceed the threshold. 000: No posted refresh command 001: One posted refresh command threshold 010: Two posted refresh command thresholds 011: Three posted refresh command thresholds 100: Four posted refresh command thresholds 101: Five posted refresh command thresholds 110: Six posted refresh command thresholds 111: Seven posted refresh command thresholds
[12:10]	Auto-refresh commands in the initial SDRAM sequence	R/W	DDR2 SDRAM initialization sequence only 000: Not used 001: Two auto-refresh commands 010: Three auto-refresh commands 011: Four auto-refresh commands 100: Five auto-refresh commands 101: Six auto-refresh commands 110: Seven auto-refresh commands 111: Eight auto-refresh commands
[9:8]	Memory width	R/W	Memory width 00: 8bit memory 01: 16bit memory 1x: Reserved
[7:6]	-	-	Reserved

Bit	Name	Type	Description
[5:4]	DDR3/DDR2 memory address mapping table selection (AMTSEL)	R/W	00: The mapping method of the AHB/AXI address to the DDR3/DDR2 SDRAM memory address may be RA, BA, or CA. 01: The mapping method of the AHB/AXI address to the DDR3/DDR2 SDRAM memory address may be BA, RA, or CA. 10: The mapping method of the Memory Bank Swizzle from the AHB address to the DDR3/DDR2 SDRAM memory address may be RA, CA, BA, or CA. **AMTSEL = 2'b10 does not support the 8bit memory.** For the 16bit memory interface, the bank address mapping is inserted between the column addresses. ADDR maps to RA, CA, BA, and CA. Bits[7:6] of the system address are configured as the bank addresses for four banks of the DDR3/DDR2 SDRAM. Bits[8:6] of the system address are configured as the bank addresses for eight banks of the DDR3/DDR2 SDRAM . 11: Reserved
3	-	-	Reserved
[2:0]	Generate DQS sampling window (GDS)	R/W	Before accessing the DDR memory, this field must be scanned from 0 to 7 until the read data are correctly captured. GDS depends on the PCB package trace length between the ASIC chip and DDRx SDRAM. Longer trace length will induce larger GDS value. Please refer to the Faraday DDRx PHY application note for more information. Note: The GDS value should be programmable at any time. If the GDS value is set in the ROM code, it should have the possibility to modify the GDS value by latching from jumper setting or other mechanism if booting from DDR SDRAM is failed.

6.1.4.3 Memory Controller State Control Register (Offset = 0x04)

Table 6-3. Memory Controller State Control Register (Offset = 0x04)

Bit	Name	Type	Description
31	Software reset	R/W	Writing '1' to this bit triggers the software reset to the DDR initial state machine (Only for the DDR3 mode). 1: Software reset request 0: No effect Users should set bit[0] of the initial command to '1' to start the initial sequence after software reset.
[30:25]	-	-	Reserved

Bit	Name	Type	Description
24	Warm Start	R/W	Users can set this bit to “1” to the warm-start controller without initial DDRx SDRAM. This mode can be used when ASIC chip is recovered from the power-down state and DDRx SDRAM is in the self-refresh mode. Please note that when the chip is at the power down state, CKE of DDRx SDRAM should be kept low and the RESET_DRAM (DDR3 only) pin should be kept high from the power-on domain of chip. When the command is completed, this bit will be cleared to zero. 1: Warm start 0: No effect
[23:18]	-	-	Reserved
17	Write leveling fail or hardware disable	RO	Only for the DDR3 mode 1: Any byte write-leveling failed or the hardware disabled in DDR3 SDRAM 0: Write-leveling hardware is enabled and all bytes pass in DDR3 SDRAM.
16	Read leveling fail or hardware disable	RO	Only for the DDR3 mode 1: Any byte read-leveling fail or hardware disable in DDR3 SDRAM 0: Read-leveling hardware is enabled and all bytes pass in DDR3 SDRAM.
15	AHB/AXI command queue empty	RO	1: All AHB/AXI command queues are empty. 0: All AHB/AXI command queues are not empty.
14	Memory controller command queue empty	RO	1: Memory controller command queue is empty. 0: Memory controller command queue is not empty.
13	ZQCL state	RO	1: DDR3 SDRAM is at the ZQCL state. (Only for DDR3) 0: DDR3 SDRAM is at other states.
12	ZQCS state	RO	1: DDR3 SDRAM is at the ZQCS state. (Only for DDR3) 0: DDR3 SDRAM is at other states.
11	Automatic power-down state	RO	1: DDR3 SDRAM is at the automatic power-down state. 0: DDR3 SDRAM is at other states.
10	Self-refresh state	RO	1: DDR3/DDR2 SDRAM is at the self-refresh state. 0: DDR3/DDR2 SDRAM is at other state.
9	Initial State	RO	1: DDR3/DDR2 SDRAM is at the initial state. 0: DDR3/DDR2 SDRAM is not at the initial state.
8	Initial OK	RO	1: DDR3/DDR2 SDRAM initial is completed. 0: DDR3/DDR2 SDRAM initial is not completed.
7	ZQCL	R/W	This bit is programmed to move the controller to enter the targeted state.

Bit	Name	Type	Description
			When the command is completed, this bit will be cleared to zero. 1: Move the memory controller to the ZQCL state (Only for DDR3). 0: No effect
6	ZQCS	R/W	This bit is programmed to move the controller to enter the targeted state. When the command is completed, this bit will be cleared to zero. 1: Move the memory controller to the ZQCS state (Only for DDR3). 0: No effect
[5:4]	Register mode	R/W	00: Mode register 01: Extended mode register 1 10: Extended mode register 2 11: Extended mode register 3
3	Exit self-refresh command	R/W	This bit moves the controller to enter the targeted state. When the command is completed, this bit will be cleared to zero. 1: Move the memory controller from the self-refresh state 0: No effect
2	Self-refresh command	R/W	This bit moves the controller to enter the targeted state. When the command is completed, this bit will be cleared to zero. 1: Move the memory controller to the self-refresh state 0: No effect
1	MRS command	R/W	This bit moves the controller to enter the targeted state. When the command is completed, this bit will be cleared to zero. Users should program MR/EMR/EMRS2/EMRS3 in the registers offsets = 0x08 and 0x0C and offset = 0x04, bits[5:4] mode register. Before starting the MRS command: 1: Move the memory controller to mode register state 0: No effect
0	Initial command	R/W	This bit moves the controller to enter the targeted state. When the command is completed, this bit will be cleared to zero. Setting this bit to “1” will perform initialization of DDR3/DDR2 SDRAM and calibration of the DDR PHY compensation block. After these tasks are completed, bit[8] of initial_ok will be set to “1”. 1: Move the memory controller to the DDR3/DDR2 SDRAM initial state 0: No effect

6.1.4.4 Mode Register Set Value Register of MR and EMR (Offset = 0x08)

This register defines the values of the mode registers and the extended mode registers to be used in the initial sequence. During the initialization of DDR3/DDR2 SDRAM, the controller uses the values in this register for the MRS command. The controller only supports AL (Additive Latency) = '0' in the DDR3/DDR2 mode.

If users want to modify the values of MR and EMR after initialization, Register Offset 0x08 should be programmed, and then bits[5:0] of Register Offsets 0x04 should be set to:
000010: Send MRS command to modify the MR value of DDRx SDRAM
010010: Send MRS command to modify the EMR value of DDRx SDRAM

Notes:

1. The mode register should be set as the fixed Nibble sequential burst length of 8 in the DDR2 mode.
2. The mode register should be set as Nibble sequential burst length fixed 8 in the DDR3 mode.
3. The write recovery time of the mode register should be set to the same value of the Timing Parameter1 register Offset = 0x18, bits[23:20], tWR.
4. AL should be set to '0' for the DDR3/DDR2 applications to improve the Write-to-Read or Read-to-Write performance.
5. In the DDR2 mode, bit26 of the Register Offset= 0x08 DQS_B (DDR2 mode) and bit3 of the Register Offset= 0x020 should be set to the same value. For example, DDR2 SDRAM and DDR PHY should operate in the same DQS mode, either in the differential mode or single-ended mode. DDR3 always operates in different DQS modes.
6. The "DLL_RESET"bit of mode register will be automatically controlled in the DDR initialization sequence.

Table 6-4. Mode Register Set Value Register of MR and EMR (Offset = 0x08)

Bit	Name	Type	Description
[31:30]	-	-	Reserved
[29:16]	Extended mode register	R/W	Extended mode register
[15:14]	-	-	Reserved
[13:0]	Mode register	R/W	Mode register

6.1.4.5 Mode Register Set Value Register of EMR2 and EMR3 (Offset = 0x0C)

This register defines the mode register value and the extended mode register value to be used in the MRS command after initiating the DDR3/DDR2 SDRAM.

If users want to modify the values of EMR2 and EMR3 after initialization, Register Offset 0x10 should be programmed first, and then bits[5:0] of Register Offset 0x04 should be set to:
100010: Send the MRS command to modify the EMR2 value of DDRx SDRAM.
110010: Send the MRS command to modify the EMR3 value of DDRx SDRAM.

Table 6-5. Mode Register Set Value Register of EMR 2 and EMR 3 (Offset = 0x0C)

Bit	Name	Type	Description
[31:30]	-	-	Reserved
[29:16]	Extended mode register 3	R/W	Extended mode register 3
[15:14]	-	-	Reserved
[13:0]	Extended mode register 2	R/W	Extended mode register 2

6.1.4.6 External Rank0/1 Register (Offset = 0x10)

Please note that the Rank1 base address will be the Rank0 base address + Rank0 size and Rank0 will always be enabled.

Table 6-6. External Rank0/1 Register (Offset = 0x10)

Bit	Name	Type	Description
[31:24]	RNK0_BASE	R/W	8bit base address of external Rank0 RNK0 (Chip Select 0) addresses range from {RNK0_BASE, 24'h0} to {{RNK0_BASE,24'h0} + RNK0_SIZE - 1'b1}) RNK1 (Chip Select 1) addresses range from {{RNK0_BASE, 24'h0} + RNK0_SIZE } to {{RNK0_BASE,24'h0} + RNK0_SIZE + RNK1_SIZE - 1'b1}}
[23:17]	-	-	Reserved
16	RNK1_EN	-	Rank1 enable
15	-	-	Reserved

Bit	Name	Type	Description
[14:12]	RNK1_TYPE	R/W	Select the DDR3/DDR2 type based on the MA table for rank1 For example: RAXCAXBA is denoted as 13X10X2. 13X10X2 indicates that the row address is 13bits, the column address is 10bits, and the bank address is 2bits. 000: 13X9X2 001: 13X10X2 010: 14X10X2 011: 12X10X3 100: 13X10X3 101: 14X10X3 110: Reserved 111: Reserved
11	-	-	Reserved
[10:8]	RNK1_SIZE	R/W	Memory size of rank1 000: 16Mbytes 001: 32Mbytes 010: 64Mbytes 011: 128Mbytes 100: 256Mbytes 101: Reserved 110: Reserved 111: Reserved
7	-	-	Reserved
[6:4]	RNK0_TYPE	R/W	Select the DDR3 type based on the MA table for rank0 For example: RAXCAXBA is denoted as 13X10X2. 13X10X2 indicates that the row address is 13bits, the column address is 10bits, and the bank address is 2bits. 000: 13X9X2 001: 13X10X2 010: 14X10X2 011: 12X10X3 100: 13X10X3 101: 14X10X3 110: Reserved 111: Reserved

Bit	Name	Type	Description
3	-	-	Reserved
[2:0]	RNK0_SIZE	R/W	Memory size of rank 0 000: 16Mbytes 001: 32Mbytes 010: 64Mbytes 011: 128Mbytes 100: 256Mbytes 101: Reserved 110: Reserved 111: Reserved

6.1.4.7 Timing Parameter 0 Register (Offset = 0x14)

Please note that all parameters are the MClk cycles (T). The "MClk:DRAM clock" mode is "1:2", the parameters should be set to half value of the DRAM clock cycle. The programmable value of 0 is invalid.

Table 6-7. Timing Parameter 0 Register (Offset = 0x14)

Bit	Name	Type	Description
[31:24]	TRFC	R/W	Refresh-to-Active/Refresh command period 8'h0: Not used 5'h1: 2 T 5'h2: 3 T ... 5'hff: 256 T
[23:21]	-	-	Reserved
[20:16]	TFAW	R/W	Four bank active times 5'h0: Not used 5'h1: 2 T 5'h2: 3 T ... 5'h1f: 32 T
[15:14]	-	-	Reserved

Bit	Name	Type	Description
[13:8]	TRC	R/W	Active-to-Active command period for the same rank 6'h0: Not used 6'h1: 2 T 6'h2: 3 T ... 6'h3f: 64 T
[7:5]	-	-	Reserved
[4:0]	TRAS	R/W	Active-to-Precharge period This parameter specifies the minimum period of an opened row. 5'h0: Not used 5'h1: 2 T 5'h2: 3 T … 5'h1f: 32 T

6.1.4.8 Timing Parameter 1 Register (Offset = 0x18)

Please note that all parameters are the MClk cycles (T). The "MClk:DRAM clock" mode is "1:2", the parameters should be set to half value of the DRAM clock cycle, except TWR and TWTR.

Table 6-8. Timing Parameter 1 Register (Offset = 0x18)

Bit	Name	Type	Description
31	-	-	Reserved
[30:28]	TWTR	R/W	Internal write-to-read delay READ command after the last write data delay cycle **tWL is odd.** 3'h0: Not used 3'h1: 1.5 T 3'h2: 2.5 T 3'h3: 2.5 T 3'h4: 3.5 T 3'h5: 3.5 T 3'h6: 4.5 T **tWL is even.** 3'h0: Not used 3'h1: 2 T 3'h2: 2 T 3'h3: 3 T 3'h4: 3 T 3'h5: 4 T 3'h6: 4 T 3'h7: 5 T
27	-	-	Reserved
[26:24]	TRTP	R/W	Internal read-to-precharge delay 3'h0: Not used 3'h1: 2 T 3'h2: 3 T ... 3'h7: 8 T

Bit	Name	Type	Description
[23:20]	TWR	R/W	Write recovery time Write-to-precharge The last non-write data cycle at the MCLK rising edge to the precharge command cycle **tWL is odd.** 3'h0: Not used 3'h1: 1.5 T 3'h2: 2.5 T 3'h3: 2.5 T 3'h4: 3.5 T 3'h5: 3.5 T 3'h6: 4.5 T 3'h7: 4.5 T 3'h8: 5.5 T 3'h9: 5.5 T 3'ha: 6.5 T 3'hb: 6.5 T 3'hc: 7.5 T 3'hd: 7.5 T 3'he: 8.5 T 3'hf: 8.5 T **tWL is even.** 3'h0: Not used 3'h1: 2 T 3'h2: 2 T 3'h3: 3 T 3'h4: 3 T 3'h5: 4 T 3'h6: 4 T 3'h7: 5 T 3'h8: 5 T 3'h9: 6 T 3'ha: 6 T 3'hb: 7 T 3'hc: 7 T 3'hd: 8 T 3'he: 8 T 3'hf: 9 T

Bit	Name	Type	Description
[19:16]	TMOD	R/W	Mode register sets the command update delay 4'h0: Not used 4'h1: 2T 4'h2: 3T ... 4'hf: 16T
[15:12]	TMRD	R/W	Cycle time of the load mode register command 4'h0: Not used 4'h1: 2T 4'h2: 3T 4'h3: 4T ... 4'hf: 16T
[11:8]	TRP	R/W	Precharge period 4'h0: Not used 4'h1: 2T 4'h2: 3T ... 4'hf: 16T
[7:4]	TRRD	R/W	Active-to-active command period for different banks 4'h0: Not used 4'h1: 2T 4'h2: 3T ... 4'hf: 16T
[3:0]	TRCD	R/W	Minimum delay between the active and read/write commands 4'h0: Not used 4'h1: 2T 4'h2: 3T ... 4'hf: 16T

6.1.4.9 Timing Parameter 2 Register (Offset = 0x1C)

Please note that all parameters are the MClk cycles (T). The "MClk:DRAM clock" mode is "1:2", the parameters should be set to half value of the DRAM clock cycle.

Table 6-9. Timing Parameter 2 Register (Offset = 0x1C)

Bit	Name	Type	Description
[31:30]	TWtoR_ctrl	R/W	Controller specific Additional delay cycles from the write command to the read command The programmed value compensates the unbalanced trace or I/O between different ranks of DDR3/DDR2 SDRAM to avoid the DQS driving contention. **(tRL - tWL) is odd.** 00: 0T 01: 1T 10: 1T 11: 2T **(tRL - tWL) is even.** 00: 0T 01: 0T 10: 1T 11: 1T
[29:28]	TWtoW_ctrl	R/W	Controller specific Additional delay cycles from the write command to the write command 00: 0T 01: 1T 10: 2T 11: 3T
[27:26]	TRtoR_ctrl	R/W	Controller specific Additional delay cycles from the read command to the read command 00: 0T 01: 1T 10: 2T 11: 3T

Bit	Name	Type	Description
[25:24]	TRtoW_ctrl	R/W	Controller specific Additional delay cycles from the read command to the write command The DQS will be driven by different devices in the read and write operations. The programmed value will compensate the round-trip delay between the chip and DDR3/DDR2 SDRAM to avoid the DQS driving contention. 00: 0T 01: 1T 10: 2T 11: 3T
[23:16]	-	-	Reserved
[15:8]	TXSR	R/W	Exit from the self-refresh mode to a command period This should be programmed to the maximum of tXSNR and tXSRD . 8'h0: Not used 8'h1: 1 X 8T 8'h2: 2 X 8T ... 8'hff: 255 X 8T Note: There are four additional MClk guard times to compensate for the internal exiting self-refresh command to the DDR PHY I/O delay.
[7:0]	TREFI	R/W	Average periodic refresh interval One refresh command should be issued if the refresh counter equals to the refresh interval. Please note that bit 7 is only for selection between x32T and x8T. Bit 7 = 0: 0_000_0000: Not used 0_000_0001: 1X32T 0_000_0010: 2X32T ... 0_111_1111: 127X32T Bit 7 = 1: 1_000_0000: 0x8T 1_000_0001: 1X8T 1_000_0010: 2X8T ... 1_111_1111: 127X8T

6.1.4.10 Command and Data Block Control Register (Offset = 0x20)

The bit 26, DQS_B (EMRS), of the register Offset = 0x08 and bit 3 of the register Offset = 0x020 should be set to the same value in the DDR2 mode.

Table 6-10. Command and Data Block Control Register (Offset = 0x20)

Bit	Name	Type	Description
[31:26]	-	-	Reserved
25	fldo_byte1	RO	Byte 1 of the DLL lock flag indicates that DLL is at the lock state. 1: DLL is at the lock state. 0: DLL is not at the lock state.
24	fldo_byte0	RO	Byte 0 of the DLL lock flag indicates that DLL is at the lock state. 1: DLL is at the lock state. 0: DLL is not at the lock state.
[23:17]	-	-	Reserved
16	fldo_cmd	RO	DDR3 PHY only 1: Command block of DLL is at the lock state. 0: Command block of DLL is not at the lock state.
15	auto_io_deep_pdn	RO	Automatic control of the I/O output buffer at the power-down state 1'b1: Enable 1'b0: Disable When auto_io_deep_pdn is enabled, cmdaddroen and dqie will be disabled at the automatic power-down state; and clkoen, cmdaddroen, dqie, and odtoen will be disabled at the self-refresh state.
14	self_bias_0	R/W	Enable the self-bias generation for the DDR3 low 16bit data block 1: Bias from inside (VREF is floating.) 0: Bias from outside (VREF is connected to bias.)
13	phy_con_update	R/W	DDR PHY digital DLL update control code 1: Slave DLL continuously updates the control code despite of the status of the "update_en" signal. 0: Slave DLL updates the control code when the "update_en" signal is asserted.
12	clkphaseshift	R/W	DDR PHY output clock for the DDR SDRAM phase 0: Clock position is at the 1/2 clock cycle phase of the DDR command. 1: Clock position is at the 3/4 clock cycle phase of the DDR command.

Bit	Name	Type	Description
11	dqie	R/W	DQ and DQS receivers enable 1: Enable 0: Disable Note: In the automatic power-down mode, the dqie bit will be controlled by hardware if auto_ioctrl_pdn is set to "1".
10	clkoen	R/W	The output controls the DDR PHY CK, CKB, and ODT drivers. 1: Enable 0: Disable
9	cmdaddroen	R/W	The output controls the DDR PHY ADDR, CK, CKB, BA, RAS, CAS, WE, and CS drivers. 1: Enable 0: Disable Note: In the automatic power-down mode, the cmdaddroen bit will be controlled by hardware if auto_ioctrl_pdn is set to "1".
8	odtoen	R/W	Enable the output drivers of ODT[1:0] 1: Enable 0: Disable Note: In the automatic power-down mode, the cmdaddroen bit will be controlled by hardware if auto_ioctrl_pdn is set to "1".
7	auto_ioctrl_pdn	R/W	Automatic control of the I/O output buffer at the power-down stage 1: Enable 0: Disable When enabling the automatic control, the odtoen, cmdaddroen, and dqie bits will be disabled at the automatic power-down and self-refresh states. The clkoen bit is disabled at the self-refresh state.
6	IO18V	R/W	For DDR3 PHY only 1: I/O is 1.8V. 0: I/O is 1.5V.
5	dqslowfen	R/W	Enable the DQS bypass DLL This bit is used for the low-frequency operation to control the DDR PHY bypass DLL.
4	dmyodten	R/W	On-Die Termination (ODT) enable for the DDR PHY DUMMY pad of the data block
3	sio	R/W	This bit should be compatible with the DQS_B enable bit in the extended mode register. 0: DQS is in the differential mode. 1: DQS is in the single-ended mode.

Bit	Name	Type	Description
[2:0]	odtmd[2:0]	R/W	Set the ODT value of the DDR3 Combo PHY 111: ODT = 20Ω 110: ODT = 24Ω 101: ODT = 30Ω 100: ODT = 40Ω 011: ODT = 40Ω 010: ODT = 60Ω 001: ODT = 120Ω 000: ODT is disabled.

6.1.4.11 Read Path DLL Delay Tuning Register (Offset = 0x24)

dllsel is used for adjusting the read DQ and read DQS. The read DQS should be in the center of the DQ data eye for more timing margin.

Table 6-11. Read Path DLL Delay Tuning Register (Offset = 0x24)

Bit	Name	Type	Description
[31:7]	-	-	Reserved
[6:4]	dllsel_byte1	R/W	Delay value control to add the delay into the read DQS or read DQ Please refer to the register Offset 0x6C for details.
3	-	-	Reserved
[2:0]	dllsel_byte0	R/W	Delay value control to add the delay into the read DQS or read DQ Please refer to the register Offset 0x6C for details.

6.1.4.12 COMPBLK Control Register (Offset = 0x28)

Table 6-12. COMPBLK Control Register (Offset = 0x28)

Bit	Name	Type	Description
[31:25]	-	-	Reserved
[24:19]	DON	RO	Value for the COMPBLK pull-down NMOS impedance
[18:13]	DOP	RO	Value for the COMPBLK pull-up PMOS impedance
[12:7]	DIN	R/W	Preset value for the COMPBLK pull-down NMOS impedance
[6:1]	DIP	R/W	Preset value for the COMPBLK pull-up PMOS impedance

Bit	Name	Type	Description
0	COMP_SEL	R/W	Compensation method selection 1: Enable the COMPBLK calibration before initiating DDRx SDRAM 0: Use the preset values of DIP and DIN

6.1.4.13 Automatic Power-down/Self-refresh Control Register (Offset = 0x2C)

Table 6-13. Automatic Power-down/Self-refresh Control Register (Offset = 0x2C)

Bit	Name	Type	Description
[31:29]	-	-	Reserved
28	auto_srf_en	R/W	Automatic self-refresh mode control 1: Enable the automatic self-refresh 0: Disable the automatic self-refresh
[27:16]	auto_srf_timer		Automatic self-refresh timer When the command queue is empty, the automatic self-refresh timer will start counting down. When the timer reaches zero, the self-refresh request will be triggered and DDRx SDRAM will enter the self-refresh mode. 12'h000: Not used 12'h001: 1 x 4 = 4 T 12'h002: 2 x 4 = 8 T … 12'hfff: 4095 x 4 = 16380 T
[15:13]	-	-	Reserved
12	auto_pdn_en	R/W	Automatic power-down mode control 1: Enable the automatic power-down mode 0: Disable the automatic power-down mode
[11:0]	Auto_pdn_timer	R/W	Automatic power-down timer When the command queue is empty, the automatic power-down counter will start counting down. When the counter reaches zero, CKE will be automatically pulled low. 12'h000: Not used 12'h001: 1 x 4T = 4T 12'h002: 2 x 4T = 8T … 12'hfff: 4095 x 4T = 16380T

6.1.4.14 Channel Arbitration Setup Register (Offset = 0x30)

Table 6-14 lists and describes the parameter setup for the channel arbitration. There are two arbitration modes:

- Two-level round-robin with the channel grant count
- Read-Write group with grant_count_group and the channel grant count

Table 6-14. Channel Arbitration Setup Register (Offset = 0x30)

Bit	Name	Type	Description
31	RW_Grp_EN	-	Enable the read-write group arbitration Set this bit to '1' to enable the read-write group arbitration. Otherwise, the normal two-level round-robin arbitration will be used. 1: Read-write group arbitration is enabled. 0: Read-Write group arbitration is disabled.
30	Indep_RW_EN	R/W	Independent R/W control 1'b1: Independent R/W is enabled. 1'b0: Independent R/W is disabled.
29	-	-	Reserved
[28:24]	Grant_Cnt_Group	R/W	For the read-write group arbitration, this field sets the initial value of the group grant window count. This field will not be used if the read-write group arbitration is disabled. 5'h0: Not used 5'h1: Issue a maximum of 1 command when each group is granted. 5'h2: Issue a maximum of 2 commands when each group is granted. … 5'h1f: Issue a maximum of 31 commands when each group is granted.
[23:10]	-	-	Reserved
9	BstOriArb	R/W	Burst oriented arbitration enable bit (Only for the AHB slave ports) When this bit is set, the burst transaction will not be broken during arbitration. '1' indicates that the burst oriented arbitration of channel 1 is enabled. '0' indicates that the burst oriented arbitration of channel 1 is disabled.
[8:4]	-	-	Reserved

Bit	Name	Type	Description
[3:0]	Ch_hi_prior	R/W	All channels are divided into two priority levels: The high-priority level and low-priority level. These bits indicate which channels belong to the high-priority level. Setting each bit to '1'b1' indicates that the corresponding channel belong to the high-priority level. Bit 0 = 1 indicates that channel 0 belongs to the high-priority group. Bit 0 = 0 indicates that channel 0 belongs to the low-priority group. Bit 1 = 1 indicates that channel 1 belongs to the high-priority group. Bit 1 = 0 indicates that channel 1 belongs to the low-priority group. Bit 2 = 1 indicates that channel 2 belongs to the high-priority group. Bit 2 = 0 indicates that channel 2 belongs to the low-priority group. Bit 3 = 1 indicates that channel 3 belongs to the high-priority group. Bit 3 = 0 indicates that channel 3 belongs to the low-priority group.

6.1.4.15 Channel Arbiter Grant Count Register - A (Offset = 0x34)

Each channel has its own grant window counter. The value set in this register will be loaded into the grant window counters of channel 0, channel 1, channel 2, and channel 3, when the granted channel is changed by arbiter. Once the granted channel pops a command to the memory controller, the grant window counter of this channel will be decreased by one. If the grant window counter of the current grated channel becomes '0', the arbiter will switch the grant to another channel. The grant window counter of each channel defines the maximum service command counts allowed for each channel when that channel is granted by the arbiter. Please note that a command indicates a DDR3 WRAP8 R/W command.

Table 6-15. Channel Arbiter Grant Count Register - A (Offset = 0x34)

Bit	Name	Type	Description
[31:29]	-	R/W	Reserved
[28:24]	ARB_CNT3	R/W	Once channel 3 is granted, the maximum allowed commands will be issued. 0: 2 commands 1: 3 commands 2: 4 commands … 30: 32 commands 31: 1 command
[23:21]	-	R/W	Reserved

Bit	Name	Type	Description
[20:16]	ARB_CNT2	R/W	Once channel 2 is granted, the maximum allowed commands will be issued. 0: 2 commands 1: 3 commands 2: 4 commands … 30: 32 commands 31: 1 command
[15:13]	-	R/W	Reserved
[12:8]	ARB_CNT1	R/W	Once channel 1 is granted, the maximum allowed commands will be issued. 0: 2 commands 1: 3 commands 2: 4 commands … 30: 32 commands 31: 1 command
[7:5]	-	R/W	Reserved
[4:0]	ARB_CNT0	R/W	Once channel 0 is granted, the maximum allowed commands will be issued. 0: 2 commands 1: 3 commands 2: 4 commands … 30: 32 commands 31: 1 command

6.1.4.16 Write/Read Data Timing Control Register (Offset = 0x3C)

tphy_wrlat specifies the number of MClk clock cycles between a write command sent on the PHY control interface and when the dfi_wrdata_en (Dqsoe) signal is asserted. tphy_wrdata specifies the number of MClk clock cycles between when the dfi_wrdata_en (Dqsoe) signal is asserted and when the associated write data are driven on the dfi_wrdata (Ddr_wdata) signal.

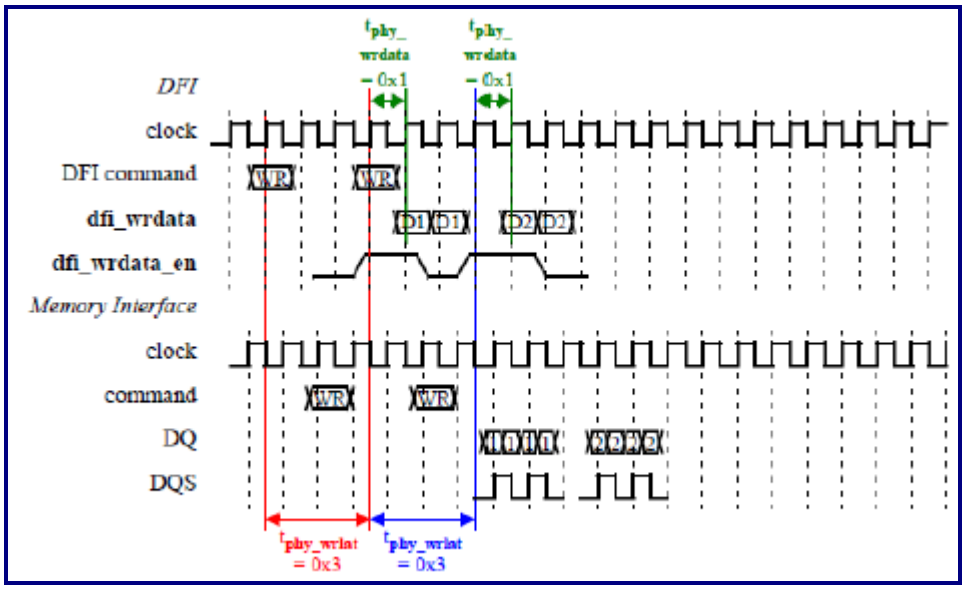


Figure 6-1. tphy_wrlat and tphy_wrdata Timing in DFI Specification 2.1

The dfi_rddata_en (rdcmd) signal must be asserted for the trddata_en cycles after the assertion of a read command on the PHY control interface and remains valid for a duration of the contiguous read data expected on the dfi_rddata (Ddr_rdata) bus.

Read data are expected to be received at the controller within tphy_rdlat cycles after the dfi_rddata_en signal is asserted.

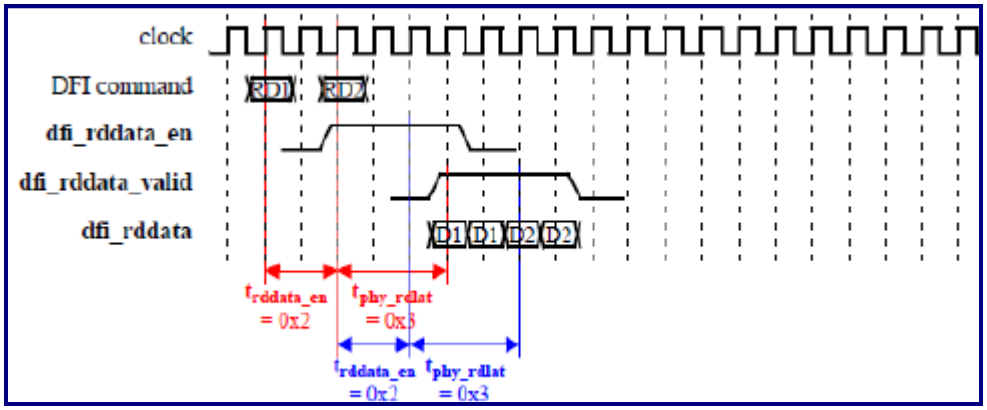


Figure 6-2. trddata_en and tphy_rdlat in DFI Specification 2.1

Table 6-16. Write/Read Data Timing Control Register (Offset = 0x3C)

Bit	Name	Type	Description
[31:24]	-	R/W	Reserved
[23:20]	tphy_rdlat	R/W	Read data are expected to be received at the controller within tphy_rdlat cycles after the rdcmd signal is asserted. Valid values are between 2 and 11. 2: ddr_rdata is expected to receive 2 MClk cycles after rdcmd is asserted. ... 11: ddr_rdata is expected to receive 11 MClk cycles after rdcmd is asserted.
[19:16]	trddata_en	R/W	The dfi_rddata_en (rdcmd) signal must be asserted for trddata_en cycles after the assertion of a read command on the PHY control interface and remained valid for a duration of contiguous read data expected on the dfi_rddata (Ddr_rdata) bus. Valid values are between 0 and 10. 1: rdcmd asserts 1 MClk cycle after the read command is asserted. 2: rdcmd asserts 2 MClk cycles after the read command is asserted. … 10: rdcmd asserts 10 MClk cycles after the read command is asserted. Others: Reserved
[15:6]	-	R/W	Reserved

Bit	Name	Type	Description
[5:4]	tphy_wrdata	R/W	tphy_wrdata specifies the number of MClk clock cycles between when the dfi_wrdata_en(Dqsoe) signal is asserted and when the associated write data are driven on the ddr_wdata signal. 0: ddr_wdata is asserted at the same time when Dqsoe asserts. 1: ddr_wdata asserts 1 MClk cycle after Dqsoe is asserted. 2: ddr_wdata asserts 2 MClk cycles after Dqsoe is asserted. 3: ddr_wdata asserts 3 MClk cycles after Dqsoe is asserted.
[3:0]	tphy_wrlat	R/W	tphy_wrlat specifies the number of MClk clock cycles between when a write command is sent on the DFI control interface and when the dfi_wrdata_en(Dqsoe) signal is asserted. Valid values are between 0 and 9. 0: dqsoe is asserted at the same time with the write command asserts. 1: dqsoe asserts 1 MClk cycle after the write command is asserted. 2: dqsoe asserts 2 MClk cycles after the write command is asserted. … 9: dqsoe asserts 10 MClk cycles after the write command is asserted. Others: Reserved

tWL is odd, tphy_wrlat = ((tWL - 1)/2) -1

tWL is even, tphy_wrlat = (tWL/2) -1

tphy_wrdata = 1

tRL is odd, trddata_en = ((tRL - 1)/2) -1

tRL is even, trddata_en = (tRL/2) -1

tphy_rdlat = GDS + 4

For example,

GDS = 0, tWL = 7, tRL = 9, tphy_wrlat = 4'h2, tphy_wrdata = 2'h1, trddata_en = 4'h3, tphy_rdlat = 4'h4

GDS = 1, tWL = 7, tRL = 9, tphy_wrlat = 4'h2, tphy_wrdata = 2'h1, trddata_en = 4'h3, tphy_rdlat = 4'h5

GDS = 2, tWL = 7, tRL = 9, tphy_wrlat = 4'h2, tphy_wrdata = 2'h1, trddata_en = 4'h3, tphy_rdlat = 4'h6

GDS = 3, tWL = 7, tRL = 9, tphy_wrlat = 4'h2, tphy_wrdata = 2'h1, trddata_en = 4'h3, tphy_rdlat = 4'h7

GDS = 4, tWL = 7, tRL = 9, tphy_wrlat = 4'h2, tphy_wrdata = 2'h1, trddata_en = 4'h3, tphy_rdlat = 4'h8

GDS = 5, tWL = 7, tRL = 9, tphy_wrlat = 4'h2, tphy_wrdata = 2'h1, trddata_en = 4'h3, tphy_rdlat = 4'h9

GDS = 6, tWL = 7, tRL = 9, tphy_wrlat = 4'h2, tphy_wrdata = 2'h1, trddata_en = 4'h3, tphy_rdlat = 4'ha

GDS = 7, tWL = 7, tRL = 9, tphy_wrlat = 4'h2, tphy_wrdata = 2'h1, trddata_en = 4'h3, tphy_rdlat = 4'hb

6.1.4.17 Command Flush Control Register (Offset = 0x40)

Table 6-17. Command Flush Control Register (Offset = 0x40)

Bit	Name	Type	Description
[31:17]	-	-	Reserved
[16]	debug_int_en	R/W	Interrupt enable of the debug function
[15:12]	-	-	Reserved
[11:8]	flush_int_en	R/W	Interrupt enable of the command flushing When the command flushing is completed, the controller will trigger an interrupt if the corresponding bit of this field is set to '1'. Bit 8: Channel 0 Bit 9: Channel 1 Bit 10: Channel 2 Bit 11: Channel 3
[7:4]	-	-	Reserved
[3:0]	cmd_flush_en	R/W	Enable of the command flushing Set the corresponding bit of this field to '1' to start the command flushing Bit 0: Channel 0 Bit 1: Channel 1 Bit 2: Channel 2 Bit 3: Channel 3 When the command flushing is completed, the corresponding bit of this field will be automatically cleared to '0' by the controller.

6.1.4.18 Command Flush Status Register (Offset = 0x44)

Table 6-18. Command Flush Status Register (Offset = 0x44)

Bit	Name	Type	Description
[31:9]	-	-	Reserved
8	debug_hit	R/W1C	Status of the debug function
[7:4]	-	-	Reserved

Bit	Name	Type	Description
[3:0]	flush_done	R/W1C	Status of the command flushing This field records the status of the command flushing for each channel. When a channel terminates the command flushing, the corresponding bit of this field will be set to '1' by the controller. Bit 0: Channel 0 Bit 1: Channel 1 Bit 2: Channel 2 Bit 3: Channel 3 This field should be cleared by writing '1' to the corresponding bit.

6.1.4.19 AHB SPLIT Control Register (Offset = 0x48)

If the AHB channels are configured with the supported SPLIT responses, the SPLIT function can be disabled by setting the corresponding bit in this register.

Table 6-19. AHB SPLIT Control Register1 (Offset = 0x48)

Bit	Name	Type	Description
[31:12]	-	-	Reserved
[11:10]	CH1_hprot_sel	-	01: Always set the channel to bufferable regardless of the value in the HPROT input port 10: Always set the channel to non-bufferable regardless of the value in the HPROT input port Others: It depends on the value in the HPROT input port.
[9:8]	Reserved	R/W	-
7~2	Reserved	R/W	-
1	CH1_split_disable	R/W	This bit works only when channel 1 is enabled and configured as an AHB channel with the SPLIT response. Set this bit to '1' to disable the SPLIT response of channel 1
0	Reserved	R/W	-

6.1.4.20 Update Control Register (Offset = 0x4C)

This register is for the DDR3 PHY DLL update and write-leveling calibration register.

Table 6-20. Update Control Register (Offset = 0x4C)

Bit	Name	Type	Description
[31:18]	-	-	Reserved
[17:16]	ZQCS and ZQCL update	R/W	2'b01: Automatically enable the ZQCS command after exiting the self-refresh state 2'b10: Automatically enable the ZQCL command after exiting the self-refresh state Others: Not used and no ZQCS or ZQCL command after exiting the self-refresh state If the ZQCS/ZQCL update is enabled, the ZQCS/ZQCL commands will be issued after tXS (tRFC +10 ns) is satisfied.
[15:8]	tWLEVEL_UP	R/W	Write-leveling calibration timing interval 0: Disable the write-leveling calibration interval 1: 1X 1024 MClk cycles 2: 2X 1024 MClk cycles … 255: 255X 1024 MClk cycles Note: This value should be reasonably set; otherwise, it may impact the performance.
[7:0]	tDLL_UP	R/W	Digital DLL update delay control code timing interval 0: Disable the digital DLL update 1: 1X 1024 MClk cycles 2: 2X 1024 MClk cycles … 255: 255X 1024 MClk cycles Note: This value should be reasonably set; otherwise, it may impact the performance.

6.1.4.21 User Define Register (Offset = 0x5C)

Table 6-21. User Define Register (Offset = 0x5C)

Bit	Name	Type	Description
[31:3]	-	-	Reserved

Bit	Name	Type	Description
[2:0]	RONMD	R/W	The driving value setting of DDR3 PHY 000: 240Ω 001: 120Ω 010: 80Ω 011: 60Ω 100: 60Ω 101: 48Ω 110: 40Ω 111: 34Ω (Default)

6.1.4.22 Write-leveling Control Register (Offset = 0x60) (Only for DDR3 Mode)

The write-leveling control register is used to adjust the skew between the command path and DQ/DQS path by using the Fly-By DDR3 SDRAM configuration. Users can use the hardware scan by setting each byte of wlevel_byten_hw_disable to 1'b0 or use the software scan to fill the PDL_set registers (Offset 0x64 and Offset 0x68). Both tDQSL and tWLO should be set to the half value of the DRAM timing parameters according to the "MClk:DRAM clock = 1:2" mode.

Table 6-22. Write-leveling Control Register (Offset = 0x60) (Only for DDR3 Mode)

Bit	Name	Type	Description
[31:26]	-	-	Reserved
25	wlevel_byte1_pass	RO	Byte 1 of the write-leveling status register 1: Write-leveling passed 0: Write-leveling failed or write-leveling hardware disabled
24	wlevel_byte0_pass	RO	Byte 0 of the write-leveling status register 1: Write-leveling passed 0: Write-leveling failed or write-leveling hardware disabled
[23:18]	-	-	Reserved
17	wlevel_byte1_hw_disable	R/W	Byte 1 of the write-leveling hardware disable 1: Write-leveling by the software scan (Hardware disabled) 0: Write-leveling by hardware

Bit	Name	Type	Description
16	wlevel_byte0_hw_disable	R/W	Byte 0 of the write-leveling hardware disable 1: Write-leveling by the software scan (Hardware disabled) 0: Write-leveling by hardware
[15:8]	-	-	Reserved
[7:0]	tWLO	R/W	Write-leveling DQS enable to latch the DQ data delay time 8'h1: 1T … 8'hff: 255T Recommanded value: 10T

6.1.4.23 Write-leveling Byte 3 ~ Byte 0 Control Register (Offset = 0x68) (Only for DDR3 Mode)

This register adjusts the multiphase DLL of DDR23 Combo PHY to compensate the skew between the Fly-By command and DQ (DQS). Please refer to Table 6-23 for more details. If the hardware tuning write-leveling is enabled (Offset = 0x60), the value in this register will reflect the hardware tuning result rather than the software programmed value.

Table 6-23. Write-leveling Byte 3 ~ Byte 0 Control Register (Offset = 0x68) (Only for DDR3 Mode)

Bit	Name	Type	Description
[31:15]	-	-	Reserved
[14:8]	PDL_set_byte1	R/W	Programmable DLL delay for byte 1 wlevel_byte1_hw_disable = 0: Hardware tuning value wlevel_byte1_hw_disable = 1: Software programmed value
7	-	-	Reserved
[6:0]	PDL_set_byte0	R/W	Programmable DLL delay for byte 0 wlevel_byte0_hw_disable = 0: Hardware tuning value wlevel_byte0_hw_disable = 1: Software programmed value

6.1.4.24 MISC Control Register 1 (Offset = 0x6C) (Only for DDR3 Mode)

Table 6-24. MISC Control Register 1 (Offset = 0x6C) (Only for DDR3 Mode)

Bit	Name	Type	Description
[31:6]	-	-	Reserved
5	dqs_sel_wr_byte1	R/W	Add a delay in DQ or DQS, the delay value is controlled by wrdll_sel_byte1 (Offset 0x78). 1: Write DQ is fixed and only delays DQS. 0: Write DQS is fixed and only delays DQ.
4	dqs_sel_byte1	R/W	Add a delay in DQ or DQS, the delay value is controlled by dllsel_byte1 (Offset 0x24). 1: Read DQ is fixed and only delays DQS. 0: Read DQS is fixed and only delays DQ.
[3:2]	-	-	Reserved
1	dqs_sel_wr_byte0	R/W	Add a delay in DQ or DQS, the delay value is controlled by wrdll_sel_byte0 (Offset 0x78). 1: Write DQ is fixed and only delays DQS. 0: Write DQS is fixed and only delays DQ.
0	dqs_sel_byte0	R/W	Add a delay in DQ or DQS, the delay value is controlled by dllsel_byte0 (Offset 0x24). 1: Read DQ is fixed and only delays DQS. 0: Read DQS is fixed and only delays DQ.

6.1.4.25 Read-leveling Control Register (Offset = 0x70) (Only for DDR3 Mode)

The read-leveling control register adjusts the read DQS filtering window. Users can use the hardware scan by setting each byte of rlevel_sel_byten_hw_disable to 1'b0 or use the software scan to fill the msdly register (Offset = 0x74).

Table 6-25. Read-leveling Control Register (Offset = 0x70) (Only for DDR3 Mode)

Bit	Name	Type	Description
[31:26]	-	-	Reserved
25	rlevel_byte1_pass	RO	Byte 1 of the read-leveling status 1: Read-leveling passed 0: Read-leveling failed or read-leveling hardware disabled

Bit	Name	Type	Description
24	rlevel_byte0_pass	RO	Byte 1 of the read-leveling status 1: Read-leveling passed 0: Read-leveling failed or read-leveling hardware disabled
[23:2]	-	-	Reserved
1	rlevel_byte1_hw_disable	R/W	Byte 1 of the read-leveling hardware disable 1: Read-leveling by software (Hardware disabled) 0: Read-leveling by hardware
0	rlevel_byte0_hw_disable	R/W	Byte 0 of the read-leveling hardware disable 1: Read-leveling by software (Hardware disabled) 0: Read-leveling by hardware

6.1.4.26 msdly Byte Control Register (Offset = 0x74)

The read DQS control register is treated as a clock in the read path. This register adjusts the read DQS filtering window at DDR PHY to filter out the unwanted pulse or glitch of read DQS. If the hardware tuning read-leveling is enabled (Offset = 0x70), the value in this register will reflect the hardware tuning result rather than the software programmed value.

Table 6-26. msdly Byte Control Register (Offset = 0x74)

Bit	Name	Type	Description
[31:8]	-	-	Reserved
[7:4]	msdly_byte1	R/W	Programmable read DQS filtering window for byte 1 rlevel_byte1_hw_disable = 0: Hardware tuning value rlevel_byte1_hw_disable = 1: Software programmed value
[3:0]	msdly_byte0	R/W	Programmable read DQS filtering window for byte 0 rlevel_byte0_hw_disable = 0: Hardware tuning value rlevel_byte0_hw_disable = 1: Software programmed value

6.1.4.27 wrdllsel Control Register (Offset = 0x78)

Table 6-27. wrdllsel Control Register (Offset = 0x78)

Bit	Name	Type	Description
[31:7]	-	-	Reserved
[6:4]	wrdll_sel_byte1	R/W	Delay value control to add a delay in write DQS or DQ Please refer to offset 0x6C for details.
3	-	-	Reserved
[2:0]	wrdll_sel_byte0	R/W	Delay value control to add a delay in write DQS or DQ Please refer to offset 0x6C for details.

6.1.4.28 Traffic Monitor Clock Cycle Register (Offset = 0x7C)

Table 6-28. Traffic Monitor Clock Cycle Register (Offset = 0x7C)

Bit	Name	Type	Description
[31:0]	TM_clk_cycle_reg	R/W	This register works only when the traffic monitor is enabled. The value of this register will be decreased by 1 for every clock cycle of MClk. When the value of this register is not zero, the traffic monitor will count the issued command number of each channel.

6.1.4.29 Command Count Register for Channel 0 (Offset = 0x80)

Table 6-29. Command Count Register for Channel 0 (Offset = 0x80)

Bit	Name	Type	Description
[31:0]	cmd_cnt_reg0	RO	This register works only when the traffic monitor and channel 0 are enabled. This register records the number of the issued commands from channel 0, and this value will be automatically cleared when register, TM_clk_cycle_reg, is written by users.

6.1.4.30 Command Count Register for Channel 1 (Offset = 0x84)

Table 6-30. Command Count Register for Channel 1 (Offset = 0x84)

Bit	Name	Type	Description
[31:0]	cmd_cnt_reg1	RO	This register works only when the traffic monitor and channel 1 are enabled. This register records the number of the issued commands from channel 1, and this value will be automatically cleared when register, TM_clk_cycle_reg, is written by users.

6.1.4.31 Command Count Register for Channel 2 (Offset = 0x88)

Table 6-31. Command Count Register for Channel 2 (Offset = 0x88)

Bit	Name	Type	Description
[31:0]	cmd_cnt_reg2	RO	This register works only when the traffic monitor and channel 2 are enabled. This register records the number of the issued commands from channel 2, and this value will be automatically cleared when register, TM_clk_cycle_reg, is written by users.

6.1.4.32 Command Count Register for Channel 3 (Offset = 0x8C)

Table 6-32. Command Count Register for Channel 3 (Offset = 0x8C)

Bit	Name	Type	Description
[31:0]	cmd_cnt_reg3	RO	This register works only when the traffic monitor and channel 3 are enabled. This register records the number of the issued commands from channel 3, and this value will be automatically cleared when register, TM_clk_cycle_reg, is written by users.

6.1.4.33 AHB INCR Read Prefetch Length 1 (Offset = 0xA0)

The controller provides two mechanisms to prefetch the read data for the AHB unspecified-length (INCR) read bursts, the limited prefetching and unlimited prefetching.

For the limited prefetching mechanism, users should define a value used as "total number" of the read commands. The controller generates the DDR read commands when receiving an AHB INCR read burst. For example, if the user-defined value is N, the controller will generate three N+1 DDR read commands to prefetch the read data for any AHB INCR bursts.

For the unlimited prefetching, users should define a value used as "speed factor" of the read commands. The controller will generate the read commands until it receives an AHB early termination condition. The user-defined value determines the maximum number of the prefetch read commands in pipeline. For example, if the user-defined value is N, the controller will first generate N+1 DDR read commands and will then suspend the command generation until the on-going command count reaches N+1. When the data belong to the first read command and arrive read FIFO, the controller will generate the fourth read command if the on-going command count that is less than N+1. This mechanism will be repeatedly performed until the early termination of AHB occurs.

Table 6-33. AHB INCR Read Prefetch Length 1 (Offset = 0xA0)

Bit	Name	Type	Description
[31:16]	-	-	Reserved
15	CH1_limited_pref	R/W	Prefetch mechanism of channel 1 1: Limited prefetching 0: Unlimited prefetching
[14:13]	-	-	Reserved
[12:8]	CH1_pref_value	R/W	User-defined value of channel 1 for the INCR read prefetching 0: One command 1: Two commands 2: Three commands …
[7:0]	-	-	Reserved

6.1.4.34 Initialization of Waiting Cycle Count 1 (Offset = 0xA8)

Table 6-34. Initialization of Waiting Cycle Count 1 (Offset = 0xA8)

Bit	Name	Type	Description
[31:20]	-	-	Reserved
[19:0]	wait_cycle_200us	R/W	After powering up the ramp of SDRAM, a waiting time of 200μs is required before the RESET signal becomes inactive. This register defines the waiting clock cycles of the controller to satisfy the requirement of 200μs.

6.1.4.35 Initialization of Waiting Cycle Count 2 (Offset = 0xAC)

Table 6-35. Initialization of Waiting Cycle Count 2 (Offset = 0xAC)

Bit	Name	Type	Description
[31:20]	-	-	Reserved
[19:0]	wait_cycle_500us	R/W	For DDR3 SDRAM, a waiting time of 500µs is required after the RESET signal becomes inactive and before CKE becomes active. This register defines the waiting clock cycles of the controller to satisfy the requirement of 500µs.

6.1.4.36 QoS Control Register (Offset = 0xB0)

Table 6-36. QoS Control Register (Offset = 0xB0)

Bit	Name	Type	Description
[31:4]	-	-	Reserved
3	QoS_Indep_RW_en	R/W	QoS independent R/W enable If this bit is set to '1', the setting of QoS_CmdCnt in 0xB4 and 0xB8 will be the limitation only for the read command. If this bit is set to '0', the setting of QoS CmdCnt in 0xB4 and 0xB8 will be the limitation for both the read and write commands.
[2:1]	QoS_PeriodMode	R/W	QoS period mode 00: 512 clock cycles 01: 1024 clock cycles 0: 2048 clock cycles 11: 4096 clock cycles
0	QoS_En	R/W	QoS enable

6.1.4.37 QoS Command Count Register A (Offset = 0xB4)

If QoS_Indep_RW_en is set to '1', Register Offset 0xB4 will be used for the read command.

If QoS_Indep_RW_en is set to '0', Register Offset 0xB4 will be used for both the read and write commands.

Table 6-37. QoS Command Control Register A (Offset = 0xB4)

Bit	Name	Type	Description
[31:24]	QoS_Ch3CmdCnt	R/W	Command Count Limitation for Channel 3 8'h00: No limit 8'h01: 1 x 8 commands 8'h02: 2 x 8 commands ... 8'hff: 255 x 8 commands
[23:16]	QoS_Ch2CmdCnt	R/W	Command Count Limitation for Channel 2 8'h00: No limit 8'h01: 1 x 8 commands 8'h02: 2 x 8 commands ... 8'hff: 255 x 8 commands
[15:8]	QoS_Ch1CmdCnt	R/W	Command Count Limitation for Channel 1 8'h00: No limit 8'h01: 1 x 8 commands 8'h02: 2 x 8 commands ... 8'hff: 255 x 8 commands
[7:0]	QoS_Ch0CmdCnt	R/W	Command Count Limitation for Channel 0 8'h00: No limit 8'h01: 1 x 8 commands 8'h02: 2 x 8 commands ... 8'hff: 255 x 8 commands

6.1.4.38 QoS Command Count Register C (Offset = 0xBC)

If QoS_Indep_RW_en is set to '1', register Offset 0xBC will be used for the write command.

If QoS_Indep_RW_en is set to '0', register Offset 0xBC will be meaningless.

Table 6-38. QoS Command Control Register C (Offset = 0xBC)

Bit	Name	Type	Description
[31:24]	QoS_Ch3CmdCnt_w	R/W	Write Command Count Limitation for Channel 3 8'h00: No limit 8'h01: 1 x 8 commands 8'h02: 2 x 8 commands ... 8'hff: 255 x 8 commands
[23:16]	QoS_Ch2CmdCnt_w	R/W	Write Command Count Limitation for Channel 2 8'h00: No limit 8'h01: 1 x 8 commands 8'h02: 2 x 8 commands ... 8'hff: 255 x 8 commands
[15:8]	QoS_Ch1CmdCnt_w	R/W	Write Command Count Limitation for Channel 1 8'h00: No limit 8'h01: 1 x 8 commands 8'h02: 2 x 8 commands ... 8'hff: 255 x 8 commands
[7:0]	QoS_Ch0CmdCnt_w	R/W	Write Command Count Limitation for Channel 0 8'h00: No limit 8'h01: 1 x 8 commands 8'h02: 2 x 8 commands ... 8'hff: 255 x 8 commands

6.1.4.39 Channel Arbitration Setup Register B (Offset = 0xC4)

If register Offset 0x30 bit[30] "Indep_RW_EN" = '1', register Offset 0xC4 will be used for the write command. If register Offset 0x30 bit[30] "Indep_RW_EN" = '0', register Offset 0xC4 will be meaningless.

Table 6-39. Channel Arbitration Setup Register B (Offset = 0xC4)

Bit	Name	Type	Description
[31:29]	-	-	Reserved
[28:24]	group_grant_count_w	R/W	This field will not be used if the Read-Write group arbitration is disabled. 5'h0: Issue a maximum of 1 command when the write group is granted. 5'h1: Issue a maximum of 2 commands when the write group is granted. 5'h2 = Issue a maximum of 3 commands when the write group is granted. … 5'h1f = Issue a maximum of 32 commands when the write group is granted.
[23:12]	-	-	Reserved
[11:8]	BstOriArb_w	R/W	Burst oriented arbitration enable bit When these bits are set, the burst transaction will not be broken during arbitration. Bit 8 = 1 indicates that the burst oriented arbitration of channel 0 is enabled for the write commands. Bit 8 = 0 indicates that the burst oriented arbitration of channel 0 is disabled for the write commands. Bit 9 = 1 indicates that the burst oriented arbitration of channel 1 is enabled for the write commands. Bit 9 = 0 indicates that the burst oriented arbitration of channel 1 is disabled for the write commands. Bit 10 = 1 indicates that the burst oriented arbitration of channel 2 is enabled for the write commands. Bit 10 = 0 indicates that the burst oriented arbitration of channel 2 is disabled for the write commands. Bit 11 = 1 indicates that the burst oriented arbitration of channel 3 is enabled for the write commands. Bit 11 = 0 indicates that the burst oriented arbitration of channel 3 is disabled for write commands.
[7:4]	-	-	reserved

Bit	Name	Type	Description
[3:0]	Ch_hi_prior_w	R/W	All channels are divided into two priority levels: High-priority level and low-priority level. These bits indicate which channels belong to the high-priority level. Setting each bit to '1'b1' indicates that the corresponding channels belong to the high-priority level. Bit 0 = 1 indicates that the write commands of channel 0 belong to the high-priority group. Bit 0 = 0 indicates that the write commands of channel 0 belong to the low-priority group. Bit 1 = 1 indicates that the write commands of channel 1 belong to the high-priority group. Bit 1 = 0 indicates that the write commands of channel 1 belong to the low-priority group. Bit 2 = 1 indicates that the write commands of channel 2 belong to the high-priority group. Bit 2 = 0 indicates that the write commands of channel 2 belong to the low-priority group. Bit 3 = 1 indicates that the write commands of channel 3 belong to the high-priority group. Bit 3 = 0 indicates that the write commands of channel 3 belong to the low-priority group.

6.1.4.40 Channel Arbiter Grant Control Register C (Offset = 0xC8)

If register Offset 0x30 bit[30] "Indep_RW_EN" = '1', register Offset 0xC8 will be used for the write command. If register Offset 0x30 bit[30] "Indep_RW_EN" = '0', register Offset 0xC8 will be meaningless.

Table 6-40. Channel Arbiter Grant Control Register C (Offset = 0xC8)

Bit	Name	Type	Description
[31:29]	-	-	Reserved
[28:24]	ARB_CNT3_w	R/W	Once channel 3 is granted, the maximum allowed write commands will be issued. 0: 1 write command 1: 2 write commands 2: 3 write commands … 30: 31 write commands 31: 32 write commands
[23:21]	-	-	Reserved

Bit	Name	Type	Description
[20:16]	ARB_CNT2_w	R/W	Once channel 2 is granted, the maximum allowed write commands will be issued. 0: 1 write command 1: 2 write commands 2: 3 write commands … 30: 31 write commands 31: 32 write commands
[15:13]	-	-	Reserved
[12:8]	ARB_CNT1_w	R/W	Once channel 1 is granted, the maximum allowed write commands will be issued. 0: 1 write command 1: 2 write commands 2: 3 write commands … 30: 31 write commands 31: 32 write commands
[7:4]	-	-	reserved
[3:0]	ARB_CNT0_w	R/W	Once channel 0 is granted, the maximum allowed write commands will be issued. 0: 1 write command 1: 2 write commands 2: 3 write commands … 30: 31 write commands 31: 32 write commands

6.1.4.41 Debug Address (Offset = 0xD0)

Table 6-41. Debug Address (Offset = 0xD0)

Bit	Name	Type	Description
[31:0]	Base_addr	R/W	This register stores the base address to be monitored.

Note: If the Debug Control register (Offset = 0xEC) is enabled, this register will be read only.

6.1.4.42 Debug Address Mask (Offset = 0xD4)

Table 6-42. Debug Address Mask (Offset = 0xD4)

Bit	Name	Type	Description
[31:0]	Addr_mask	R/W	This register controls the mask field to the base address.

Note: If the Debug Control register (Offset = 0xEC) is enabled, this register will be read only.

6.1.4.43 Debug Write Data (Offset = 0xD8)

Table 6-43. Debug Write Data (Offset = 0xD8)

Bit	Name	Type	Description
[31:0]	Write_data	R/W	This register stores data for comparing with the write data of access.

Note: If the Debug Control register (Offset = 0xEC) is enabled, this register will be read only.

6.1.4.44 Debug Write Data Mask (Offset = 0xDC)

Table 6-44. Debug Write Data Mask (Offset = 0xDC)

Bit	Name	Type	Description
[31:0]	Data_mask	R/W*	This register controls the mask field to write data.

Note: If the Debug Control register (Offset = 0xEC) is enabled, this register will be read only.

6.1.4.45 Debug Master Control (Offset = 0xE0)

Table 6-45. Debug Master Control (Offset = 0xE0)

Bit	Name	Type	Description
[31:20]	-	-	Reserved
19	Mst_extype	R/W	Exclusive type 1: Access the monitored master will trigger the interrupt. 0: Access any other master will trigger the interrupt.
[18:16]	Mst_ch	R/W	Monitored master channel number
[15:0]	Mst_id	R/W	Monitored master ID number

Note: If the Debug Control register (Offset = 0xEC) is enabled, this register will be read only.

6.1.4.46 Debug Access Control (Offset = 0xE4)

Table 6-46. Debug Access Control (Offset = 0xE4)

Bit	Name	Type	Description
[31:2]	-	-	Reserved
1	lsw_sel	R/W	Last Significant Word (LSW) selection (Used only for the double-word access) When lsw_sel = 1: The content of the Debug Write Data register (Offset = 0xD8) will be compared with LSW of the current write access [31:0]. When lsw_sel = 0: The content of the Debug Write Data register (Offset = 0xD8) will be compared with LSW of the current write access [63:32].
0	Rw_sel	R/W	Read/Write selection 0: Monitor only the read access 1: Monitor only the write access

Note: If the Debug Control register (Offset = 0xEC) is enabled, this register will be read only.

6.1.4.47 Debug Policy Control (Offset = 0xE8)

User should know that enabling “read access” and “write data” simultaneously will not be reasonable. If the “read access” policy is enabled, the “write data” policy should be “don’t care”.

Table 6-47. Debug Policy Control (Offset = 0xE8)

Bit	Name	Type	Description
[31:5]	-	-	Reserved
4	Rw_en	R/W	Read/Write access policy enable
3	Mst_ch_en	R/W	Master channel policy enable
2	Mst_id_en	R/W	Master ID policy enable
1	Wdata_en	R/W	Write data policy enable
0	Add_en	R/W	Address policy enable

Note: If the Debug Control register (Offset = 0xEC) is enabled, this register will be read only.

6.1.4.48 Debug Control (Offset = 0xEC)

Table 6-48. Debug Control (Offset = 0xEC)

Bit	Name	Type	Description
[31:1]	-	-	Reserved
0	Debug_en	R/W*	Debug_enable

Note: The enable bit automatically becomes low after interrupt

6.1.4.49 Debug Address Status (Offset = 0xF0)

Table 6-49. Debug Address Status (Offset = 0xF0)

Bit	Name	Type	Description
[31:0]	Addr_status	R	This register records the address of an access.

6.1.4.50 Debug Write Data Status (Offset = 0xF4)

Table 6-50. Debug Write Data Status (Offset = 0xF4)

Bit	Name	Type	Description
[31:0]	Data_status	R	This register records the write data of an access.

6.1.4.51 Debug Master Status (Offset = 0xF8)

Table 6-51. Debug Master Status (Offset = 0xF8)

Bit	Name	Type	Description
[31:19]	-	-	Reserved
[18:16]	Mst_ch_status	R	The field records the number of the master channels of an access.
[15:0]	Mst_id_status	R	The field records the number of the master IDs of an access.

6.1.4.52 Debug Access Status (Offset = 0xFC)

Table 6-52. Debug Access Status (Offset = 0xFC)

Bit	Name	Type	Description
[31:1]	-	-	Reserved
0	Ac_stutus	R	The field records the type of an access.

6.1.4.53 DDR PHY Read Path DLL Delay Tuning for Falling Edge Register (PHYRDTFR, Offset = 0x130)

DLSEL_RD_FALL_byten is used to adjust the timing relationship between read DQ and read DQS at the falling edge and register Offset 0x24 is used to adjust the timing relationship at the rising edge.

Table 6-53. DDR PHY Read Path DLL Delay Tuning for Falling Edge Register (PHYRDTFR, Offset = 0x130)

Bit	Name	Type	Default Value	Description
[31:8]	-	-	Rsvd	Reserved
7	-	-	Rsvd	Reserved
[6:4]	DLSEL_RD_FALL_byte1	R/W	0x0	Delay setting of byte1 to adjust the falling-edge delay of DQS or DQ for read

Bit	Name	Type	Default Value	Description
3	-	-	Rsvd	Reserved
[2:0]	DLSEL_RD_FALL_byte0	R/W	0x0	Delay setting of byte0 to adjust the falling-edge delay of DQS or DQ for read

6.1.4.54 DDR PHY MISC2 Control Register (PHYMISCR2, Offset = 0x134)

Please refer to the DDRx PHY application note for more detailed information.

Table 6-54. DDR PHY MISC2 Register (PHYMISCR2, Offset = 0x134)

Bit	Name	Type	Default Value	Description
[31:15]	-	-	Rsvd	Reserved
14	IO15V	R/W	0x1	I/O voltage selection 1: Combo PHY is operated at1.5V or 1.8V. 0: Combo PHY is operated at 1.2V or 1.35V.
13	RDLVL_GATE_EN	R/W	0x0	MSDLY margin check 1: Enable 0: Disable
12	EYE_RDLVL_EN	R/W	0x0	Eye margin check of DQ for DQS latch 1: Enable 0: Disable
11	-	-	Rsvd	Reserved
[10:8]	VREF_SELECT	R/W	0x4	Bias voltage by PHY for VREF when 0x20 bit[15] self_bias = 1 000: 0.55 * VCC15O_DDR 001: 0.525 * VCC15O_DDR 010: 0.475 * VCC15O_DDR 011: 0.45 * VCC15O_DDR 100 ~ 111: 0.5 * VCC15O_DDR
7	-	-	Rsvd	Reserved
[6:4]	RONMD_DATA	R/W	0x7	The driving impedance setting of DDR I/O (Data block) 3'b111: 34Ω 3'b110: 40Ω 3'b101: 48Ω 3'b100: 60Ω

Bit	Name	Type	Default Value	Description
				3’b011: 60Ω
				3’b010: 80Ω
				3’b001: 120Ω
				3’b000: 240Ω
3	-	-	Rsvd	Reserved
[2:0]	RONMD_CMDADDR	R/W	0x7	The driving impedance setting of DDR I/O (CMDADDR block)
				3’b111: 34Ω
				3’b110: 40Ω
				3’b101: 48Ω
				3’b100: 60Ω
				3’b011: 60Ω
				3’b010: 80Ω
				3’b001: 120Ω
				3’b000: 240Ω

6.1.5 Memory Address Table (MA Table)

The DDR controller supports various DDR3/DDR2 SDRAMs. Users must refer to the MA tables for the external rank0 configuration register, RNK_TYPE. Each rank must choose the correct MA table. The controller supports three types of MA tables, which can be configured through the AMTSEL bit of the configuration register (Offset = 0x00).

- When AMTSEL = 00, the system address maps with RA, BA, and CA.
- When AMTSEL = 01, the system address maps with BA, RA, and CA.
- When AMTSEL = 10, the system address maps with RA, CA, BA, and CA.

For a 16bit memory interface, the bank address mapping is inserted between the column addresses. ADDR is mapped to RA, CA, BA, and CA.

- Bits[7:6] of the system address are configured as the bank addresses for four banks of DDR3 SDRAM.
- Bits[8:6] of the system address are configured as the bank addresses for eight banks of DDR3 SDRAM .

Table 6-55 and Table 6-56 are the MA tables in the DDR3/DDR2 modes when AMTSEL = 00. Table 6-57 and Table 6-58 are the MA tables in the DDR3/DDR2 mode when AMTSEL = 01. Table 6-59 is the MA table in the DDR3/DDR2 mode when AMTSEL = 10.

Table 6-55. MA Table (AMTSEL = 00) in 8bit DDR3/DDR2 Mode

MATABLE	Row Address x Column Address x Bank Address					
TYPE	**13X9X2**	**13X10X2**	**14X10X2**	**12X10X3**	**13X10X3**	**14X10X3**
SELECT	**000**	**001**	**010**	**011**	**100**	**101**
Column	**CA**	**CA**	**CA**	**CA**	**CA**	**CA**
MA0	0	0	0	0	0	0
MA1	1	1	1	1	1	1
MA2	2	2	2	2	2	2
MA3	3	3	3	3	3	3
MA4	4	4	4	4	4	4
MA5	5	5	5	5	5	5
MA6	6	6	6	6	6	6
MA7	7	7	7	7	7	7
MA8	8	8	8	8	8	8
MA9	-	9	9	9	9	9
MA10	-	-	-	-	-	-
Bank	BA	BA	BA	BA	BA	BA
BA0	9	10	10	10	10	10
BA1	10	11	11	11	11	11
BA2	-	-	-	12	12	12
Row	RA	RA	RA	RA	RA	RA
MA0	11	12	12	13	13	13
MA1	12	13	13	14	14	14
MA2	13	14	14	15	15	15
MA3	14	15	15	16	16	16
MA4	15	16	16	17	17	17
MA5	16	17	17	18	18	18
MA6	17	18	18	19	19	19
MA7	18	19	19	20	20	20
MA8	19	20	20	21	21	21

MATABLE	Row Address x Column Address x Bank Address					
TYPE	**13X9X2**	**13X10X2**	**14X10X2**	**12X10X3**	**13X10X3**	**14X10X3**
SELECT	**000**	**001**	**010**	**011**	**100**	**101**
Column	**CA**	**CA**	**CA**	**CA**	**CA**	**CA**
MA9	20	21	21	22	22	22
MA10	21	22	22	23	23	23
MA11	22	23	23	24	24	24
MA12	23	24	24	-	25	25
MA13	-	-	25	-	-	26

Table 6-56. MA Table (AMTSEL = 00) in 16bit DDR3/DDR2 Mode

MATABLE	Row Address x Column Address x Bank Address					
TYPE	**13X9X2**	**13X10X2**	**14X10X2**	**12X10X3**	**13X10X3**	**14X10X3**
SELECT	**000**	**001**	**010**	**011**	**100**	**101**
Column	**CA**	**CA**	**CA**	**CA**	**CA**	**CA**
MA0	1	1	1	1	1	1
MA1	2	2	2	2	2	2
MA2	3	3	3	3	3	3
MA3	4	4	4	4	4	4
MA4	5	5	5	5	5	5
MA5	6	6	6	6	6	6
MA6	7	7	7	7	7	7
MA7	8	8	8	8	8	8
MA8	9	9	9	9	9	9
MA9	-	10	10	10	10	10
MA10	-	-	-	-	-	-
Bank	BA	BA	BA	BA	BA	BA
BA0	10	11	11	11	11	11
BA1	11	12	12	12	12	12
BA2	-	-	-	13	13	13
Row	RA	RA	RA	RA	RA	RA
MA0	12	13	13	14	14	14
MA1	13	14	14	15	15	15
MA2	14	15	15	16	16	16

MATABLE	Row Address x Column Address x Bank Address					
TYPE	**13X9X2**	**13X10X2**	**14X10X2**	**12X10X3**	**13X10X3**	**14X10X3**
SELECT	**000**	**001**	**010**	**011**	**100**	**101**
Column	**CA**	**CA**	**CA**	**CA**	**CA**	**CA**
MA3	15	16	16	17	17	17
MA4	16	17	17	18	18	18
MA5	17	18	18	19	19	19
MA6	18	19	19	20	20	20
MA7	19	20	20	21	21	21
MA8	20	21	21	22	22	22
MA9	21	22	22	23	23	23
MA10	22	23	23	24	24	24
MA11	23	24	24	25	25	25
MA12	24	25	25	-	26	26
MA13	-	-	26	-	-	27

Table 6-57. MA Table (AMTSEL = 01) in 8bit DDR3/DDR2 Mode

MATABLE	Row Address x Column Address x Bank Address					
TYPE	**13X9X2**	**13X10X2**	**14X10X2**	**12X10X3**	**13X10X3**	**14X10X3**
SELECT	**000**	**001**	**010**	**011**	**100**	**101**
Column	**CA**	**CA**	**CA**	**CA**	**CA**	**CA**
MA0	0	0	0	0	0	0
MA1	1	1	1	1	1	1
MA2	2	2	2	2	2	2
MA3	3	3	3	3	3	3
MA4	4	4	4	4	4	4
MA5	5	5	5	5	5	5
MA6	6	6	6	6	6	6
MA7	7	7	7	7	7	7
MA8	8	8	8	8	8	8
MA9	-	9	9	9	9	9
MA10	-	-	-	-	--	-
Bank	BA	BA	BA	BA	BA	BA
BA0	22	23	24	22	23	24

MATABLE	Row Address x Column Address x Bank Address					
TYPE	**13X9X2**	**13X10X2**	**14X10X2**	**12X10X3**	**13X10X3**	**14X10X3**
SELECT	**000**	**001**	**010**	**011**	**100**	**101**
Column	**CA**	**CA**	**CA**	**CA**	**CA**	**CA**
BA1	23	24	25	23	24	25
BA2	-	-	-	24	25	26
Row	RA	RA	RA	RA	RA	RA
MA0	9	10	10	10	10	10
MA1	10	11	11	11	11	11
MA2	11	12	12	12	12	12
MA3	12	13	13	13	13	13
MA4	13	14	14	14	14	14
MA5	14	15	15	15	15	15
MA6	15	16	16	16	16	16
MA7	16	17	17	17	17	17
MA8	17	18	18	18	18	18
MA9	18	19	19	19	19	19
MA10	19	20	20	20	20	20
MA11	20	21	21	21	21	21
MA12	21	22	22	-	22	22
MA13	-	-	23	-	-	23

Table 6-58. MA Table (AMTSEL = 01) in 16bit DDR3/DDR2 Mode

MATABLE	Row Address x Column Address x Bank Address					
TYPE	**13X9X2**	**13X10X2**	**14X10X2**	**12X10X3**	**13X10X3**	**14X10X3**
SELECT	**000**	**001**	**010**	**011**	**100**	**101**
Column	**CA**	**CA**	**CA**	**CA**	**CA**	**CA**
MA0	1	1	1	1	1	1
MA1	2	2	2	2	2	2
MA2	3	3	3	3	3	3
MA3	4	4	4	4	4	4
MA4	5	5	5	5	5	5
MA5	6	6	6	6	6	6
MA6	7	7	7	7	7	7

MATABLE	Row Address x Column Address x Bank Address					
TYPE	**13X9X2**	**13X10X2**	**14X10X2**	**12X10X3**	**13X10X3**	**14X10X3**
SELECT	**000**	**001**	**010**	**011**	**100**	**101**
Column	**CA**	**CA**	**CA**	**CA**	**CA**	**CA**
MA7	8	8	8	8	8	8
MA8	9	9	9	9	9	9
MA9	-	10	10	10	10	10
MA10	-	-	-	-	-	-
Bank	BA	BA	BA	BA	BA	BA
BA0	23	24	25	23	24	25
BA1	24	25	26	24	25	26
BA2	-	-	-	25	26	27
Row	RA	RA	RA	RA	RA	RA
MA0	10	11	11	11	11	11
MA1	11	12	12	12	12	12
MA2	12	13	13	13	13	13
MA3	13	14	14	14	14	14
MA4	14	15	15	15	15	15
MA5	15	16	16	16	16	16
MA6	16	17	17	17	17	17
MA7	17	18	18	18	18	18
MA8	18	19	19	19	19	19
MA9	19	20	20	20	20	20
MA10	20	21	21	21	21	21
MA11	21	22	22	22	22	22
MA12	22	23	23		23	23
MA13	-	-	24	-	-	24

Table 6-59. MA Table (AMTSEL = 10) in 16bit DDR3/DDR2 Mode

MATABLE	Row Address x Column Address x Bank Address					
TYPE	**13X9X2**	**13X10X2**	**14X10X2**	**12X10X3**	**13X10X3**	**14X10X3**
SELECT	**000**	**001**	**010**	**011**	**100**	**101**
Column	**CA**	**CA**	**CA**	**CA**	**CA**	**CA**
MA0	1	1	1	1	1	1
MA1	2	2	2	2	2	2
MA2	3	3	3	3	3	3
MA3	4	4	4	4	4	4
MA4	5	5	5	5	5	5
MA5	8	8	8	9	9	9
MA6	9	9	9	10	10	10
MA7	10	10	10	11	11	11
MA8	11	11	11	12	12	12
MA9	-	12	12	13	13	13
MA10	-	-	-	-	-	-
Bank	BA	BA	BA	BA	BA	BA
BA0	6	6	6	6	6	6
BA1	7	7	7	7	7	7
BA2	-	-	-	8	8	8
Row	RA	RA	RA	RA	RA	RA
MA0	12	13	13	14	14	14
MA1	13	14	14	15	15	15
MA2	14	15	15	16	16	16
MA3	15	16	16	17	17	17
MA4	16	17	17	18	18	18
MA5	17	18	18	19	19	19
MA6	18	19	19	20	20	20
MA7	19	20	20	21	21	21
MA8	20	21	21	22	22	22
MA9	21	22	22	23	23	23
MA10	22	23	23	24	24	24
MA11	23	24	24	25	25	25
MA12	24	25	25	-	26	26

MATABLE	Row Address x Column Address x Bank Address					
TYPE	**13X9X2**	**13X10X2**	**14X10X2**	**12X10X3**	**13X10X3**	**14X10X3**
SELECT	**000**	**001**	**010**	**011**	**100**	**101**
Column	**CA**	**CA**	**CA**	**CA**	**CA**	**CA**
MA13	-	-	26	-	-	27

6.1.6 Initialization Sequence

6.1.6.1 Initialization Sequence of DDR2 Mode

Please follow the sequence below to initialize the DDRx controller with Faraday pure DDR2 PHY.

1. Wait until all the power supplies, reference voltages, and clocks become stable.
   (a) Wait 200μs
   (b) Release the DDRx PHY DLLPDN
   (c) Set the Configuration Register (Offset = 0x00)
   Set the Mode Register and Extended Mode Value Register (Offset = 0x08)
   Set the Extended Mode Register2 and Extended Mode Register 3 (Offset = 0x0C)
   Set the Rank0/1 Register (Offset = 0x10)
   Set the DDRx Timing Parameter 0 Register (Offset = 0x14)
   Set the DDRx Timing Parameter 1 Register (Offset = 0x18)
   Set the DDRx Timing Parameter 2 Register (Offset = 0x1C)
   Set the PHY Command and Data Block Control Register (Offset = 0x20)
   Set the PHY Read Path DLL Delay Tuning Control Register (Offset = 0x24)
   Set the PHY Compensation Block control Register (Offset = 0x28)
   Set the Arbitration Setup Register (Offset = 0x30)
   Set the Channel Arbiter Grant Count –A Register (Offset = 0x34)
   Set the Channel Arbiter Grant Count –B Register (Offset = 0x38)
   Set the DDRx PHY Write/Read Timing Control Register (Offset = 0x3C)
   Set the msdly byte control Register (Offset = 0x74)
   (d) Release DDR PHY RESETN
   (e) Set the DDRx Controller State Control Register (Offset = 0x04)
   Bit 0: '1' (Initial Command) to initiate the DDRx SDRAM

(f) Read bit 8 of the DDRx Controller State Control Register (Offset = 0x04). When this bit (Initial OK) is set to '1', it indicates that the DDRx SDRAM is well initialized and ready for use.

2. Please refer to the Initial Code in the DDR2 667 32bit system.

   (Assume that the Offset = 0x00 Register system address is at 0x90300000, DDR2 Memory Space is at 0x10000000.)

   mem.write 0x90300000 0x08035e01 (DDR2 1:2 mode) 0x08075e01 (DDR2 1:1 mode)
   mem.write 0x90300008 0x04040852
   mem.write 0x90300010 0x10015656
   mem.write 0x90300014 0x2b10120d
   mem.write 0x90300018 0x22401434
   mem.write 0x9030001C 0x01003251
   mem.write 0x90300020 0x00000f41
   mem.write 0x90300024 0x11111111
   mem.write 0x90300030 0x90000000
   mem.write 0x90300034 0x05050505
   mem.write 0x90300038 0x05050505
   mem.write 0x9030003C 0x00450013
   mem.write 0x90300074 0x11111111 (Please refer to DDRx PHY application note.)
   mem.write 0x90300004 0x00000001
   mem.read 0x90300004

3. Please refer to the typical DDR2 SDRAM timing parameter.

   MClk = 266MHz (DDR2 533); users should refer to the DDR3/DDR2 SDRAM data sheet for the timing parameters of each specific DDR2 SDRAM.

   TREFI = 7800ns is for the commercial requirement and 3900ns is for the industrial requirement.

   TRFC may be different depending on different chip sizes.

DDR2 SDRAM Timing Parameters	Typical Value(DDR2 400) unit : ns	unit : MClk	Typical Value(DDR2 533) unit : ns	unit : MClk	Typical Value(DDR2 667) unit : ns	unit : MClk	Typical Value(DDR2 800) unit : ns	unit : MClk
tRCD	15	3	15	4	15	5	15	6
tRAS	40	8	45	12	45	15	45	18
tRC	55	11	55	15	57	19	60	24
tRRD	10	2	10	3	10	4	10	4
tFAW	50	10	50	14	50	17	50	20
tRP	15	3	15	4	15	5	15	6
tWTR	7.5	2	7.5	2	7.5	3	7.5	3
tWR	15	3	15	4	15	5	15	6
tRTP	7.5	2	7.5	2	7.5	3	7.5	3
CL	3tCK	3	4tCK	4	5tCK	5	6tCK	6
tMRD	2tCK	2	2tCK	2	2tCK	2	2tCK	2
tXSNR	137.5	37	137.5	37	137.5	46	137.5	55
tXSRD	200 tCK	200	200 tCK	200	200 tCK	200	200 tCK	200

6.1.6.2 Initialization Sequence of DDR3 Mode

Please follow the sequence below to initialize the DDR3/DDR2 controller and PHY. Users should wait until all power supplies, reference voltages, and clocks become stable.

(A) Wait 200μs

(B) Release PLL_PDN of DDR PHY

(C) Set the Configuration Register (Offset = 0x00)

- Set the Mode Register and Extended Mode Value Register (Offset = 0x08)
- Set the Extended Mode Register2 and Extended Mode Register3 (Offset = 0x0C)
- Set the Rank0/1 Register (Offset = 0x10)
- Set the DDR3 Timing Parameter 0 Register (Offset = 0x14)
- Set the DDR3 Timing Parameter 1 Register (Offset = 0x18)
- Set the DDR3 Timing Parameter 2 Register (Offset = 0x1C)
- Set the PHY Command and Data Block Control Register (Offset = 0x20)
- Set the PHY Read Path DLL Delay Tuning Control Register (Offset = 0x24)
- Set the PHY Compensation Block control Register (Offset = 0x28)
- Set the Arbitration Setup Register (Offset = 0x30)
- Set the Channel Arbiter Grant Count –A Register (Offset = 0x34)
- Set the Channel Arbiter Grant Count –B Register (Offset = 0x38)
- Set the DDRx PHY Write/Read Timing Control Register (Offset = 0x3C)

- Set the Write-leveling Control Registers (Offsets =0x60, 0x64, 0x68)
- Set the Read-leveling Control registers (Offsets = 0x70, 0x74)

(D) Wait 100μs after (B) to release PHY_PLL RESETN of DDR

(E) Release PHY DLL_PDN of DDR

(F) Wait 100μs after (E) to release DLL_RESETN of DDR PHY

(G) Set the DDRx Controller State Control Register (Offset = 0x04)
Set bit 0 to '1' (Initial command) to initiate DDRx SDRAM

(H) Read bit 8 of the DDRx Controller State Control Register (Offset = 0x04). When this bit (Initial OK) is set to '1', it means that the DDRx SDRAM is well initialized and ready for use.

Please refer to the initial code in the 16bit DDR3-1066 system.

(Assume that the Offset 0x00 Register System Address is at 0x90300000, the DDR3 memory space will be at 0x10000000)

```
mem.write 0x90300000 0x00005d01
mem.write 0x90300008 0x00041a40
mem.write 0x9030000C 0x00000010
mem.write 0x90300010 0x10015656
mem.write 0x90300014 0x681f271c
mem.write 0x90300018 0x636d3b5a
mem.write 0x9030001C 0x01105551
mem.write 0x90300020 0x00000f41
mem.write 0x90300024 0x00000000
mem.write 0x90300028 0x00000e3a
mem.write 0x90300030 0x1600d5e5
mem.write 0x90300034 0x5835f556
mem.write 0x90300038 0x05050305
mem.write 0x9030003C 0x00530012
mem.write 0x90300060 0x00000f0e
mem.write 0x9030006C 0x44444444
mem.write 0x90300004 0x00000001
mem.read 0x90300004
```

Please refer to the DDR3 SDRAM timing parameter.

Users should refer to the DDR3 SDRAM data sheet for the timing parameters of specific DDR3 SDRAM. TREFI = 7800ns is the commercial requirement and TREFI = 3900ns is the industrial requirement. TRFC may be different depending on different chip sizes.

Note: 1T is the DDR3 SDRAM clock cycle.

DDR3 SDRAM Timing Parameters	Symbol	Typical Value(DDR3 800) unit : ns AL=0, CL=6, CWL=5	Typical Value(DDR3 1066) unit : ns AL=0, CL=7, CWL=6	Typical Value(DDR3 1333) unit : ns AL=0, CL=9, CWL=7	Typical Value(DDR3 1600) unit : ns AL=0, CL=11, CWL=8
Command and Address Timing	tRCD	15	15	15	13.75
	tRP	15	15	15	13.75
	tRC	52.5	52.5	51	48.75
	tRAS	37.5	37.5	36	35
	tRRD	10	10	7.5	7.5
	tFAW	50	50	45	40
	tWR	15	15	15	15
	tWTR	7.5	7.5	7.5	7.5
	tRTP	7.5	7.5	7.5	7.5
	tCCD	4T	4T	4T	4T
	tMRD	4T	4T	4T	4T
	tMOD	12T	12T	12T	12T
	tMPRR	1T	1T	1T	1T
Self Refresh Timing	tXS	120	120	120	120
	tXSDLL	512T	512T	512T	512T
	tCKESR	4T	4T	4T	4T
	tCKSRE	10	10	10	10
	tCKSRX	10	10	10	10
Power Down Timing	tCKE	7.5	5.625	5.625	5
	tCPDED	1T	1T	1T	1T
	tPD	tCKE			
	tANPD	tWL -1			
	tPDE	119	119	119	119
	tPDX	tANPD + tXPDLL			
Power Down Entry	tACTPDE	1T	1T	1T	1T
	tPRPDEN	1T	1T	1T	1T
	tREFPDEN	1T	1T	1T	1T
	tMRSPDE	tMOD			
	tRDPDEN	tRL+4+1			
	tWRPDEN	tWL +4 + tWR			
	tWRAPDE	tWL + 4 + tWR +1			
Power Down	tXP	7.5	7.5	6	6
	tXPDLL	24	24	24	24
Initialization and Reset Timing	tDLLK	512T	512T	512T	512T
	tZQinit	512T	512T	512T	512T
	tZQoper	256T	256T	256T	256T
	tZQcs	64T	64T	64T	64T
ODT Timin	ODTL on	tWL -2	tWL -2	tWL -2	tWL -2
	ODTL off	tWL -2	tWL -2	tWL -2	tWL -2
	tAONPD	1T	1T	1T	1T
	tAOFPD	1T	1T	1T	1T
	tODTH8	6T	6T	6T	6T
Dynamic ODT	ODTLcnw	tWL -2	tWL -2	tWL -2	tWL -2
	ODTLcnw	6 + ODTL off	6 + ODTL off	6 + ODTL off	6 + ODTL off
Write Leveling	tWLMRD	40T	40T	40T	40T
	tWLDQSE	25T	25T	25T	25T

6.2 SPI NAND/NOR Flash Controller

6.2.1 General Description

The SPI controller has one AHB slave port and one SPI interface controller to execute the SPI Flash command. Moreover, it also provides the PIO mode or DMA mode to access data from the AHB data port.

6.2.2 Features

- Supports SPI NAND/NOR Flash
- Supports DTR mode
- Supports DMA handshake mode and CPU PIO mode
- Supports SPI serial mode, dual mode, and quad mode

6.2.3 Memory Map/Register Definition

6.2.3.1 Summary of Control Registers

Table 6-60. Summary of Control Registers

Address	Type	Description	Reset Value
Command queue control			
0x0_0000	R/W	Command Queue Word 0	0x0000_0000
0x0_0004	R/W	Command Queue Word 1	0x0100_0003
0x0_0008	R/W	Command Queue Word 2	0x0000_0000
0x0_000C	R/W	Command Queue Word 3	0x0000_0000
SPI control			
0x0_0010	R/W	Control Register	0x0000_0000
0x0_0014	R/W	AC Timing Register	0x0000_00FF
0x0_0018	R	Status Register	0x0000_0001
0x0_0020	R/W	Interrupt Control Register	0x0000_0000
0x0_0024	R/W1C	Interrupt Status Register	0x0000_0000
0x0_0028	R	SPI Read Status Register	0x0000_0000
0x0_0050	R	Revision Register	-

Address	Type	Description	Reset Value
0x0_0054	R	Feature Register	-
0x0_0100	R/W	Data Port Register	-

6.2.3.2 Register Definitions

The following subsections provide the detailed descriptions of the control registers

6.2.3.3 Command Word 0 ~ 3 (Address = 0x0000 ~ 0x000C)

One command contains four words. One command will be executed by writing the fourth word of a command and completely executed by the host SPI controller.

Notes:

1. The command must be executed before reading/writing data to FIFO.
2. The reserved region of the command queue is not allowed to be filled to '1'.

6.2.3.3.1 Command Usage

For the write command, users must check and clear the command complete interrupt before writing the next command. For the read status command, users must check and clear the command complete interrupt before writing the next command. For the read data command, users must finish reading data from RXFIFO, and check and clear the command complete interrupt before writing the next command.

Table 6-61. Command Queue First Word (Address = 0x0000)

Bit	Name	Type	Description
[31:0]	SPI Flash Address	R/W	SPI Flash address This field decides the values of the SPI Flash address and issues this address to the SPI Flash. The byte of the address is executed by the address length when offset = 0x04.

Table 6-62. Command Queue Second Word (Address = 0x00004)

Bit	Name	Type	Description
[31:29]	-	-	Reserved

Bit	Name	Type	Description
28	Continuous Read Mode Enable	R/W	The continuous read mode state is located between the address and second dummy state. It can eliminate the instruction state after the second read command in the continuous read mode. 0: Disable the continuous read mode 1: Enable the 1byte continuous read mode
[27:26]	-	-	Reserved
[25:24]	Instruction length	R/W	Instruction code length When users want to execute a SPI Flash command, the instruction code must be included. Different SPI Flash venders have different instruction lengths; therefore, users can set this field to meet different behaviors. Instruction code is normally 1byte; however, if users set the 2byte instruction length, the host controller will issue this instruction code twice. 00: No instruction code 01: 1byte instruction code 10: 2byte instruction code (Repeat the instruction code) 11: Reserved
[23:16]	dum_2nd_cyc	R/W	Second dummy state cycle Second dummy state is located between the address and the data state that exclude the continuous read mode state. Users can check whether the dummy state exists between the address and the data state or not in the SPI Flash specification. The host controller will issue logic 1 in the dummy cycle. 8'd0: No second dummy state 8'd1 ~ 8'd32: One dummy second cycle to 32 dummy second cycles
[15:3]	-	-	Reserved
[2:0]	address length	R/W	This field decides the number of the SPI Flash address bytes. Users can set this field to set the address byte ranging from 1byte to 4bytes. 000: No address 001: 1byte address 010: 2byte address 011: 3byte address 100: 4byte address Others: Reserved

Table 6-63. Command Queue Third Word (Address = 0x0008)

Bit	Name	Type	Description
[31:0]	Data counter	R/W	Read/Write data counter This register must be set to '0' when performing the read status command. 32'b0: No read/write data 32'h1 ~ 32'hFFFF_FFFF: 1-byte data to 32'hFFFF_FFFF data Please note that no matter it is a data read or data write, this register is not allowed to be filled as '0'. However, for the read status, this register must set to '0'.

Table 6-64. Command Queue Fourth Word (Address = 0x000C)

Bit	Name	Type	Description
[31:24]	Instruction Code	R/W	Instruction code Users can set this code to execute the SPI Flash command.
[23:16]	Continuous Read Mode Code	R/W	Continuous read mode operate code Users can fill this code to execute the Continuous Read Mode.
[15:10]	-	-	Reserved
[9:8]	Start ce	R/W	The SPI controller can connect four SPI Flashes (Max.) and this field is used to select ce. 00: ce 0 01: ce 1 10: ce 2 11: ce 3
[7:5]	SPI Operate Mode	R/W	SPI operate mode 000: Serial mode 001: Dual mode 010: Quad mode 011: dual_io mode 100: quad_io mode Others: Reserved
4	DTR Mode	R/W	DTR (Double Transfer Rate) mode 0: Disable The read status, write data command, and DTR mode must be set to '0'. 1: Enable The DTR mode is valid for the read data command.

Bit	Name	Type	Description
3	Read Status	R/W	Read the SPI Flash status by using software or hardware It is only available when the read status is enabled and the write enable = 0. Users must issue the SPI Read Status command. 0: Read status by hardware, the controller will poll the status until the status is ready (Not busy), and then report the status register. 1: Read status by software, read status once and report the status register until users can read it.
2	Read Status Enable	R/W	Enable the Read SPI status It is available at write_enable = '0' and users must issue the SPI Read Status command. 0: Disable 1: Enable
1	Write enable	R/W	Enable the SPI write data, except for the read data or read status (Read the data return path); users must set the write enable = '1' for other SPI commands. Please note that in the write data or erase Flash command, write enable is set to '1'. Only when in the read data or read status command, write enable must be set to '0'.
0	-	-	Reserved

6.2.3.4 Control Register (Address = 0x0010)

This register must be set before executing the command, and it is prohibited to set this register when performing the command. Reserved region is not allowed to be filled to '1'.

Table 6-65. Control Register (Address = 0x0010)

Bit	Name	Type	Description
[31:19]	-	-	Reserved
[18:16]	rdy_loc	R/W	Busy bit of the SPI status Host will poll this busy bit and be ready at the HW read status. 3'b000 ~ 3'b111: Bit 0 ~ bit 7
[15:9]	-	-	Reserved
8	abort	R/W	Flush all commands/FIFOs and reset the state machine. When abort occurs, users must fill commands again. This bit will be automatically cleared to zero.
[7:5]	-	-	Reserved
4	spi_clk_mode	R/W	SPI clk mode at the IDLE state 0: mode0, sck_out will be low at the IDLE state. 1: mode3, sck_out will be high at the IDLE state.

Bit	Name	Type	Description
[3:2]	-	-	Reserved
[1:0]	spi_clk_divider	R/W	spi_clk divider. sck_out is divided by spi_clk and factor is listed as follows: 00: Divided by 2 01: Divided by 4 10: Divided by 6 11: Divided by 8 Note: When programming the DTR mode, only the divider by 4/8 is allowed.

6.2.3.5 AC Timing Register (Address = 0x0014)

Table 6-66. AC Timing Register (Address = 0x0014)

Bit	Name	Type	Description
[31:4]	-	-	Reserved
[3:0]	cs_delay	R/W	cs delay is the time from inactive cs to active cs, which means the timing from high to low. Please follow the specification for details. The unit is sck_out period. 4'b0000 ~ 4'b1111: 1cycle ~ 16cycles

6.2.3.6 Status Register (Address = 0x0018)

Table 6-67. Status Register (Address = 0x0018)

Bit	Name	Type	Description
[31:2]	-	-	Reserved
1	RXFIFO Ready	R	RXFIFO ready status When RXFIFO is ready, it indicates that RXFIFO is full, and the remain data of RXFIFO is less than the RXFIFO Depth and this is the last data.
0	TXFIFO Ready	R	TXFIFO ready status When TXFIFO is ready, it indicates that TXFIFO is empty, and users can transfer the data into TXFIFO to full.

6.2.3.7 Interrupt Control Register (Address = 0x0020)

Table 6-68. Interrupt Control Register (Address = 0x0020)

Bit	Name	Type	Description
[31:14]	-	-	Reserved
[13:12]	RXFIFO THOD	R/W	This field is used to set the trigger level for the RXFIFO threshold interrupt. The trigger level value is the data in RXFIFO. The unit is WORD.
[11:10]	-	-	Reserved
[9:8]	TXFIFO THOD	R/W	This field is used to set the trigger level for the TXFIFO Threshold Interrupt. The trigger level value is the remained data in TXFIFO. The unit is WORD.
[7:2]	-	-	Reserved
1	Cmd_cmplt_intr_en	R/W	Command complete interrupt enable 0: No interrupt 1: Enable command complete interrupt
0	DMA_EN	R/W	Enable the DMA handshake

Please note that this register can only be used for the DMA handshake.

Table 6-69. RXFIFO Trigger Level

RXFIFO_THOD	RXFIFO Trigger Level		
RXFIFO_DEPTH	**8**	**16**	**32**
2'b00	2	4	8
2'b01	4	8	16
2'b10	6	12	24
2'b11	-	-	-

Table 6-70. TXFIFO Trigger Level

TXFIFO_THOD	TXFIFO Trigger Level		
TXFIFO_DEPTH	**8**	**16**	**32**
2'b00	2	4	8
2'b01	4	8	16
2'b10	6	12	24
2'b11	-	-	-

6.2.3.8 Interrupt Status Register (Address = 0x0024)

Table 6-71. Interrupt Status Register (Address = 0x0024)

Bit	Name	Type	Description
[31:1]	-	-	Reserved
0	cmd_cmplt_intr	R/W1C	Command complete interrupt will be set when the command complete and intr_en enable at the command queue.

6.2.3.9 SPI Read Status Register (Address = 0x0028)

Table 6-72. SPI Read Status Register (Address = 0x0028)

Bit	Name	Type	Description
[31:8]	-	-	Reserved
[7:0]	SPI Read Status	R	Host issues the read SPI Flash status command and stores the return data at this register. Users can read this register to check the SPI Flash status.

6.2.3.10 Revision Register (Address = 0x0050)

Table 6-73. Revision Register (Address = 0x0050)

Bit	Name	Type	Description
[31:0]	Revision_Number	R	TBD

6.2.3.11 Feature Register (Address = 0x0054)

Table 6-74. Feature Register (Address = 0x0054)

Bit	Name	Type	Description
[31:26]	-	-	Reserved
25	CLK_MODE	R	The SPI controller clock mode 0: BYPORT mode 1: SYNC mode
24	DTR_MODE	R	DTR mode 0: Disable 1: Enable
[23:16]	-	-	Reserved
[15:8]	RXFIFO_DEPTH	R	RX FIFO depth
[7:0]	TXFIFO_DEPTH	R	TX FIFO depth

6.2.3.12 Data Port Register (Address = 0x0100)

Table 6-75. Data Port Register (Address = 0x0100)

Bit	Name	Type	Description
[31:0]	Data Port	R/W	Data port register Users can read/write data from the data port.

6.2.4 Functional Description

6.2.4.1 PIO Mode

Users can access the data port via the PIO mode or DMA mode. In the PIO mode, users must poll the RXFIFO/TXFIFO ready register (0x18). When RXFIFO is ready, users can read the whole RXFIFO data or remained data of RXFIFO out. When TXFIFO is ready, users can write the whole data into TXFIFO. Whenever operating in the PIO mode, please be sure to poll the status.

6.2.4.2 DMA Mode

DMA mode is the other option to access the data port. Users must set the DMA controller register to the DMA handshake mode and burst size of one handshake. The burst size of DMA depends on the SPI controller RX/TX threshold trigger level (0x20). If the RX/TX threshold trigger level is '8' (Unit: WORD), one DMA burst size must be 8 * 4 = 32bytes of the DMA transfer width. If the RX/TX threshold trigger level is 8 (Unit: WORD), the DMA burst size must be 8 * 1 = 8 WORDS for the DMA transfer width. If the RX/TX threshold trigger level is 8 (Unit: WORD), the DMA burst size must be 8 * 2 = 16 HALFWORDS for the DMA transfer width. The RX/TX threshold trigger unit is WORD. The host controller will issue a DMA request by reaching the threshold trigger level of TX/RX FIFO.

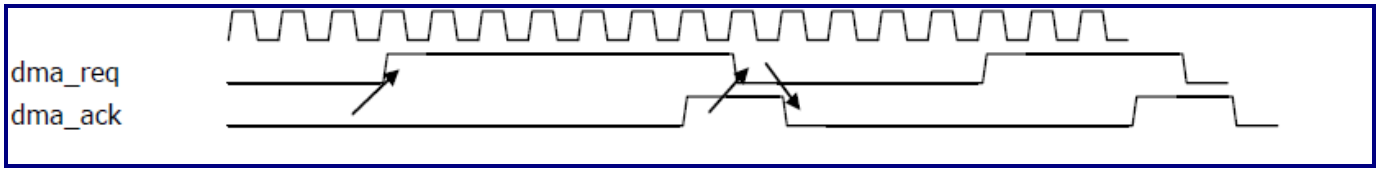


Figure 6-3. DMA Handshake Mode

6.2.4.3 SPI Flash Operation

A: Serial Mode

The host controller supports the SPI protocols, mode 0 and mode 3. Users must set the command queue register based on the specification of the SPI Flash. Please refer to the following waveforms to set the command queue.

If users want to execute the write enable command, please set the instruction code to 06h, write enable, and '1' for the instruction length. Please refer to Figure 6-4 for more details.

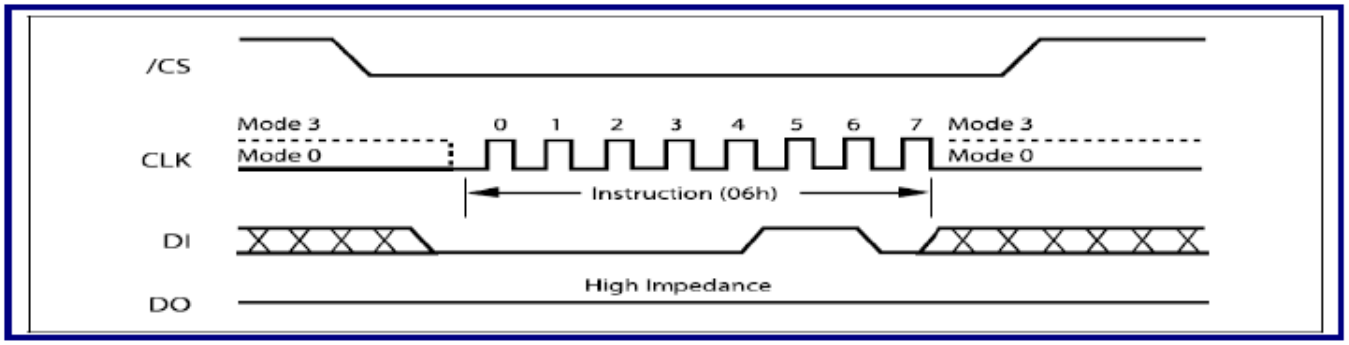


Figure 6-4. Write Enable

If users want to execute the page programming command, please set the instruction code to 02h, address register, address length, write enable, and '1' for the instruction length.

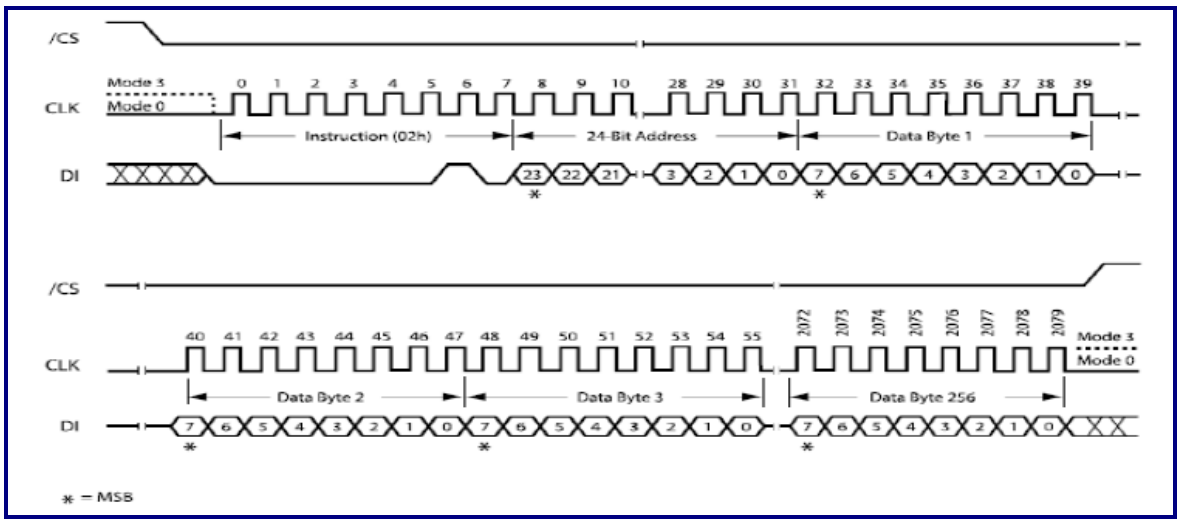


Figure 6-5. Page Program

If users want to execute the read status command, please set the instruction code to 05h/35h, read status enable, optional read status by hw/sw, and '1' to the instruction length. Please refer to Figure 6-6 for more details.

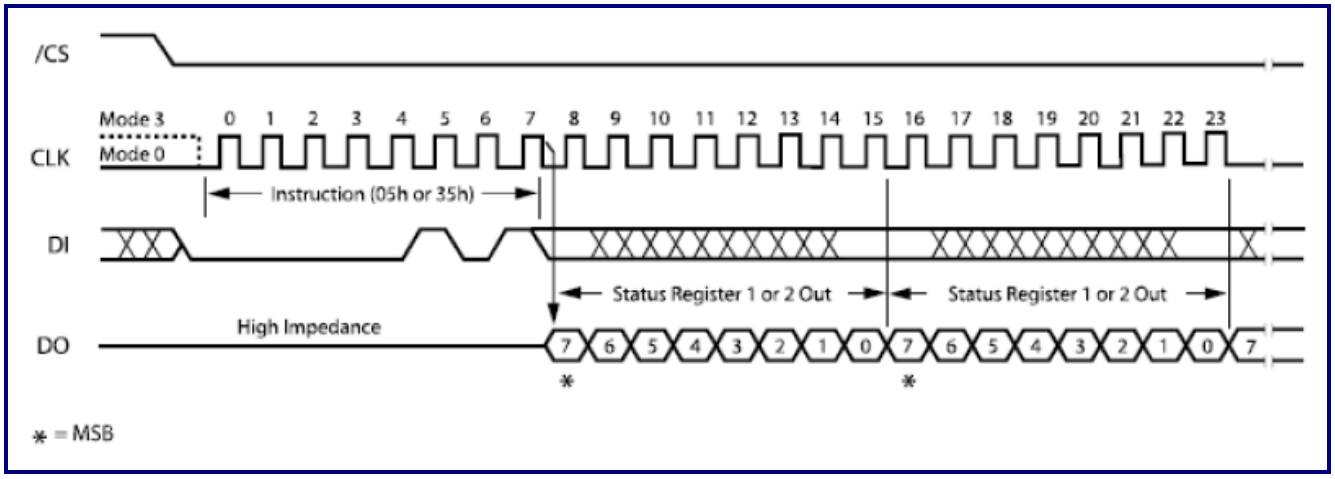


Figure 6-6. Read Status

If users want to execute the read data command, please set the instruction code/length to 1byte, address/address length to 3bytes, and write enable to '0'. Please refer to Figure 6-7 for more details.

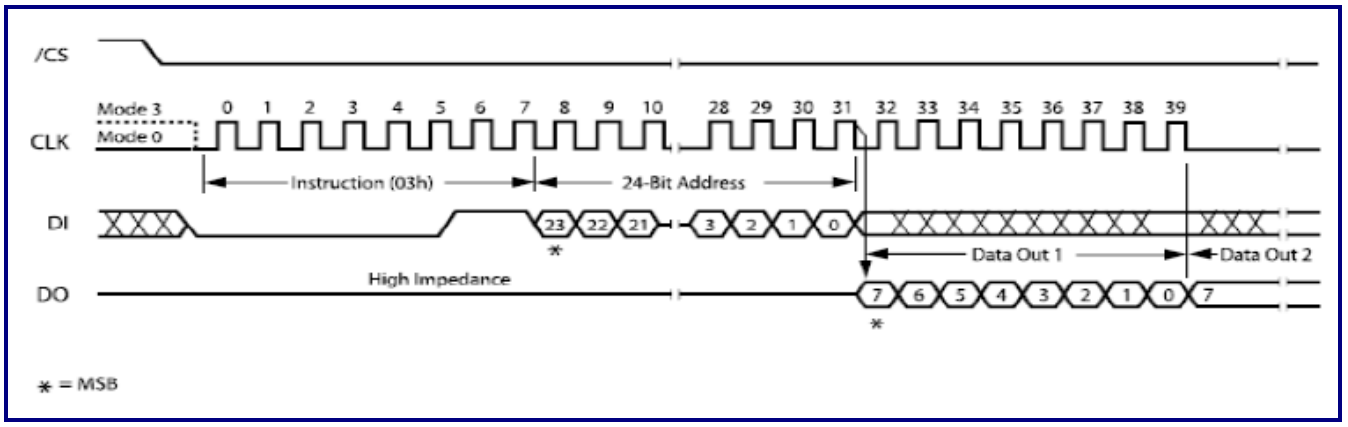


Figure 6-7. Read Data

B: Dual Mode/dual_io Mode

If users want to execute the fast read dual output command, please set the instruction code/length to '1', address/address length to 3bytes, dummy second cycle to 8, and operation mode to dual mode. Please refer to Figure 6-8 for more details.

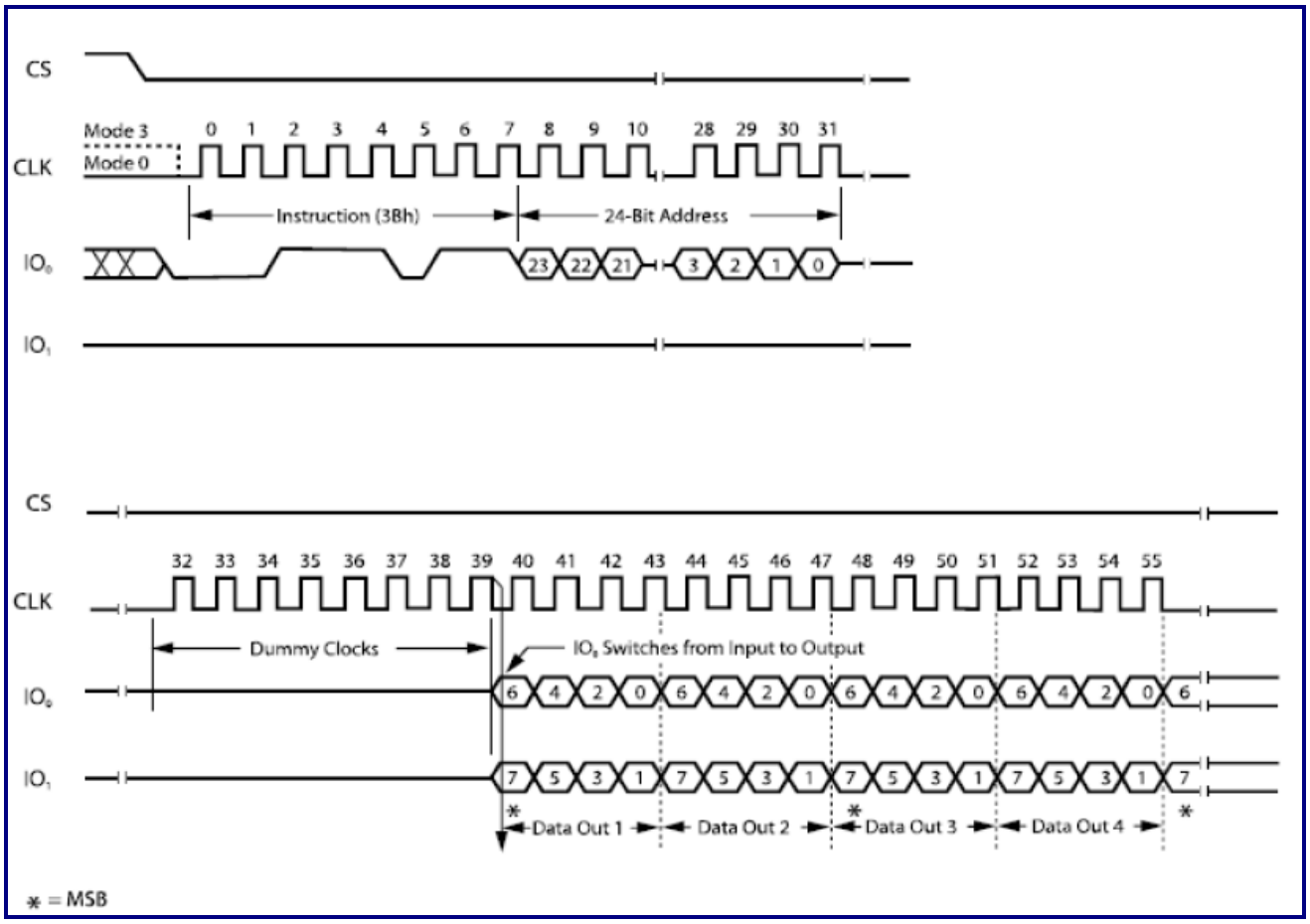


Figure 6-8. Fast Read Dual Output

If users want to execute the fast read dual I/O command, please set the instruction of code/length, address/address length, continuous read mode to enable, continuous read mode code, and operation to dual_io mode. Please refer to Figure 6-9 for more details.

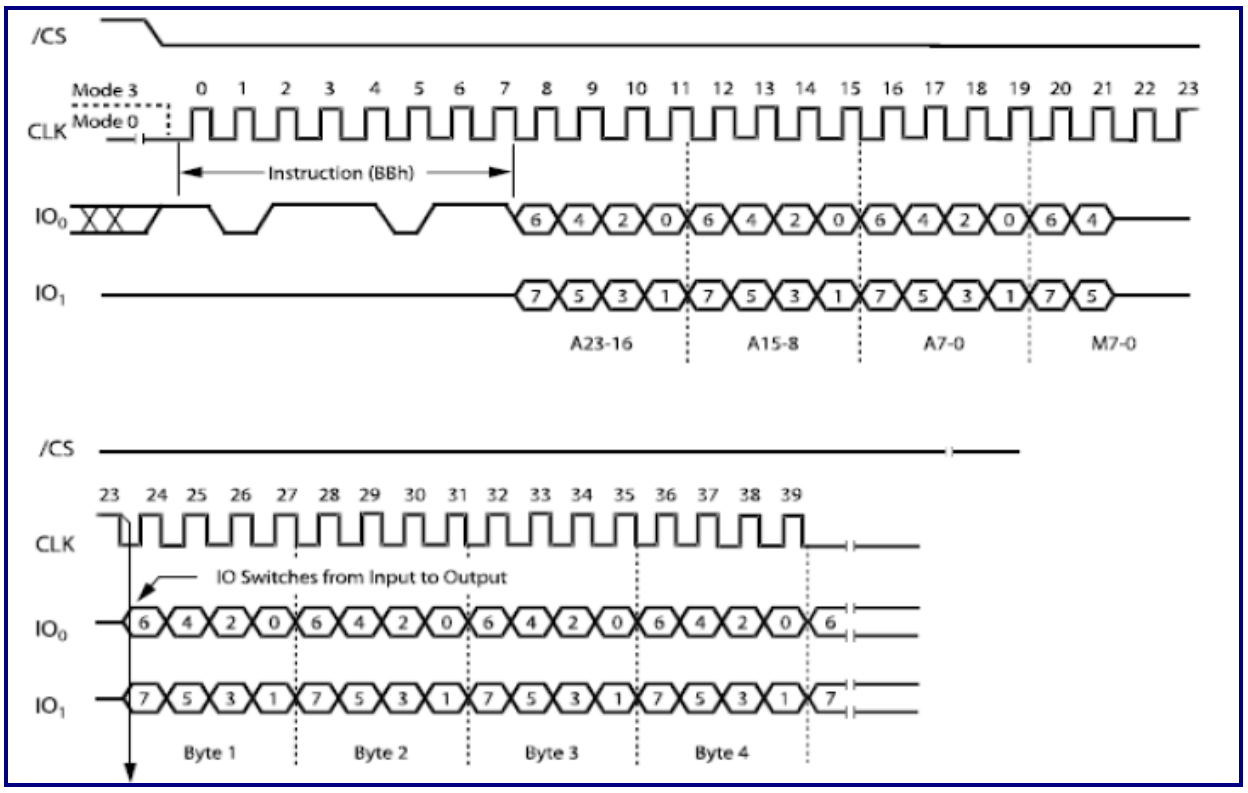


Figure 6-9. Fast Read Dual I/O

C: DTR Mode

The SPI controller also supports the Double Transfer Rate (DTR) mode and data are latched at the rising edge and falling edge. Users must fill the DTR mode (Offset = 0x0C) and other corresponding registers to execute the SPI command.

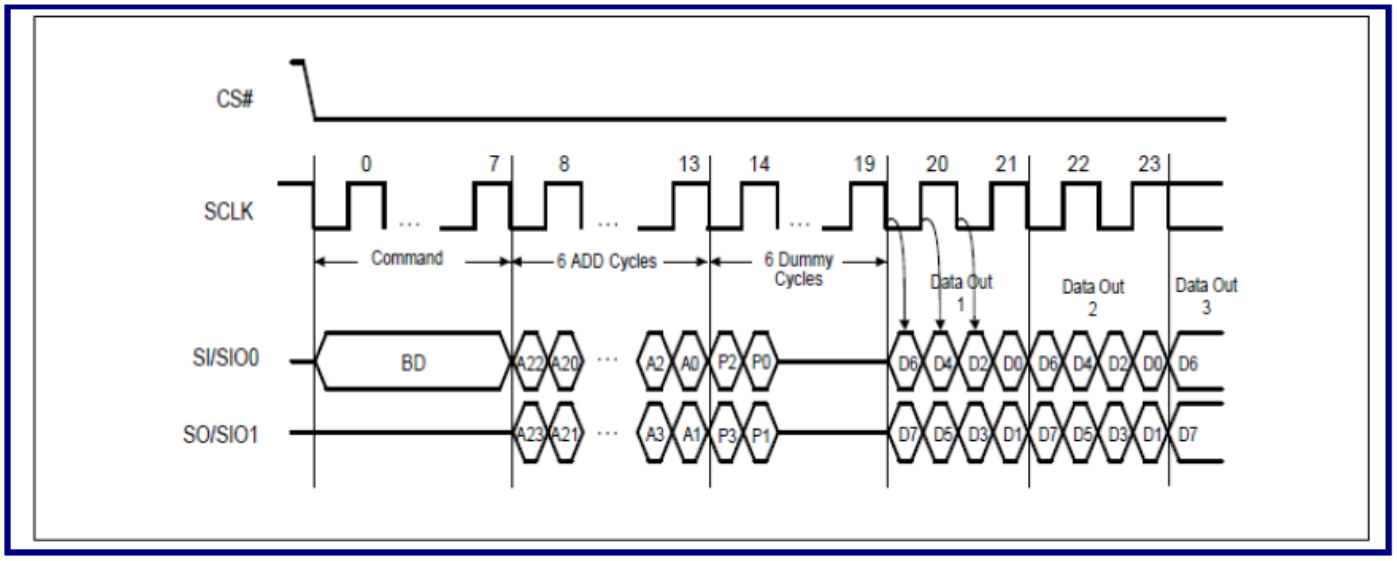


Figure 6-10. Fast Dual I/O DT Read (2DTRD) Sequence (Command BD)

6.2.5 Initialization

This section contains the following subsection:

- Software Programming Sequence

6.2.5.1 Software Programming Sequence

6.2.5.1.1 PIO Mode of Data Transfer

1. Set one command queue to meet the SPI operation flow and the command complete interrupt enable
2. Poll txfifo_ready/rxfifo_ready to write/read data to/from the data port
3. Wait the command complete interrupt and clear the interrupt

6.2.5.1.2 DMA Mode of Data Transfer

1. Set the SPI controller DMA threshold value and enable DMA
2. Set one command queue to meet the SPI operation flow and the command complete interrupt enable
3. Set the DMA controller register
4. Enable the DMA controller
5. Wait the command complete interrupt and clear the interrupt

6.2.5.1.3 Other Operations

1. Set one command queue to meet the SPI operation flow and the command complete interrupt enable
2. Wait the command complete interrupt and clear the interrupt

Chapter 7

Video Input Interface

This chapter contains the following sections:

7.1 Video Input Interface

GM8136S/GM8135S video input interface accepts multi-channel BT.656/BT.1120/BT.1302/CCIR-601 digital video formats from the external peripherals, also receives the image data from ISP (Image Signal Processor) and stores these sources to the video line buffer. It provides linear scaling up/down up to four video resolution outputs, video quality enhancement, OSD, Mark, Mask, and powerful motion detection function through the original image processing.

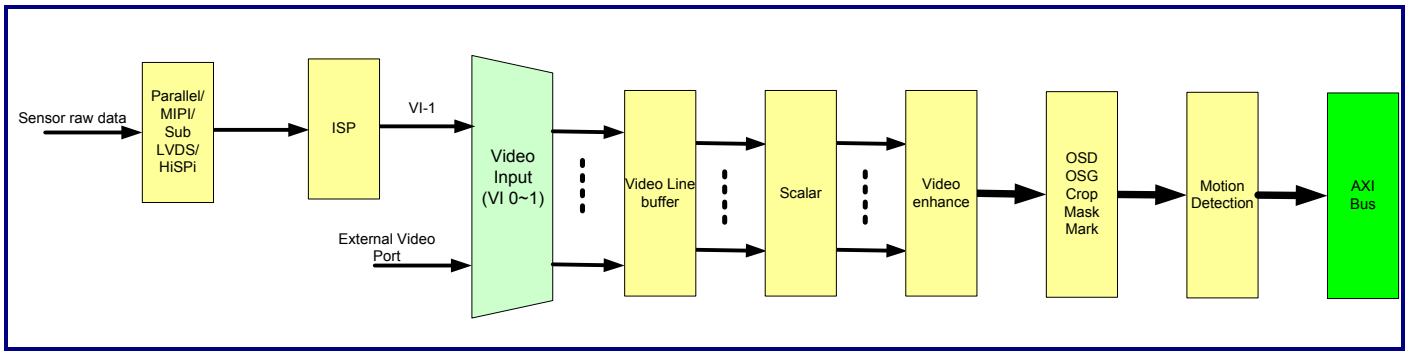


Figure 7-1. GM8136S/GM8135S Video Input Interface Function Blocks

7.2 Capture (Including MD)

7.2.1 General Description

GM8136S/GM8135S video input interface captures video data through a variety of video interfaces and outputs data to AMBA AXI. It provides the de-interlace function to reduce the video artifact for the interlace video. The noise reduction function can remove unwanted noise and preserve fine details and edges. With the scaling ability, users can individually reduce the image sizes of four paths to the necessary resolution. Up to 8 mask windows are allowed before scaling to mask assigned region on image. The color OSD function can help users paste characters at the captured video. Users can paste four individual logos by the mark function. The window clipping function can clip the target region from the image before or the image after scaling.

7.2.2 Features

Video Interface (VI)

- Scalable channel numbers and image resolutions depending on various applications
- Supports BT.656/BT.1120/BT.1302/CCIR-601 interfaces
- Embeds de-multiplexer function in video input front-end
- Supports VBI detection and extraction

Image Cropping and Channel Arbitration (SC, Source Cropping)

- Supports front-end image source cropping function
- Supports maximum image width of 2048pixels for image data from VI-0
- Supports maximum image width of 2048pixels for image data from VI-1

Scaling and de-noise (SD)

- Supports 1-D false color suppression
- Supports 1-D de-noise
- Supports four output resolutions, including scaling-down/up function on three output resolutions and bypasses another one
- Supports smooth and sharpness
- Supports field scaling (Top/Bottom offset)
- Supports frame to field extraction
- Supports four resolutions independent frame control

Motion Detection

- Maximum image size of up to 2048x1024pixels
- MD statistics region extracted from blocks in image
- In a frame, maximum block number up to MxN (Less than 128x128) and user programmable
- In a frame, maximum block size of up to 32x32
- Average pixel value of Y, Cb, or Cr output per block
- Calculates three Gaussian parameters (w, μ, and σ)
- Supports shadow detection

Mask

- Supports up to eight windows in one frame for each channel
- User programmable 16type palette color for mask

OSD Mark

- Programmable mark in RAM
- Data format: YCbCr422 (16bits per pixel)
- Flexible allocation on 1024pixel RAM size
  (For example, one 64x64 graph, two 64x32 graphs, or four 32x32 graphs)
- Supports flexible dimension on both horizontal and vertical direction:
  - 16pixels, 32pixels, 64pixels, 128pixels, 256pixels, and 512pixels
  - Zoom-in with locked aspect ratios: x2 and x4
- Supports up to four windows in one frame for each channel
- Selectable overlapping condition between font and mark
  (i.e.: Mark above font/Mark below font)
- 8 types of font/background transparencies:
  - 0%, 25%, 37.5%, 50%, 62.5%, 75%, 87.5%, and 100%

OSD Font

- Programmable font in RAM
- Font size: 12x18pixels
- 4bit (16type) palette for foreground and background
- Supports up to 455 standard fonts
- Zoom-in with locked aspect ratio: x2, x3, and x4
- Zoom-out with locked aspect ratio: x1/2
- Zoom-in without locked aspect ratio (Horizontal/Vertical):
  - x2/x1, x4/x1, x4/x2 ; x1/x2, x1/x4, and x2/x4
- Supports up to 8 windows in one frame for each channel
- Maximum 64 fonts on horizontal or vertical direction and maximum 256 fonts in one font window
- Supports up to 2048 fonts flexible allocated on all channels
  (Ex.: 1200 fonts on Ch00, 108 fonts on Ch01, 20 fonts on Ch03, ...)

- 8 types of font/background transparencies:
  - 0%, 25%, 37.5%, 50%, 62.5%, 75%, 87.5%, and 100%
- Supports border function with 16type border color
- Programmable border size (32pixels at maximum)
- Supports marquee function with 3 modes on each font window:
  - Marquee effect on one horizontal line word
  - Marquee effect on one vertical line word
  - Flip effect on one horizontal line word
- Programmable 12-type marquee steps:
  - 1024-step, 512-step, 256-step, 128-step, 64-step, 32-step, 16-step, 8-step, and 4-step
- Programmable marquee speed
- Supports mirror-flip mode

7.2.3 Memory Map/Register Definition

7.2.3.1 Mapping of Usual Special-purpose Registers

Table 7-1. Mapping of Usual Special-purpose Registers

Channel	Symbol	Offset			Access	Description
0	Sub-channel	0x0000	~	0x0027	R/W	Sub-channel setting of ch0
	FCS	0x0028	~	0x002F	R/W	False color suppression setting of ch0
	Mask	0x0030	~	0x006F	R/W	Mask function setting of ch0
	DN	0x0070	~	0x00B7	R/W	De-noise/Scaling setting of ch0
	OSD	0x00B8	~	0x015F	R/W	OSD/Mark function setting of ch0
	TC	0x0160	~	0x017F	R/W	Target cropping setting of ch0
	MD	0x0180	~	0x0197	R/W	Motion Detection setting of ch0
	DMA	0x0198	~	0x01BF	R/W	DMA function setting of ch0
	LL	0x01C0	~	0x01CF	R/W	Link-list setting of ch0
1	Sub-channel	0x01D8	~	0x01FF	R/W	Sub-channel setting of ch1
	FCS	0x0200	~	0x0207	R/W	False color suppression setting of ch1
	Mask	0x0208	~	0x0247	R/W	Mask function setting of ch1
	DN	0x0248	~	0x028F	R/W	De-noise/Scaling setting of ch1
	OSD	0x0290	~	0x0337	R/W	OSD/Mark function setting of ch1
	TC	0x0338	~	0x0357	R/W	Target cropping setting of ch1
	MD	0x0358	~	0x036F	R/W	Motion Detection setting of ch1
	DMA	0x0370	~	0x0397	R/W	DMA function setting of ch1
	LL	0x0398	~	0x03A7	R/W	Link-list setting of ch1
2	Sub-channel	0x03B0	~	0x03D7	R/W	Sub-channel setting of ch2
	FCS	0x03D8	~	0x03DF	R/W	False color suppression setting of ch2
	Mask	0x03E0	~	0x041F	R/W	Mask function setting of ch2
	DN	0x0420	~	0x0467	R/W	De-noise/Scaling setting of ch2
	OSD	0x0468	~	0x050F	R/W	OSD/Mark function setting of ch2
	TC	0x0510	~	0x052F	R/W	Target cropping setting of ch2
	MD	0x0530	~	0x0547	R/W	Motion Detection setting of ch2
	DMA	0x0548	~	0x056F	R/W	DMA function setting of ch2
	LL	0x0570	~	0x057F	R/W	Link-list setting of ch2

Channel	Symbol	Offset			Access	Description
3	Sub-channel	0x0588	~	0x05AF	R/W	Sub-channel setting of ch3
	FCS	0x05B0	~	0x05B7	R/W	False color suppression setting of ch3
	Mask	0x05B8	~	0x05F7	R/W	Mask function setting of ch3
	DN	0x05F8	~	0x063F	R/W	De-noise/Scaling setting of ch3
	OSD	0x0640	~	0x06E7	R/W	OSD/Mark function setting of ch3
	TC	0x06E8	~	0x0707	R/W	Target cropping setting of ch3
	MD	0x0708	~	0x071F	R/W	Motion Detection setting of ch3
	DMA	0x0720	~	0x0747	R/W	DMA function setting of ch3
	LL	0x0748	~	0x0757	R/W	Link-list setting of ch3
4	Sub-channel	0x0760	~	0x0787	R/W	Sub-channel setting of ch4
	FCS	0x0788	~	0x078F	R/W	False color suppression setting of ch4
	Mask	0x0790	~	0x07CF	R/W	Mask function setting of ch4
	DN	0x07D0	~	0x0817	R/W	De-noise/Scaling setting of ch4
	OSD	0x0818	~	0x08BF	R/W	OSD/Mark function setting of ch4
	TC	0x08C0	~	0x08DF	R/W	Target cropping setting of ch4
	MD	0x08E0	~	0x08F7	R/W	Motion Detection setting of ch4
	DMA	0x08F8	~	0x091F	R/W	DMA function setting of ch4
	LL	0x0920	~	0x092F	R/W	Link-list setting of ch4
VI		0x5000		0x505F	R/W	Video input setting
SC		0x5100	~	0x5207	R/W	Source cropping setting
Palette		0x5300	~	0x533F	R/W	OSD palette setting
Global		0x5400	~	0x547F	R/W	VCAP global setting
Status		0x5500	~	0x55FF	R/W	VCAP status
OSD Font		0x10000	~	0x1FFFF	R/W	OSD font memory
OSD Mark		0x20000	~	0x23FFF	R/W	OSD mark memory
OSD display		0x30000	~	0x3FFFF	R/W	OSD display memory
Link-List		0x40000	~	0x47FFF	R/W	Link-list memory

7.3 CSIRX (MIPI) Controller

7.3.1 General Description

CSIRX is a high-speed and high-resolution interconnection for the CSI receiver. It is compliant with the Camera Serial Interface (CSI-2) [MIPI02] and supports D-PHY[MIPI01]. It provides the Display Pixel Interface (DPI) for the image processor. The supported data rate is up to 1.0Gbps per lane and is scalable from one to two data lanes. The maximum throughput will be 2.0Gbps when four data lanes are active in the High-Speed RX (HS-RX) mode. Various resolutions and pixel formats are supported.

7.3.2 Features

7.3.2.1 Specifications

- Compliant with MIPI Alliance Specification for Camera Serial Interface (CSI-2), Version 1.01
- Compliant with MIPI Alliance Standard for Display Pixel Interface (DPI-2), Version 2.00
- Compliant with MIPI Alliance Specification for D-PHY, Version 1.00.00
- Implements Logical PHY-Protocol Interface (PPI) according D-PHY, Version 1.00.00 Annex A

7.3.2.2 D-PHY Interface

- Supports up to two data lanes
- Supports High-Speed RX (HS-RX) and Low-Power RX (LP-RX)
- Supports ULPS and Receive Trigger event in Escape mode
- Accepts D-PHY without EoT processing
- Supports PPI HS-RX byte clock (RxByteClkHS) stopping after next clock of last valid data byte

7.3.2.3 CSI Features

- Supports horizontal line interleaving stream among virtual channels
- Supports maximum pixel of the horizontal line of 2048pixels for RAW format
- Supports data formats: RAW8, RAW10, RAW12, RAW14, Generic 8bit Long Packet Data Types,

Generic Short Packet Data Types, and User-Defined Data Types

- Supports error logging
- Supports Rx trigger messages
- Supports ECC and CRC checking
- Supports CRC checking turn-on/off
- Supports HS RX timeout detection
- Supports data type interleaving

7.3.2.4 DPI Features

- Programmable DPI Sync Pulse Width
- Provides pixel FIFO threshold control

7.3.2.5 References

- [MIPI01] MIPI Alliance Specification for D-PHY, Version 1.00.00
- [MIPI02] MIPI Alliance Specification for Camera Serial Interface (CSI-2), Version 1.01
- [MIPI03] MIPI Alliance Standard for Display Pixel Interface (DPI-2), Version 2.00

7.3.3 Programming Model

Table 7-2. Definition of Register Abbreviation

Register Type	Description
RO	Read-only: Register bits are read-only and cannot be altered by software. Register bits are initialized by hardware mechanisms, such as the pin strapping, hardware configuration, or serial EEPROM.
RW	Read-Write: Register bits are read-write and can be set or cleared by software to the desired state.
RW1C	Write-1-to-clear status: Register bits indicate status when read, a set bit indicating a status event may be cleared by writing a ‘1’. Writing a ‘0’ to RW1C bits has no effect.
RW1S	Write-1-to-set status: Register bits indicate status when read. Writing a ‘0’ to RW1S bits has no effect.

Register Type	Description
Rsvd	Reserved: Reserved for future RO implementations. Registers or memory that shall be treated as read-only by the system software. The Rsvd registers shall return to '0' when read. Software shall ignore the value read from these bits. Default value: 0

Besides the status register bits with the RW1C attributes and the Interrupt Enable Register (0x3C), after programming all registers, CR.SR (Offset = 0x04) shall be set to re-initial CSIRX.

Table 7-3. CSIRX Registers

Offset (Hex)	Name	Size (Byte)	Type	Description	Default Value (Hex)
0x00	VIDR	1	RO	Vendor ID Register	0x8b
0x01	DIDR	1	RO	Device ID Register	0x01
0x04	CR	1	RO, RW, RW1S	Control Register	0x0c
0x05	VSCR	1	RO, RW	DPI V Sync Control Register	0
0x06	ECR	1	RO, RW	Extended Control Register	0
0x08	TCNR	2	RO, RW	Timer Count Number Register	0x64
0x0A	HRTVR	1	RW	HS RX Timeout Value Register	0x64
0x12	ITR	2	RO, RW	Initialization Timer Register	0x20
0x14	VSTR0	1	RW	DPI VC0 V Sync Timing Register	0x02
0x15	HSTR0	1	RW	DPI VC0 H Sync Timing Register	0x08
0x1C	MCR	1	RO, RW	DPI Mapping Control Register	0
0x1E	VSTER	1	RO, RW	DPI V Sync Timing Extended Register	0
0x20	HPNR	2	RO, RW	DPI Horizontal Pixel Number Register	0
0x28	PECR	1	RO, RW	PPI Enable Control Register	0x0
0x2A	DLMR	1	RO, RW	Data Lane Mapping Register	0x10
0x30	CSIERR	2	RO	CSI Error Report Register	0
0x33	INTSTS	1	RO, RW1C	Interrupt Status register	0

Offset (Hex)	Name	Size (Byte)	Type	Description	Default Value (Hex)
0x34	ESR	2	RO	Escape Mode and Stop State Register	0x400f
0x38	DPISR	1	RO, RW1C	DPI Status Register	0
0x3C	INTER	1	RW	Interrupt Enable Register	0x41
0x40	FFR	1	RO	Feature Register	0x54
0x48	DPCMR	1	RO, RW	DPCM Register	0x39
0x4C	FRR	4	RO	Revision Register	0x00010400
0x50	PFTR	1	RO, RW	Pixel FIFO Threshold Register	0
0x80	FRCR	2	RO, RW	Frame Rate Control Register	0xff00
0x88	FNR	2	RO	Frame Number Register	0x0000
0x90	BPGR	2	RO, RW	DPI Built-in Pattern Generator Register	0x0400

7.3.3.1 Vender ID Register (VIDR, Offset = 0x00)

Table 7-4. Vendor ID Register (VIDR, Offset = 0x00)

Bit	Name	Type	Default Value	Description
[7:0]	VID	RO	0x8B	Vendor ID This register identifies the manufacture of a device.

7.3.3.2 Device ID Register (DIDR, Offset = 0x01)

Table 7-5. Device ID Register (DIDR, Offset = 0x01)

Bit	Name	Type	Default Value	Description
[7:0]	DID	RO	0x1	Device ID This register identifies a particular device.

7.3.3.3 Control Register (CR, Offset = 0x04)

Table 7-6. Control Register (CR, Offset = 0x04)

Bit	Name	Type	Default Value	Description
[7:6]	-	RO	0x0	Reserved
5	HEG	RW	0x0	Hsync Enable for Generic Long and User Defined Packets When this bit is 1'b1: dpi_vc0_Hsync can be asserted to receive the virtual channel#0 generic long or user-defined packets. 1'b0: dpi_vc0_Hsync cannot be asserted to receive the virtual channel#0 generic long or user-defined packets.
4	EAPBL	RW	0x0	Enable Auto-Padding for Broken Line When this bit is 1'b1: When the line with the image stream is broken (De-assert dpi_vc0_DE), CSIRX will treat this event as the exception and the pad will automatically be the pixels up to the number defined in the corresponding DPI VC0 Horizontal Pixel Number Register (HPNR0, Offset = 0x20). 1'b0: Auto-padding is disabled.
3	ECCCE	RW	0x1	ECC Checking Enable When this bit is set, CSIRX can report the ECC single bit error (CSIERR1.ECCES) and multiple-bit error (CSIERR1.ECCEM). Otherwise, the reporting is disabled.
2	CRCCE	RW	0x1	CRC Checking Enable This bit controls CRC checking. When this bit is set, CSIRX will calculate CRC (checksum) and check the result with the checksum of the received packet. If the CRC error is found, it will reflect CSIERR1.CE. When this bit is cleared, it will not check the result.
1	TME	RW	0x0	Test Mode Enable This test mode is only for the internal usage.
0	SR	RW1S	0x0	Software Reset Writing '1' to this bit will trigger CSIRX to perform software reset. This bit is still read as '0' after writing '1' to it. Setting this bit is not effective when the clock of the corresponding clock domain is not valid. For example, to clear the pixel FIFO of the virtual channel0, the dsi_clk and vc0_PCLK clocks shall be valid.

7.3.3.4 DPI VSync Control Register (VSCR, Offset = 0x05)

This register determines the de-assertion condition and the time unit of pulse width of the dpi_vc0_Vsync signal. CSIRX asserts dpi_vc0_Vsync (This signal is low active) for receiving VC0 Frame Start short packet.

For the de-assertion condition:
When the VSPC0 bit of this register is set, CSIRX de-asserts this signal after receiving VC0 Frame End.

When this bit is clear, the low-active pulse width of DPI VC0 VSync Timing Register, DPI VSync Timing Extended Register and VSTU0 bit of this register. That is,

VC0 VSync width = {VSTER0.VC*0*[3:0], VSTR0.W[7:0]} or
4096 when both VSTER0.VC0 and VSTR0 are '0'.

In the above expression, the time unit is "pixel clock" for VSCR.VSTU0: 1 and the time unit is "horizontal line" for VSCR.VSTU0: 0. When the time unit is "horizontal line", the pulse width counter increases by each DPI VC0 HSync de-assertion, including the supported image data packets, user-defined byte-based packet, and generic 8bit long packet with the embedded 8bit non-image data (Packet type = 0x12).

Table 7-7. DPI VSync Control Register (VSCR, Offset = 0x05)

Bit	Name	Type	Default Value	Description
[7:5]	-	RO	0x0	Reserved
4	VSPC0	RW	0x0	VSync Pulse Control for VC0 This bit determines Vsync signal behavior. Please refer to the above description.
[3:1]	-	RO	0x0	Reserved
0	VSTU0	RW	0x0	VSync Time Unit for VC0 This bit determines the time unit of Vsync signal width. When this bit is set, the time unit is the pixel clock (vc0_PCLK). Otherwise, the time unit is the horizontal line.

7.3.3.5 Extended Control Register (ECR, Offset = 0x06)

This register is only for the internal usage.

Table 7-8. Extended Control Register (ECR, Offset = 0x06)

Bit	Name	Type	Default Value	Description
[7:2]	-	RO	0x0	Reserved
1	SF	RW	0	Software Flag
0	PCE	RW	0	PPI Data Checking Enable This bit is the software enable bit for PPI Data Checking. When this bit is 0: PPI Interface Data Checking is not enabled by software. 1: PPI Interface Data Checking is enabled by software. The normal operation is disabled.

7.3.3.6 Timer Count Number Register (TCNR, Offset = 0x08 ~ 0x09)

This register determines the count number of the clock (csi_clk) for 1μs. The default value of the timer count number is 0x64 (= 100), which means that CSIRX counts 100 clocks (csi_clk) as 1μs. When the value is set as '0', this timer will be disabled and the mechanisms which count μs or ms will not work at this situation.

Table 7-9. Timer Count Number Lower Register (TCNLR, Offset = 0x08)

Bit	Name	Type	Default Value	Description
[7:0]	TCN	RW	0x64	Timer Count Number This field is the lower bits (Bit 7 to bit 0) of the timer count number for 1μs.

Table 7-10. Timer Count Number Higher Register (TCNHR, Offset = 0x09)

Bit	Name	Type	Default Value	Description
[7:2]	-	RO	0x0	Reserved
[1:0]	TCN	RW	0x0	Timer Count Number This field is the higher bits (Bit 9 to bit 8) of the timer count number for 1μs.

7.3.3.7 HS RX Timeout Value Register (HRTVR, Offset = 0x0A)

The timeout value of HRX_TO timer is (Value of this register) * 128μs. When the value of this field is zero, this timer is disabled and never expires. The default value is 0x64 (The decimal is 100.) so the default of the timeout value is 128 * 100μs = 12.8ms.

Table 7-11. HS RX Timeout Value Register (HRTVR, Offset = 0x0A)

Bit	Name	Type	Default Value	Description
[7:0]	TV	RW	0x64	Timeout Value This value of this register indicates the timeout value of HRX_TO timer.

7.3.3.8 Initialization Timer Register (ITR, Offset = 0x12 ~ 0x13)

Both registers (ITHR and ITLR) determine the period of asserting the PPI signal, ForceRxmode, after the power-on reset, system reset, or writing '1' to the CR.SR bit. The period is {ITHR.IT[1:0], ITLR.IT[7:0]} * 1μs. Here "1μs" length is defined in TCNR (Offset = 0x08 ~ 0x09). The initialization timer counts for this period of the PPI Stopstate signal continuously keeping as high. For example, if {ITHR.IT[1:0], ITLR.IT[7:0]} = 10'h3_FF, ForceRxmode will be asserted at power-on-reset or system reset being active and de-asserted after the Stopstate signal continuously keeping as high for 1023μs.

For the case {ITHR.IT[1:0], ITLR.IT[7:0]} = zero, the PPI signal ForceRxmode behaves as:

a. CSIRX still asserts ForceRxmode 1~3 csi_clk clocks after both power-on reset and system reset being released. The ForceRxmode pulse variation depends on the timing relationship between csi_clk and power-on reset and system reset.
b. ForceRxmode will not be asserted by writing '1' to the CR.SR bit.

Table 7-12. Initialization Timer Lower Register (ITLR, Offset = 0x12)

Bit	Name	Type	Default Value	Description
[7:0]	IT	RW	0x20	Initialization Timer Value This field is the lower bits (Bit 7 to bit 0) of the initialization timer value.

Table 7-13. Initialization Timer Higher Register (ITHR, Offset = 0x13)

Bit	Name	Type	Default Value	Description
[7:2]	-	RO	0x0	Reserved
[1:0]	IT	RW	0x0	Initialization Timer Value This field is the higher bits (Bit 9 to bit 8) of the initialization timer value.

7.3.3.9 DPI VSync Timing Register (VSTR, Offset = 0x14)

When the VSCR.VSPC0 register bit (Offset = 0x05) is cleared, the DPI VC0 VSync Timing Register can determine the VSync pulse width of VC0 at DPI.

Table 7-14. DPI VC0 VSync Timing Register (VSTR0, Offset = 0x14)

Bit	Name	Type	Default Value	Description
[7:0]	W	RW	0x02	VSync Width VC0 VSync width = {VSTER0.VC0[3:0], VSTR0.W[7:0]} The unit is “Line”, which is defined in the DPI specification [MIPI03]. VSTER0 register locates at the address 0x1E. The default value is 0x002, which means that the width of the DPI Vsync signal is 2 lines. If the value of {VSTER0.VC0[3:0], VSTR0.W [7:0]} is ‘0’, the width will be 4096 lines.

7.3.3.10 DPI HSync Timing Register (HSTR, Offset = 0x15)

The DPI VC0 HSync Timing Register determines the HSync pulse width of VC0 at DPI.

Table 7-15. DPI VC0 HSync Timing Register (HSTR0, Offset = 0x15)

Bit	Name	Type	Default Value	Description
[7:0]	W	RW	0x08	HSync Width VC0 HSync width = HSTR0.W[7:0] The unit is vcn_PCLK (DPI pixel clock). The default value is 0x08, which means that the width of the DPI Hsync signal is eight vc0_PCLK clock cycles. If this register is ‘0’, the width will be 256 vc0_PCLK clock cycles.

7.3.3.11 DPI Mapping Control Register (MCR, Offset = 0x1C)

Table 7-16. DPI Mapping Control Register (MCR, Offset = 0x1C)

Bit	Name	Type	Default Value	Description
[7:1]	-	RO	0x0	Reserved
0	M0	RW	0x0	Mapping Control for VC0 When this bit is set, the VC0 pixel outputs for all RAW types are aligned to dpi_vc0_D[13] (MSB). The pixel output for RAW8 is dpi_vc0_D[13:6] and for RAW10 is dpi_vc0_D[13:4]. When this bit is '0', the VC0 pixel outputs for all RAW types are aligned to dpi_vc0_D[0] (LSB). The pixel output for RAW8 is dpi_vc0_D[7:0 and for RAW10 is dpi_vc0_D[9:0].

7.3.3.12 DPI VSync Timing Extended Register (VSTER, Offset = 0x1E)

This register is used to extend the active pulse width of the DPI VSync signals.

Table 7-17. DPI VSync Timing Extended Register 0 (VSTER0, Offset = 0x1E)

Bit	Name	Type	Default Value	Description
[7:4]	-	RO	0x0	Reserved
[3:0]	VC0	R/W	0x0	VC0 Extended Timing It is used to extend the width of Vsync and effective only when the VSCR.VSPC0 register bit (Address = 0x05) is cleared.

7.3.3.13 DPI VC0 Horizontal Pixel Number Register (HPNR, Offset = 0x20)

HPNR0 contains the HPNLR0 (Offset = 0x20) and HPNHR0 (Offset = 0x21) registers. These registers determine the number of the VC0 DPI horizontal pixels. When the value is '0', it means that the number of the DPI horizontal pixels is 8192.When CR.EAPBL (Offset = 0x04) is set and the line with the image stream is broken, CSIRX will pad the pixels up to the number defined in the corresponding pixel number. Therefore, if CR.EAPBL is set, DPI DE will be de-asserted during transmitting the horizontal line and will be treated as an exception and padding automatically starts.

The pixel number of the received line shall consist with the corresponding pixel number setting; otherwise, the residue pixels are not sent to DPI, the pixel number exceeding (DPISR0.PNE) and discontinuous data enable (DPISR0.DDE) flags are not effective.

Table 7-18. VC0 Horizontal Pixel Number Lower Register (HPNLR0, Offset = 0x20)

Bit	Name	Type	Default Value	Description
[7:0]	HPN	RW	0x0	Horizontal Pixel Number This field is the lower bits (Bit 7 to bit 0) of the DPI horizontal pixel number.

Table 7-19. VC0 Horizontal Pixel Number Higher Register (HPNHR0, Offset = 0x21)

Bit	Name	Type	Default Value	Description
7	-	RO	0x0	Reserved
6	VP	RW	0x0	Vsync Polarity 0: Vsync is low active. 1: Vsync is high active.
5	HP	RW	0x0	Hsync Polarity 0: Hsync is low active. 1: Hsync is high active.
[4:0]	HPN	RW	0x0	Horizontal Pixel Number This field is the higher bits (Bit 12 to bit 8) of the DPI horizontal pixel number.

7.3.3.14 PPI Enable Control Register (PECR, Offset = 0x28)

Table 7-20. PPI Enable Control Register (PECR, Offset = 0x28)

Bit	Name	Type	Default Value	Description
[7:4]	-	RO	0x0	Reserved
[3:0]	PEC	RW	0x0	PPI enable control

7.3.3.15 Data Lane Mapping Register (DLMR, Offset = 0x2A)

These registers (DLMR0 and DLMR1) determine the logic data lanes mapping to the physical data lanes. DLMR0 register contains the data lane#0 and #1 mapping.

By default, the logical data lane#***n*** maps to the physical data lane#***n***. If DLMR0.L0: 1 and the hardmacro, it means that the logical data lane#0 maps to the physical data lane#1, that is, CSIRX treats the physical data lane#1 as the data lane#0 in the internal logic design. This register cannot be dynamically changed during normal operating.

Table 7-21. Data Lane Mapping Register 0 (DLMR0, Offset = 0x2A)

Bit	Name	Type	Default Value	Description
[7:6]	-	RO	0	Reserved
[5:4]	L1	RW	1	Logical Data Lane#1 Mapping This field indicates the physical data lane number mapping to the logical data lane#1.
[3:2]	-	RO	0	Reserved
[1:0]	L0	RW	0	Logical Data Lane#0 Mapping This field indicates the physical data lane number mapping to the logical data lane#0.

7.3.3.16 CSI Error Report Register (CSIERR, Offset = 0x30 ~ 0x31)

In addition to the CSIERR0.PCR bit (Offset = 0x30, bit#4), the bits of this register are the error report bits. CSIRX can log the error events in the registers, CSIERR0 and CSIERR1. These bits (Except for the CSIERR0.PCR bit) are cleared after write-one-clear to the INTSTS.CSI bit (Offset = 0x33) or write-one to the CR.SR bit (Offset = 0x04).

Table 7-22. CSI Error Report Register 0 (CSIERR0, Offset = 0x30)

Bit	Name	Type	Default Value	Description
7	-	RO	0x0	Reserved
6	FCE	RO	0x0	False Control Error When this bit is set, it indicates either of ErrControl[n] has ever asserted. To detect this event, the high pulse of ErrControl[n] shall be longer than two csi_clk clock cycles and csi_clk clock shall exist when the event occurs. ErrControl[n] is the PPI signal, ErrControl, for data lane n. If the corresponding data lane n doesn't exist, ErrControl[n] is treated as low.
5	HSRT	RO	0x0	High-Speed Receiver Timeout When this bit is set, it indicates that "HS RX Timeout" (HRX-TO) has occurred. To detect this event, the csi_clk clock shall be available for the timeout counter.
4	PCR	RO	0x0	PPI Checking Result This is the status bit of the internal test mode.

Bit	Name	Type	Default Value	Description
3	EMECE	RO	0x0	Escape Mode Entry Command Error When this bit is set, it indicates either of ErrEsc[n] has ever asserted. To detect this event, the high pulse of ErrControl[n] shall be longer than two csi_clk clock cycles and csi_clk clock shall exist when the event occurs. ErrEsc[n] is the PPI signal ErrEsc for the data lane n. If the corresponding data lane n doesn't exist, ErrEsc[n] is treated as low.
2	ESE	RO	0x0	EoT Sync Error When this bit is set, it indicates either of ErrEotSyncHS[n] has ever asserted. To detect this event, both csi_clk and RxByteClkHS clocks shall exist until logging this event in this bit. ErrEotSyncHS[n] is the input signal ErrEotSyncHS for the data lane n. If the corresponding lane n doesn't exist, ErrEotSyncHS [n] is treated as low.
1	SSE	RO	0x0	SoT Sync Error When this bit is set, it indicates either of ErrSotSyncHS[n] has ever asserted. To detect this event, both csi_clk and RxByteClkHS clocks shall exist until logging this event in this bit. ErrSotSyncHS[n] is the PPI signal ErrSotSyncHS for the data lane n. If the corresponding lane n doesn't exist, ErrSotSyncHS[n] is treated as low.
0	SE	RO	0x0	SoT Error When this bit is set, it indicates either of ErrSotHS[n] has ever asserted. To detect this event, both csi_clk and RxByteClkHS clocks shall exist until logging this event in this bit. ErrSotHS[n] is the PPI signal ErrSotHS for the data lane n. If the corresponding lane n doesn't exist, ErrSotHS[n] is treated as low.

Table 7-23. CSI Error Report Register 1 (CSIERR1, Offset = 0x31)

Bit	Name	Type	Default Value	Description
[7:6]	-	RO	0x0	Reserved
5	ITL	RO	0x0	Invalid Transmission Length
4	CSIVIDI	RO	0x0	CSI VC ID Invalid
3	UPT	RO	0x0	Unsupported Packet Type
2	CE	RO	0x0	Checksum Error (Long packet only)
1	ECCEM	RO	0x0	ECC Error, multi-bit (Detected but not corrected)

Bit	Name	Type	Default Value	Description
0	ECCES	RO	0x0	ECC Error, single-bit (Detected and corrected)

7.3.3.17 Interrupt Status Register (INTSTS, Offset = 0x33)

Table 7-24. Interrupt Status Register (INTSTS, Offset = 0x33)

Bit	Name	Type	Default Value	Description
7	-	RO	0x0	Reserved
6	RT	RW1C	0x0	Receive Trigger This bit is set when CSIRX receives the trigger events from the PPI signal, RxTriggerEsc. The bus value of RxTriggerEsc can be found in the register (ESR1, Offset = 0x35).
5	ULPS_E	RW1C	0x0	ULPS Entry This bit is set when either the enabled data lanes or clock lane enters ULPS. This event is determined by detecting the assertion of UlpsActiveNot and ClkUlpsActiveNot. Here, the enabled data lanes means the data lanes with the corresponding ppi_Enable bit being active.
4	ULPS_X	RW1C	0x0	ULPS Exit This bit is set when all of the data lanes and clock lane exit from ULPS. This event is determined by detecting the de-assertion of UlpsActiveNot and ClkUlpsActiveNot. Here, the enabled data lanes means the data lanes with the corresponding ppi_Enable bit being active.
3	FS	RW1C	0x0	Receive Frame Start Packet
2	DPI	RW1C	0x0	DPI Events This bit is set when either bit of the register DPISR0 (Offset = 0x38) is asserted. Please note that writing one to clear this bit can also clear the register, DPISR0.
1	-	RO	0x0	Reserved
0	CSI	RW1C	0x0	CSI Error Events This bit is set when the CSI error event occurs. The CSI error events are reported in the registers, CSIERR0 (Offset = 0x30) and CSIERR1 (Offset = 0x31). Please note that write one to clear this bit can also clear the CSI Error Report Registers 0 and 1 (CSIERR0 and CSIERR1).

7.3.3.18 Escape Mode and Stop State Register (ESR, Offset = 0x34 ~ 0x35)

These registers are used to record the PPI signals about the escape mode and stop state so the software can reach these signals state at PPI through APB interface.

Table 7-25. Escape Mode and Stop State Register 0 (ESR0, Offset = 0x34)

Bit	Name	Type	Default Value	Description
[7:4]	STOP	RO	0x0	Data Lane Stopstate This field reflects the input PPI signals Stopstate, for the data lanes. Bit 4 indicates the PPI signal, Stopstate, for the data lane#0; bit 5 indicates the PPI signal, Stopstate, for the data lane#1; bit 6 indicates the PPI signal, Stopstate, for the data lane#2; bit 7 indicates the PPI signal, Stopstate, for the data lane#3.
[3:0]	ULPS	RO	0xf	Data Lane UlpsActiveNot This field reflects the input PPI signals, UlpsActiveNot, for the data lanes. Bit 0 indicates the PPI signal, UlpsActiveNot, for the data lane#0; bit 1 indicates the PPI signal, UlpsActiveNot, for the data lane#1; bit 2 indicates the PPI signal, UlpsActiveNot, for the data lane#2; and bit 3 indicates the PPI signal, UlpsActiveNot, for the data lane#3.

Table 7-26. Escape Mode and Stop State Register 1 (ESR1, Offset = 0x35)

Bit	Name	Type	Default Value	Description
7	CSTOP	RO	0x0	Clock Lane Stopstate This bit reflects the PPI signal, ClkStopState.
6	CUAN	RO	0x1	Clock Lane ULP State (Not) Active The bit reflects the PPI signal, ClkUlpsActiveNot.
5	RUE	RO	0x0	Data Lane Escape Ultra-Low Power (Receive) Mode This bit reflects the PPI signal, RxUlpsEsc, for data lane#0.
4	RUCN	RO	0x0	Clock Lane Receive Ultra-Low Power State This bit reflects the PPI signal, RxUlpsClkNot.
[3:0]	RTE	RO	0x0	Receive Trigger Event The value of this field is the bus content of the PPI signal, RxTriggerEsc, for data lane#0.

7.3.3.19 DPI Status Register (DPISR, Offset = 0x38)

Table 7-27. DPI VC0 Status Register (DPISR0, Offset = 0x38)

Bit	Name	Type	Default Value	Description
[7:6]	-	RO	0x0	Reserved
5	DPI_OF	RW1C	0x0	DPI FIFO Overflow When this bit is set, it indicates that DPI VC0 FIFO is overflow.
4	DEOV	RW1C	0x0	Data Enable Overlaps Vsync Signal When this bit is set, it indicates that DPI Data Enable overlaps the Vertical Synchronization Timing at DPI. Otherwise, this event will not occur.
3	DEOH	RW1C	0x0	Data Enable Overlaps Hsync Signal When this bit is set, it indicates that DPI Data Enable overlaps the Horizontal Synchronization Timing signal at DPI. Otherwise, this event will not occur.
2	PNE	RW1C	0x0	Pixel Number Exceeding When this bit is set, it indicates that the pixel number of the VC0 received horizontal line exceeds the setting of the Horizontal Pixel Number Register (HPNR0). Otherwise, this bit will not be set. This bit will only be effective when the value of the HPNR0 register is correct.
1	-	RO	0x0	Reserved
0	DDE	RW1C	0x0	Discontinuous Data Enable When this bit is set, it indicates that the discontinuous DPI Data Enable (DE) occurs during the pixel data (Excluding generic 8bit long packets, user-defined 8bit data packets, and short packets) transmission of a horizontal line at DPI. Otherwise, this event will not occur. This bit will only be effective when the value of the Horizontal Pixel Number Register (HPNR0) is correct.

7.3.3.20 Interrupt Enable Register (INTER, Offset = 0x3C)

This register controls the interrupt enable.

Table 7-28. Interrupt Enable Register (INTER, Offset = 0x3C)

Bit	Name	Type	Default Value	Description
7	ACDF	RW	0x0	Automatically Clear DPI FIFO This it is not the interrupt enable bit. When this bit is set and VC0 frame start packet is received, CSIRX will clear the VC0 DPI FIFO. Otherwise, receiving VC0 frame start packet won't clear VC0 DPI FIFO.eserved
6	RT	RW	0x1	Receive Trigger This is the interrupt enable bit for receiving the trigger event at the data lane#0 (INTSTS.RT). When this bit is set, the interrupt of this event will be enabled. Otherwise, it will be disabled.
5	ULPS_E	RW	0x0	ULPS Entry This bit is the interrupt enable bit for the event of ULPS entry (INTSTS.ULPS_E). When this bit is set, the interrupt of this event will be enabled. Otherwise, it will be disabled.
4	ULPS_X	RW	0x0	ULPS Exit This bit is the interrupt enable bit for the event of ULPS exit (INTSTS.ULPS_X). When this bit is set, the interrupt of this event will be enabled. Otherwise, it will be disabled.
3	-	RO	0x0	Reserved
2	DPI	RW	0x0	DPI Events This bit is the interrupt enable bit of DPI events (INTSTS.DPI). When this bit is set, the interrupt of this event will be enabled. Otherwise, it will be disabled.
1	DEM	RW	0x0	DPI Event Mask This it is not the interrupt enable bit. This bit effects the register bit INTSTS.DPI (Address = 0x33). When this bit is set, INTSTS.DPI asserts only when DPISR0.DPI_OF (Address = 0x38) is set. Otherwise, INTSTS.DPI can assert when either bit of the register DPISR0 is set.eserved

Bit	Name	Type	Default Value	Description
0	CSI	RW	0x1	CSI Error Events This bit is the interrupt enable bit of CSI error events (INTSTS.CSI). When this bit is set, the interrupt of this event will be enabled. Otherwise, it will be disabled.

7.3.3.21 CSIRX Feature Register (FFR, Offset = 0x40)

Table 7-29. CSIRX Feature Register 0 (FFR0, Offset = 0x40)

Bit	Name	Type	Default Value	Description
[7:4]	DFS	RO	0x5	DPI FIFO Size The total DPI FIFO size is 2^5x30x2 bits.
[3:2]	LN	RO	0x1	Lane Number The encoding of this field: 2'b01: Two data lanes
[1:0]	VCN	RO	0x0	Virtual Channel Number

7.3.3.22 DPCM Register (DPCMR, Offset = 0x48)

Only DPCM 10-8-10 and 12-8-12 decode schemes with the predictor1 are supported.

CSIRX doesn't support padding the pixels (Please refer to CR.EAPBL, Offset = 0x04) and detecting the pixel number exceeding (Please refer to DPISR0.PNE, Offset = 0x38) for receiving DPCM packets.

Table 7-30. VC0 DPCM Register (DPCMR0, Offset = 0x48)

Bit	Name	Type	Default Value	Description
7	DE	RW	0	DPCM Enable When this bit is: 1: Enable the DPCM decoder. CSIRX will decode DPCM data and transmit the pixel at DPI. If DPCMR.DDSS = '0', CSIRX will decode DPCM as RAW10 pixel and transmit the pixel at DPI. If DPCMR.DDSS = '1', CSIRX will decode DPCM as RAW12 pixel and transmit the pixel at DPI. 0: Disable the DPCM decoder

Bit	Name	Type	Default Value	Description
6	DDSS	RW	0	DPCM Decode Scheme Selection When this bit is: 0: 10-8-10 scheme is selected. 1:12-8-12 scheme is selected.
[5:0]	DPT	RW	6'h39	DPCM Packet Type This field indicates the data type is treated as the DPCM packet.

7.3.3.23 CSIRX Revision Register (FRR, Offset = 0x4C ~ 0x4F)

This register is read-only.

Table 7-31. CSIRX Revision Register (FRR, Offset = 0x4C)

Bit	Name	Type	Default Value	Description
[31:0]	Revision	RO	32'h00010400	CSIRX_REVISION

7.3.3.24 Pixel FIFO Threshold Register (PFTR, Offset = 0x50)

The pixel FIFO threshold is used to control the assertion of the DPI Data Enable signal for the first pixel of each horizontal line. Each virtual channel owns the individual pixel FIFO threshold.

The pixel FIFO threshold = (Value of the corresponding Pixel FIFO Threshold Register * 4) pixels

For example, when the Pixel FIFO Threshold Register 0 (PFTR0) = 8'h5, the VC0 pixel FIFO threshold is 5 * 4 = 20 pixels. In this case, the corresponding VC (VC0) DPI Data Enable signal will assert to transmit the first pixel once the pixel FIFO contains at least 20 pixels.

By default, there is no threshold for the DPI Data Enable signal. Consequently, DPI Data Enable will assert to transmit the first pixel once the FIFO contains at least one pixel and Hsync signal also asserts at the same clock. Please note that the threshold cannot exceed the entry number of its pixel FIFO.

Table 7-32. Pixel FIFO Threshold Register (PFTR0, Offset = 0x50)

Bit	Name	Type	Default Value	Description
[7:0]	V	RW	0x0	Pixel FIFO Threshold Setting The VC0 pixel FIFO threshold is (Value of this register * 4) pixels.

7.3.3.25 CSIRX PHY Control Registers (PHYCTRL0, Offset = 0x60 ~ 0x6F)

Table 7-33. CSIRX PHY Control Register 0a (PHYCTRL0a, Offset = 0x60)

Bit	Name	Type	Default Value	Description
[7:5]	-	RW	0x0	Reserved
4	ClkLnEn	RW	0x0	Enable the clock lane module 0: Disable 1: Enable
3	ParallelEn	RW	0x1	Enable the parallel interface pin share 0: Disable 1: Enable
2	ISO_EN	RW	0x1	Enable the internal power isolation cell 0: Disable 1: Enable
1	ISO_PD	RW	0x1	Turn off VCC11K (Core power) for the digital portion 0: ON 1: OFF
0	PHY_EN	RW	0x0	Enable 0: Disable 1: Enable

Table 7-34. CSIRX PHY Control Register 0b (PHYCTRL0b, Offset = 0x61)

Bit	Name	Type	Default Value	Description
[7:0]	-	RW	0x0	Reserved

Table 7-35. CSIRX PHY Control Register 0c (PHYCTRL0c, Offset = 0x62)

Bit	Name	Type	Default Value	Description
[23:21]	-	RW	0x0	Reserved
[20:16]	ModeSel	RW	0x0	Selection for the operation mode of DPHY 0x0: MIPI mode 0x1: Reserved 0x6: subLVDS mode Others: Reserved

Table 7-36. CSIRX PHY Control Register 0d (PHYCTRL0d, Offset = 0x63)

Bit	Name	Type	Default Value	Description
[23:21]	-	RW	0x0	Reserved
[20:16]	PclkSel	RW	0x0	Select the desired clock level 0: Clock lane input 1: Clock lane input/2 2: Clock lane input/4 3: Clock lane input/8

Table 7-37. CSIRX PHY Control Register 1a (PHYCTRL1a, Offset = 0x64)

Bit	Name	Type	Default Value	Description
[7:0]	RxTinit_cnt[7:0]	RW	0x0	To set a counter value of lane initiation for RX The recommend value is 0xE2.

Table 7-38. CSIRX PHY Control Register 1b (PHYCTRL1b, Offset = 0x65)

Bit	Name	Type	Default Value	Description
7	-	RW	0x0	Reserved
[6:0]	RxTinit_cnt[14:8]	RW	0x0	To set a counter value of lane initiation for RX The recommend value is 0x0A.

Table 7-39. CSIRX PHY Control Register 1c (PHYCTRL1c, Offset = 0x66)

Bit	Name	Type	Default Value	Description
[7:0]	PatternCnt[7:0]	RW	0x0	To set a counter value of the TX/RX pattern length for BIST test The recommend value is 0x00.

Table 7-40. CSIRX PHY Control Register 1d (PHYCTRL1d, Offset = 0x67)

Bit	Name	Type	Default Value	Description
[7:0]	PatternCnt [15:8]	RW	0x0	To set a counter value of the TX/RX pattern length for BIST test The recommend value is 0x01.

Table 7-41. CSIRX PHY Control Register 2a (PHYCTRL2a, Offset = 0x68)

Bit	Name	Type	Default Value	Description
[7:0]	RxDataHsSettleCnt	RW	0x0	To set a counter value of HS settle for data RX lane The recommend value is 0x0C. Please refer to Table 7-38.

Table 7-42. RxDataHsSettleCnt

Suggested Value (HEX)	Max. Clock Range (MHz)	Min. Clock Range (MHz)	Min. Time (ns)	Max. Time (ns)
0x01	99	80	161.62	200
0x01	110	100	145.45	160
0x02	160	111	150	216.22
0x02	200	161	120	149.07
0x03	250	201	128	159.2
0x04	300	251	133.33	159.36
0x05	350	301	137.14	159.47
0x06	400	351	140	159.54
0x06	450	401	124.44	139.65
0x07	500	451	128	141.91
0x08	550	501	130.91	143.71
0x08	600	551	120	130.67
0x09	650	601	123.08	133.11
0x09	700	651	125.71	135.18
0x0A	750	701	117.33	125.53
0x0B	800	751	120	127.83
0x0C	850	801	122.35	129.84
0x0D	900	851	124.44	131.61
0x0D	950	901	117.89	124.31
0x0E	1000	951	120	126.18

Table 7-43. CSIRX PHY Control Register 2b (PHYCTRL2b, Offset = 0x69)

Bit	Name	Type	Default Value	Description
[7:0]	RxDataHsSyncCnt	RW	0x0	To set a counter value of HS SYNC for data RX lane. This is a time-out value, and the SYNC leader must be synchronized before time out. The recommended value is 0x30.

Table 7-44. CSIRX PHY Control Register 2c (PHYCTRL2c, Offset = 0x6A)

Bit	Name	Type	Default Value	Description
[7:4]	RxDfRctl	RW	0x0	Only for the analog performance adjustment The recommended value is 0x7.
3	-	RW	0x0	Reserved
[2:0]	RxClkHsSettleCnt	RW	0x0	To set a counter value of lane initiation for RX The recommended value is 0x0A.

Table 7-45. CSIRX PHY Control Register 2d (PHYCTRL2d, Offset = 0x6B)

Bit	Name	Type	Default Value	Description
[7:6]	RxTcdtrim	RW	0x0	Only for the analog performance adjustment The recommended value is 0x1
[5:4]	RxLpSel	RW	0x0	Only for the analog performance adjustment
3	RxSwapPdata	RW	0x0	For debugging purpose
2	RxSlvds	RW	0x0	Select subLVDS or MIPI b1: subLVDS b0: MIPI
1	RxPuNoOff	RW	0x0	Only for the analog performance adjustment The recommended value is 0x1.
0	RxDfLoad	RW	0x0	Only for the analog performance adjustment

Table 7-46. CSIRX PHY Control Register 3a (PHYCTRL3a, Offset = 0x6C)

Bit	Name	Type	Default Value	Description
7	-	RW	0x0	Reserved
[6:4]	RxTrimcd	RW	0x0	Only for the analog performance adjustment The recommend value is 0x4.
[3:0]	RxTrimbg	RW	0x0	Only for the analog performance adjustment The recommend value is 0x4.

Table 7-47. CSIRX PHY Control Register 3b (PHYCTRL3b, Offset = 0x6D)

Bit	Name	Type	Default Value	Description
[7:6]	-	RW	0x0	Reserved
[5:4]	RxRegEn	RW	0x0	Only for the analog performance adjustment The recommended value is 0x2.
3	-	RW	0x0	Reserved
[2:0]	XcfgRxTrim	RW	0x0	Only for the analog performance adjustment The recommended value is 0x3.

Table 7-48. CSIRX PHY Control Register 3c (PHYCTRL3c, Offset = 0x6E)

Bit	Name	Type	Default Value	Description
7	-	RW	0x0	Reserved
6	ByPassClkInitHs	RW	0x0	For the debugging
5	XorSource	RW	0x0	For the debugging
4	ErrEscOpt	RW	0x0	For the debugging
3	ErrSyncEscOpt	RW	0x0	For the debugging
2	DelayLpRxEnOpt	RW	0x0	For the debugging
1	StopStToSmOpt	RW	0x0	For the debugging
0	ErrControlOpt	RW	0x0	For the debugging

Table 7-49. CSIRX PHY Control Register 3d (PHYCTRL3d, Offset = 0x6F)

Bit	Name	Type	Default Value	Description
[7:0]	-	RW	0x0	Reserved

7.3.3.26 Frame Rate Control Registers (FRCR, Offset = 0x80 ~ 0x81)

The frame rate control registers include both "Frame Enable Register" and "Frame Control Wrap Register". The setting of these registers can control the corresponding frame transmitting at DPI. When FCWR***0***.W = ***m***, it means CSIRX performs (***m***+1) frame enable bits wrapping for VC0 by using the register bits FER0.E[***m***:0]. The wrapping starts at the bit FER0.E[0] so the VC0 1st frame is controlled by FER0.E[0].

For example, when FCWR0.W = 0b101 and FER0.E = 0b10111001, CSIRX will perform 6 frame enable bits wrapping by using FER0.E[5:0] for VC0. So the sequence of the frame enable will be FER0.E[0] (Enable), FER0.E[1] (Disable), FER0.E[2] (Disable), FER0.E[3] (Enable), FER0.E[4] (Enable), FER0.E[5] (Enable), FER0.E[0] (Enable), FER0.E[1] (Disable), …….

By default, FCWR0.W = 0 and FER0.E = 0xFF, CSIRX will depend on FER0.E[0] to transmit the frame so the frame transmitting is enabled by default.

Table 7-50. Frame Control Wrap Register for VC0 (FCWR0, Offset = 0x80)

Bit	Name	Type	Default Value	Description
[7:3]	-	RO	0	Reserved
[2:0]	W	RW	0	Wrap Size This field indicates the wrap size. Please refer to the above description of this section.

Table 7-51. Frame Enable Register for VC0 (FER0, Offset = 0x81)

Bit	Name	Type	Default Value	Description
[7:0]	E	RW	0xFF	Frame Enable This field controls the corresponding frame enable. Please refer to the above description of this section. 1: Enable this frame transmitting 0: Disable this frame transmitting

7.3.3.27 Frame Number Register (FNR, Offset = 0x88 ~ 0x89)

These registers reflect the value of the IP interface signal csi_vc0_fnum.

Table 7-52. Frame Number Lower Register for VC0 (FNLR0, Offset = 0x88)

Bit	Name	Type	Default Value	Description
[7:0]	N	RO	0	Frame Number This field reflects the value of csi_vc0_fnum[7:0].

Table 7-53. Frame Number Higher Register for VC0 (FNHR0, Offset = 0x89)

Bit	Name	Type	Default Value	Description
[7:0]	N	RO	0	Frame Number This field reflects the value of csi_vc0_fnum[15:8].

7.3.3.28 DPI Built-in Pattern Generator Registers (BPGR, Offset = 0x90 ~ 0x91)

DPI built-in pattern generator register defines the vertical line number, generator enable, and pattern type selection. About the horizontal size of the built-in pattern, it is defined by HPNR registers (Address = 0x20 ~ 0x21). MCR register is still effective for DPI built-in pattern.

The vertical line size of the VC***0*** DPI built-in pattern (Unit is line) is the value of register {BPGHR***0***.VLN, BPGLR***0***.VLN}. This value is zero, the vertical line size of the VC***0*** DPI built-in pattern is 4096 lines.

Table 7-54. DPI Built-in Pattern Generator Lower Register for VC0 (BPGLR0, Offset = 0x90)

Bit	Name	Type	Default Value	Description
[7:0]	VLN	RW	0	Vertical Line Number This field defines the bit#7~0 of the vertical line number.

Table 7-55. DPI Built-in Pattern Generator Higher Register for VC0 (BPGHR0, Offset = 0x91)

Bit	Name	Type	Default Value	Description
[7:6]	PT	RW	0	Pixel Type This field selects the pixel type of the built-in pattern. When this field is: 2'b00: Pixel type is RAW8. 2'b01: Pixel type is RAW10. 2'b10: Pixel type is RAW12. 2'b11: Pixel type is RAW14.
5	PS	RW	0	Pattern Selection This bit selects the built-in pattern type. When this bit is: 0: No swap pattern. 1: Swap pattern per 15 frames.
4	GE	RW	0	Generator Enable

Bit	Name	Type	Default Value	Description
				This bit controls the built-in pattern generator enable. When this bit is: 0: Disable the built-in pattern generator. 1: Enable the built-in pattern generator.
[3:0]	VLN	RW	4'b0100	Vertical Line Number This field defines the bit#11~8 of the vertical line number.

7.3.4 Initialization/Application Information

7.3.4.1 Configure by Programming Registers

CSIRX provides the registers for software configuration, which are described in "Programming Model". After finishing the register configuration, the software shall perform the software reset to reset the IP interface signals, state machine, FIFOs, counters, and status registers. Please refer to "CSIRX Programming Model" for details.

7.3.4.2 ForceRxmode Signal for Initialization Period

"The Initialization period" is defined in the D-PHY specification. The ITR registers (Address = 0x12 ~ 0x13) are used to adjust the assertion duration of PPI signal, ForceRxmode, to meet the $T_{INIT,\ SLAVE}$ timing parameter. The value of ITR registers may be changed for different Slave side PHY (CSI receiver) or Master side PHY (CSI transmitter).

7.3.4.3 Software Reset

Software can issue the software reset by the following two steps:

Step 1: Set the CR.SR bit.

Step 2: Polling the CR.SR bit unit it is read as zero.

CSIRX performs the software reset as below when the related clocks are valid. The software reset must be finished after the transactions of the above steps are finished.

- Reset the status registers: CSIERR0 (Offset = 0x30), CSIERR1 (Offset = 0x31), INTSTS (Offset = 0x33), and DPISR0 (Offset = 0x38)
- PPI interface returns to the initial state.
- Clear the pixel (DPI) FIFO(s) and the counter for the pixel number
- Clear timer counters mentioned in the register CSIERR0.HSRT bit (Offset = 0x30)
- DPI interface signals return to the initial state.
- Clear the interrupt
- Assert the ForceRxmode signal when the values of the Initialization Timer Register (ITR, Offset = 0x12 and 0x13) are not zero.

7.4 subLVDS Controller

7.4.1 General Description

The subLVDS controller is a high-speed and high-resolution interconnection for the subLVDS receiver. The supported data rate is up to 1Gbps per lane and is scalable from one to two data lanes and the maximum throughput will be 2Gbps when two data lanes are active. Various resolutions and pixel formats are supported.

7.4.2 Features

- Receives subLVDS serial data of Sony, Panasonic, and OmniVision sensor
- Supports active lane number: 1 and 2 lanes
- Supports serial input bit number per pixel: 8, 10, 12, and 14bits
- Supports maximum image size of 1920x1080
- Supports user-defined synchronization code on 4^{th} word

- Supports reference lane of synchronization switch option
- Supports data bit swap option
- Supports lane switch option on lane0-0 and lane-1
- Supports alignment automatic redo function
- Supports interrupt for alignment automatic redo alarm

7.4.3 Programming Model

7.4.3.1 Summary of the subLVDS Controller Registers

Table 7-56 provides a summary of the subLVDS controller registers.

Table 7-56. Summary of subLVDS Controller Registers

Offset	Name	Reset Value
+0x00	subLVDS_CTRL_Reg	0x0000 0000
+0x04	subLVDS_SIZE_Reg	0x0000 0000
+0x08	subLVDS_SYNC_Reg	0x0000 0000
+0x0C ~ 0x7C	Reserved	0x0000 0000
+0x80	subLVDS_VER_Reg	0x0000 0000
+0x84 ~ 0x9C	subLVDS_STATUS_Reg	0x0000 0000
+0xA0 ~ 0xCC	subLVDS_DEBUG_Reg	0x0000 0000

7.4.3.2 Register Descriptions

The following sections describe the subLVDS controller registers in more detail.

The abbreviations below represent the access types used throughout the register descriptions:

- R/W: Read/Write
- RO: Read Only
- WO: Write Only
- W1C: Write 1 Clear

7.4.3.3 subLVDS_CTRL_Reg (Offset = 0x00)

Table 7-57. subLVDS_CTRL_Reg (Offset = 0x00)

Bit	Name	Type	Description	Reset Value
[31]	lvds_dbgcnt_reset	WO	Write 1 to reset the subLVDS debug counter	0x0
[30:25]	-	-	Reserved	-
[24]	Lvds_resy_intr_en	R/W	subLVDS alignment auto-redo interrupt mask 0: Disable the interrupt 1: Enable the interrupt	0x0
[23:20]	-	-	Reserved	-
[19]	data_lane_order	R/W	subLVDS data transmission mode 0: lane order = 0-1 1: lane order = 1-0	0x0
[18]	rx_syn_code_shift	R/W	subLVDS 4th sync. code alignment location 0: 8bit code shift to right for aligned on LSB 1: 8bit code shift to left for aligned on MSB	0x0
[17]	rx_syn_code_sel	R/W	subLVDS uses sync. code type selection 0: Use SAV/EAV for valid/invalid line (For Sony or OmniVision sensor) 1: Use SOF/SOL/EOF/EOL (For Panasonic sensor)	0x0
[16]	rx_syn_code_mode	R/W	subLVDS uses user-defined sync. code control 0: Use the default sync. code for the SONY/OmniVision/Panasonic sensors 1: Use user-defined sync code (On subLVDS_SYNC_Reg)	0x0
[15]	rx_align_ard_dis	R/W	subLVDS alignment auto-redo function disable control 0: Enable 1: Disable	0x0
[14]	rx_setting_ard_en	R/W	subLVDS alignment auto-redo function enable control (When setting is changed.) 0 : Enable 1 : Disable	0x0
[13]	din_align_weight	R/W	subLVDS alignment reference behavior setting 0: Controller will be aligned when the reference channel is aligned. 1: Controller will be aligned when at least one more channel is aligned beside the reference channel.	0x0

Bit	Name	Type	Description	Reset Value
[12]	-	-	Reserved	0x0
[11:10]	lane_num_mode	R/W	subLVDS channel/port number 0: One channel/port 1: Two channels/ports 3: Four channels/ports	0x0
[9:8]	-	-	Reserved	-
[7]	pixel_bit_order	R/W	subLVDS pixel data transmission mode 0: subLVDS pixel data are transmitted LSB first. 1: subLVDS pixel data are transmitted MSB first.	0x0
[6:4]	bit_num_mode	R/W	subLVDS bit number per pixel 0: 8bit per pixel 1: 10bit per pixel 2: 12bit per pixel 3: 14bit per pixel Others: Reserved	0x0
[3:2]	din_align_base	R/W	subLVDS alignment and sync reference channel/port 0: Alignment and sync. according to channel0/port0 1: Alignment and sync. according to channel1/port1 2: Alignment and sync. according to channel2/port2 3: Alignment and sync. according to channel3/port3	0x0
[1]	din_flip_mode	R/W	subLVDS byte data input flip mode Set to 1'b1	0x0
[0]	lvds_rx_enable	R/W	subLVDS Controller enable control 0: Disable 1: Enable	0x0

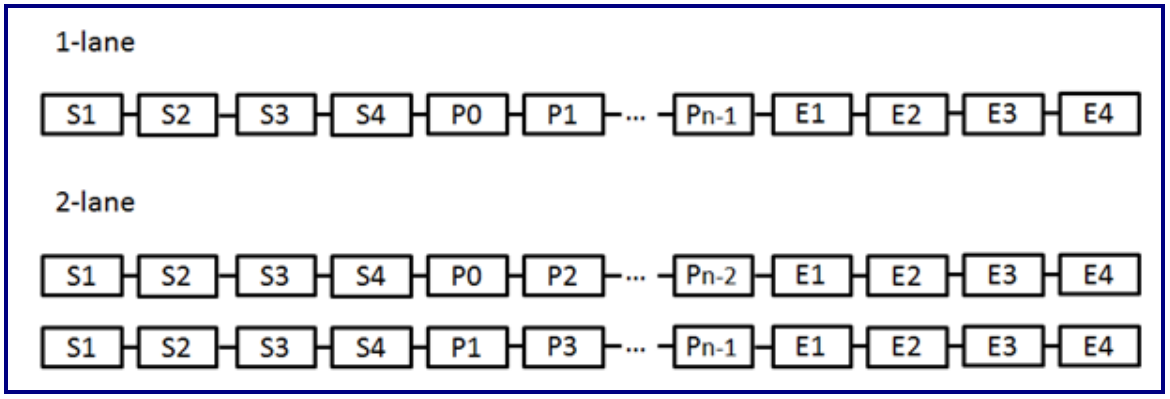


Figure 7-2. Relationship between Pixel Data and Sync. Codes of 1-lane and 2-lane

Note: S*n* means the *n*-th word of start code, E*n* means the *n*-th word of end code, and P*n* means the *n*-th pixel data.

7.4.3.4 subLVDS_SIZE_Reg (Offset = 0x04)

Table 7-58. subLVDS_SIZE_Reg (Offset = 0x04)

Bit	Name	Type	Description	Reset Value
[31:29]	-	-	Reserved	-
[28:16]	src_height	R/W	subLVDS source image height	0x0
[15:14]	-	-	Reserved	-
[13:0]	src_width	R/W	subLVDS source image width	0x0

7.4.3.5 subLVDS_SYNC_Reg (Offset = 0x08)

Table 7-59. subLVDS_SYNC_Reg (Offset = 0x08)

Bit	Name	Type	Description	Reset Value
[31:24]	sync_code_e1	R/W	subLVDS 4th sync. code of end code type 2 For Sony/OmniVision sensor: Valid line EAV For Panasonic sensor: EOL	0x0
[23:16]	sync_code_s1	R/W	subLVDS 4th sync code of start code type 2 For Sony/OmniVision sensor: Valid line SAV For Panasonic sensor: SOL	0x0

Bit	Name	Type	Description	Reset Value
[15:8]	sync_code_e0	R/W	subLVDS 4th sync code of end code type 1 For Sony/OmniVision sensor: Invalid line EAV For Panasonic sensor: EOF	0x0
[7:0]	sync_code_s0	R/W	subLVDS 4th sync code of start code type 1 For Sony/OmniVision sensor: Invalid line SAV For Panasonic sensor: SOF	0x0

7.4.3.6 subLVDS_INTR_Reg (Offset = 0x20)

Table 7-60. subLVDS_INTR_Reg (Offset = 0x20)

Bit	Name	Type	Description	Reset Value
[31:1]	-	-	Reserved	-
[0]	lvds_resync_intr	W1C	subLVDS alignment auto-redo interrupt status	0x0

7.4.3.7 subLVDS_VER_Reg (Offset = 0x80)

Table 7-61. subLVDS_VER_Reg (Offset = 0x80)

Bit	Name	Type	Description	Reset Value
[31:0]	ip_version	RO	subLVDS controller release date and version	0x1303270F

7.4.3.8 subLVDS_STATUS_Reg0 (Offset = 0x84)

Table 7-62. subLVDS_STATUS_Reg0 (Offset = 0x84)

Bit	Name	Type	Description	Reset Value
[31:28]	-	-	Reserved	-
[27:16]	hb_cnt_lat	RO	subLVDS horizontal blank length	0x0
[15:8]	-	-	Reserved	-
[7:4]	rx_fsm_state	RO	subLVDS RX state of finite state machine	0x0
[3]	di_align_on	RO	subLVDS DI data aligned flag	0x0
[2:0]	di_align_loc	RO	subLVDS DI first aligned location	0x0

7.4.3.9 subLVDS_STATUS_Reg1 (Offset = 0x88)

Table 7-63. subLVDS_STATUS_Reg1 (Offset = 0x88)

Bit	Name	Type	Description	Reset Value
[31:29]	-	-	Reserved	-
[28:16]	line_cnt_lat	RO	subLVDS received lines per frame	0x0
[15:14]	-	-	Reserved	-
[13:0]	pixel_cnt_lat	RO	subLVDS received pixels per line	0x0

7.4.3.10 subLVDS_STATUS_Reg2 (Offset = 0x8C)

Table 7-64. subLVDS_STATUS_Reg2 (Offset = 0x8C)

Bit	Name	Type	Description	Reset Value
[31:24]	-	-	Reserved	-
[23:0]	dv_cnt_lat	RO	subLVDS data output pixels per frame	0x0

7.4.3.11 subLVDS_DEBUG_Reg0 (Offset = 0xA0)

Table 7-65. subLVDS_DEBUG_Reg0 (Offset = 0xA0)

Bit	Name	Type	Description	Reset Value
[31:24]	di_align_msmh_01_cnt	RO	subLVDS alignment mismatch debug counter for comparing channel 0 and channel 1	0x0
[23:0]	-	-	Reserved	-

7.4.3.12 subLVDS_DEBUG_Reg1 (Offset = 0xA4)

Table 7-66. subLVDS_DEBUG_Reg1 (Offset = 0xA4)

Bit	Name	Type	Description	Reset Value
[31:0]	-	-	Reserved	-

7.4.3.13 subLVDS_DEBUG_Reg2 (Offset = 0xA8)

Table 7-67. subLVDS_DEBUG_Reg2 (Offset = 0xA8)

Bit	Name	Type	Description	Reset Value
[31:0]	-	-	Reserved	-

7.4.3.14 subLVDS_DEBUG_Reg3 (Offset = 0xAC)

Table 7-68. subLVDS_DEBUG_Reg3 (Offset = 0xAC)

Bit	Name	Type	Description	Reset Value
[31:0]	-	-	Reserved	-

7.4.3.15 subLVDS_DEBUG_Reg4 (Offset = 0xB0)

Table 7-69. subLVDS_DEBUG_Reg4 (Offset = 0xB0)

Bit	Name	Type	Description	Reset Value
[31:24]	rx_ali_redo_req_cnt	RO	subLVDS RX alignment of the auto-redo request debug counter	0x0
[23:16]	chk_sync_unknown_cnt	RO	subLVDS RX detect of the unknown sync. code debug counter.	0x0
[15:8]	do_fifo_full_cnt	RO	subLVDS DO FIFO overflow debug counter	0x0
[7:0]	di_fifo_full_cnt	RO	subLVDS DI FIFO overflow debug counter	0x0

7.4.3.16 subLVDS_DEBUG_Reg5 (Offset = 0xB4)

Table 7-70. subLVDS_DEBUG_Reg5 (Offset = 0xB4)

Bit	Name	Type	Description	Reset Value
[31:16]	-	-	Reserved	-
[15:8]	line_start_cnt	RO	subLVDS RX received line counter	0x0
[7:0]	line_end_cnt	RO	subLVDS DO sent line counter	0x0

7.4.3.17 subLVDS_DEBUG_Reg6 (Offset = 0xB8)

Table 7-71. subLVDS_DEBUG_Reg6 (Offset = 0xB8)

Bit	Name	Type	Description	Reset Value
[31:16]	frm_start_cnt	RO	subLVDS RX received frame counter	0x0
[15:0]	frm_end_cnt	RO	subLVDS DO sent frame counter	0x0

7.4.3.18 subLVDS_DEBUG_Reg7 (Offset = 0xBC)

Table 7-72. subLVDS_DEBUG_Reg7 (Offset = 0xBC)

Bit	Name	Type	Description	Reset Value
[31:0]	-	-	Reserved	-

Chapter 8

Video Output Interface

This chapter contains the following sections:

8.1 LCD Controller

8.1.1 General Description

GM8136S/GM8135S LCD controller (LCDC) is used to get the video data from the frame buffer and output to provide all necessary control signals for various TFT LCD monitors or TV encoders.

8.1.2 Features

- TFT panel interface
  - Pixel clock rate of up to 150MHz
  - RGB parallel output: 16bit bus interface
  - Programmable resolution up to 1920x1080
  - Programmable polarity/duration for output enable, vertical sync., horizontal sync., and pixel clock
  - Data/Synchronization on/off controls
  - Virtual screen
- TV interface
  - ITU-R BT. 656 output (SDTV)
  - ITU-R BT. 1120 output (HDTV: 1080i and 1080p)
- Image input format
  - RGB 12 (4:4:4)/16 (5:6:5)/15 (5:5:5)/24 (8:8:8) bits per pixel
  - Palette (8bit, 4bit, 2bit, and 1bit pixel)
  - YCbCr422 (16bits per pixel)
  - Chooses input format (Except for YCbCr420 format) as interlace or progress arrangement for TV output
  - Little-endian, big-endian, and WinCE
- Palette
  - 256 entries 16bit RGB color palette RAM
- Dithering
- Picture-in-Picture (PiP)
  - Displays a maximum of two PiP windows
  - Resolution of PiP window similar to the main window

- Supports 8bit global blending levels
- Supports 8bit ARGB (8888) for each picture
- Supports 1bit ARGB (1555) for each picture
- Supports color-key

- Interrupt control
  - Master bus error
  - Frame status
  - FIFO under-run
  - Memory base update
- General-purpose inputs/outputs
  - Provides eight input ports and eight output ports for general-purpose applications
- Scaler (Does not support TV interlace input or interlace output)
  - Down-scaling ration ranging from $\frac{1}{256} \times \frac{1}{256}$ to 1x1
  - Arbitrary ration between 1x1 and 2x2 for up-scaling stage
  - Three interpolation modes for up-scaling: nearly bilinear interpolation, threshold nearly bilinear interpolation, and most neighborhood interpolation

8.1.3 Programming Model

8.1.3.1 Summary of LCD Controller Registers

Table 8-1. Summary of LCD Controller Registers

LCD Global Parameters					
Address	**Type**	**Width**	**Reset Value**	**Name**	**Description**
0x0000	R/W	17	0x00	lcd_function_en	LCD function enable
0x0004	R/W	21	0x00	lcd_panel_pixel	LCD panel pixel parameter
0x0008	R/W	4	0x00	lcd_intr_mask	LCD interrupt enable mask
0x000C	W	4	0x00	lcd_intr_clr	LCD interrupt status clear
0x0010	R	4	0x00	lcd_intr_status	LCD interrupt status
0x0014	R/W	16	0x00	lcd_frame_buffer	LCD frame buffer parameter
0x0018	R/W	30	0x00	lcd_im0frm0base	LCD panel image0 frame0 base address
0x0024	R/W	30	0x00	lcd_im1frm0base	LCD panel image1 frame0 base address

LCD Global Parameters					
Address	**Type**	**Width**	**Reset Value**	**Name**	**Description**
0x0030	R/W	30	0x00	-	Reserved
0x003C	R/W	30	0x00	-	Reserved
0x0048	R/W	8	0x00	lcd_patbarcolor	LCD pattern generator
0x004C	R/W	32	0x04040404	lcd_fifoth	LCD FIFO threshold
0x0050	R/W	10	0x200	lcd_bandwithCtrl	LCD bandwidth control
0x0058	R	-	-	-	LCD revision
0x005C	R/W	30	0x00	-	Reserved
0x0060	-	-	-	-	Reserved
LCD timing and polarity parameters					
0x0100	R/W	32	0x00	lcd_hortiming	LCD horizontal timing control
0x0104	R/W	32	0x00	lcd_vertiming0	LCD vertical timing control0
0x0108	R/W	8	0x00	lcd_vertiming1	LCD vertical timing control1
0x010C	R/W	23	0x00	lcd_polarity	LCD polarity control
LCD output format parameters					
0x0200	-	-	-	-	Reserved
0x0204	R/W	4	0x00	lcd_tv_parm0	LCD TV parameters
0x0208	R/W	24	0x00	lcd_tv_parm1	LCD TV line/cycle parameters
0x020C	R/W	24	0x00	lcd_tv_parm2	LCD TV field polarity parameters
0x0210	R/W	22	0x00	lcd_tv_parm3	LCD TV vertical blank 0 ~ 1
0x0214	R/W	22	0x00	lcd_tv_parm4	LCD TV vertical blank 2 ~ 3
0x0218	R/W	24	0x00	lcd_tv_parm5	LCD TV vertical active parameters
0x021C	R/W	22	0x00	lcd_tv_parm6	LCD TV horizontal blank 0 ~ 1
0x0220	R/W	10	0x00	lcd_tv_parm7	LCD TV horizontal blank2
0x0224	R/W	12	0x00	lcd_tv_parm8	LCD TV horizontal active parameters
0x0228	R/W	22	0x00	lcd_tv_parm9	LCD TV vertical blank 4 ~ 5
0x022C	R/W	22	-	-	Reserved
LCD image parameters					
0x0300	R/W	24	0x00	lcd_pipblend	LCD PIP parameters
0x0304	R/W	29	0x00	lcd_pip1pos	PiP sub-picture1 position
0x0308	R/W	27	0x00	lcd_pip1dim	PiP sub-picture1 dimension
0x030C	R/W	27	0x00	-	Reserved
0x0310	R/W	27	0x00	-	Reserved
0x0314	R/W	8	0xE4	lcd_pipriority	PiP image priority
0x0318	R/W	32	0x00	lcd_param1	PiP image format 1
0x031C	R/W	6	0x00	-	Reserved

LCD Global Parameters					
Address	**Type**	**Width**	**Reset Value**	**Name**	**Description**
0x0320	R/W	25	0x00	lcd_colorkey1	PiP color key1
0x0324	-	-	-	-	Reserved
0x0328	-	-	-	-	Reserved
0x032C	R/W	27	0x00	-	Reserved
0x0330	R/W	27	0x00	-	Reserved
LCD palette accessing port					
0x0A00 ~ 0x0BFC	W	32	-	PaletteWritePort	LCD palette RAM write accessing port
Scaler control registers					
0x1100	R/W	12	0x00	scal_hor_no_in	Scaler horizontal resolution of input
0x1104	R/W	12	0x00	scal_ver_no_in	Scaler vertical resolution of input
0x1108	R/W	14	0x00	scal_hor_no_out	Scaler horizontal resolution of output
0x110C	R/W	14	0x00	scal_ver_no_out	Scaler vertical resolution of output
0x1110	R/W	9	0x00	scal_misc	Scaler miscellaneous control registers
0x1114	R/W	9	0x00	scal_hor_high_th	Horizontal up-scaling high threshold
0x1118	R/W	9	0x00	scal_hor_low_th	Horizontal up-scaling low threshold
0x111C	R/W	9	0x00	scal_ver_high_th	Vertical up-scaling high threshold
0x1120	R/W	9	0x00	scal_ver_low_th	Vertical up-scaling low threshold
0x112C	R/W	16	0x00	scal_hor_ver_num	Scaler resolution parameter
Virtual registers					
0x1500	R/W	21	0x00	vs_control	Virtual screen control
0x1504	R/W	21	0x00	vs1_control	Virtual screen1 control
0x1508	-	-	-	-	Reserved
0x150C	-	-	-	-	-

8.1.3.2 LCD Global Parameters

8.1.3.2.1 LCD Function Enable Parameters (Address = 0x0000)

Table 8-2. LCD Function Enable Parameters (Address = 0x0000)

Bit	Name	Type	Description
19	Deflicker_En	R/W	Deflicker enable When this bit set to '1', the LCD controller will perform the deflicker function to replace the original sharpness function. 0: Disable the deflicker function 1: Enable the deflicker function
18	-	-	Reserved
17	Double_En	R/W	Double word command enable When this bit set to '1', the DMA will use the double word command (Eight bytes) to catch the image data from frame buffer. 0: Disable the double word access 1: Enable the double word access
16	AddrSyn_En	R/W	Address sync. enable When this bit sets to '1', the initial address of each DMA buffer can be updated to new address with the "AddrUpdate" bit set to '1'. When user wants to update the base address of all DMA buffers at same time, this bit must be set to enable the sync. address update. 0: Disable the address synchronous 1: Enable the address synchronous
15	-	-	Reserved Please set to "0".
14	PatGen	R/W	Test pattern generator When the pattern generator is enabled, the LCD controller will automatically send out the patterns to the panel depending on the programmed value of 0x48. 0: Turn-off pattern generator 1: Turn-on pattern generator
13	TVEn	R/W	TV output function When this bit is set, the LCD controller supports the TV output. 0: Disable the TV output 1: Enable the TV output
12	-	-	Reserved Please set to "0".

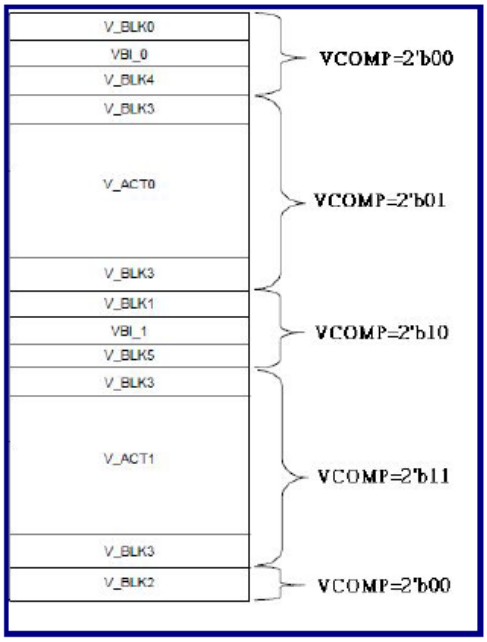
Bit	Name	Type	Description
[11:10]	PiPEn	R/W	Picture-in-Picture (PiP) mode When this field is set, the LCD controller can support the PiP function. The PiP function can support windows with different frame buffers. 00: Disable PiP 01: Single PiP window 10: Reserved 11: Reserved
[9:8]	BlendEn	R/W	Alpha Blending enable When this bit is set, the PiP function can support the Alpha Blending function. 00: Disable the Alpha Blending function 01: Enable the global Alpha Blending function. The blending levels are depending on the “PiPBlend” register. 10: Enable the pixel Alpha Blending function. The format of the input images must be ARGB8888 or ARGB1555. The blending levels are depending on the “A” component of the input data. 11: Reserved
7	-	-	Reserved
6	DitherEn	R/W	Dither enable When this bit is set, the LCD controller can support the dithering function. The dithering type is depending on the value of DitherType. 0: Disable the dithering function 1: Enable the dithering function
5	ScalerEn	R/W	Scaler enable When this bit is set, the LCD controller can support the scaling function. The detailed scaling dimension is depending on the scaler control registers. 0: Disable the scaling function 1: Enable the scaling function
4	Reserved-	-	Reserved
3	EnYCbCr	R/W	YCbCr input mode control When this bit is set, it informs the LCD controller that the data type in the frame buffer is the YCbCr format. 0: Enable the RGB format 1: Enable the YCbCr format
2	-	R/W	Reserved
1	LCDon	R/W	LCD screen on/off control 0: Disable the LCD screen (All data output pins are forced to ‘0’.) 1: Enable the LCD screen (Normal operation)

Bit	Name	Type	Description
0	LCDen	R/W	LCD controller enable control 0: Disable the LCD controller and force all LCD signals, including synchronization and the overall data, to zero. 1: Enable the LCD controller (Normal operation)

8.1.3.2.2 LCD Panel Pixel Parameters (Address = 0x0004)

Table 8-3. LCD Panel Pixel Parameters (Address = 0x0004)

Bit	Name	Type	Description
20	TVRST	R/W	TV_CLK domain reset The signal can reset all status in the TV clock domain. This register can only be cleared by users.
19	PRST	R/W	PCLK domain reset The signal can reset all status except the configuration parameters in the APB clock domain. This register can only be cleared by users.
18	LRST	R/W	LC_CLK domain reset The signal can reset all status in the LCD clock domain. This register can only be cleared by users.
17	MRST	R/W	Master domain reset (AXI or AHB) The signal can reset all status in the AXI/AHB domain. This register can only be cleared by users.
16	AddrUpdate	R/W	Address update This bit is valid when AddrSyn_En (0000H) is set. When AddrUpdate is set and AddrSyn_En = '1', the base address of each DMA buffer can be updated.
[15:14]	UpdateSrc	R/W	Update source selection The bit, IntNxtBase, is used to specify the image sources, which can be: 00: Image0 01: Image1 10: Reserved 11: Reserved

Bit	Name	Type	Description
[13:12]	DitherType	R/W	Dithering type This field decides which dithering algorithm can be used. The value is valid when DitherEn is set. 00: Transfer 888 to 565 01: Transfer 888 to 555 10: Transfer 888 to 444 11: Reserved
11	-	-	Reserved and set to “0”
[10:9]	Vcomp	R/W	Generate the vertical status interrupt: If the output is TFT: 00: Start of the vertical sync. 01: Start of the vertical back porch 10: Start of the vertical active image 11: Start of the vertical front porch If TV is enabled, the interrupt will be triggered as follows: 

Bit	Name	Type	Description
[8:7]	RGBTYPE	R/W	RGB input format This bit identifies the input RGB format when BppFifo = '100'. 00: RGB 565 input 01: RGB 555 input 10: RGB 444 input 11: Reserved
[6:5]	Endian	RW	Frame buffer data endianness control 00: Little-endian byte little-endian pixel 01: Big-endian byte big-endian pixel 10: Little-endian byte big-endian pixel (WinCE) 11: Reserved
4	BGRSW	R/W	RGB or BGR output format selection 0: RGB normal output 1: BGR red and blue swapped
3	PWROFF	R/W	This bit directly rives the output pin, LC_PWROFF.
[2:0]	BppFifo	R/W	Pixel format in FIFO The value informs the data format in each DMA buffer (Bits per pixel). 000: 1bpp 001: 2bpp 010: 4bpp 011: 8bpp 100: 16bpp 101: 24bpp Others: Reserved

8.1.3.2.3 LCD Interrupt Enable Mask Parameter (Address = 0x0008)

Table 8-4. LCD Interrupt Enable Mask Parameters (Address = 0x0008)

Bit	Name	Type	Description
3	IntBusErrEn	R/W	AXI/AHB master error interrupt enable This bit is set to enable the bus error interrupt. 1: Enable 0: Disable
2	IntVstatusEn	R/W	Vertical duration comparison interrupt enable This bit is set to enable the vertical status interrupt. 1: Enable 0: Disable
1	IntNxtBaseEn	R/W	Next frame base address updated interrupt enable This bit is set to enable the successfully base address updated interrupt. 1: Enable 0: Disable
0	IntFIFOUdnEn	R/W	FIFO under-run interrupt enable This bit is set to enable the FIFO under-run interrupt. 1: Enable 0: Disable

8.1.3.2.4 LCD Interrupt Status Clear Parameters (Address = 0x000C)

Table 8-5. LCD Interrupt Status Clear Parameters (Address = 0x000C)

Bit	Name	Type	Description
3	ClrBusErr	W	Setting this bit to ‘1’ clears the AXI/AHB master error interrupt status. Setting this bit to ‘0’ has no effect.
2	ClrVstatus	W	Setting this bit to ‘1’ clears the interrupt status of vertical duration comparison. Setting this bit to ‘0’ has no effect.
1	ClrNxtBase	W	Setting this bit to ‘1’ clears the frame buffer base address update interrupt status. Setting this bit to ‘0’ has no effect.
0	ClrFIFOUdn	W	Setting this bit to ‘1’ clears the FIFO under-run interrupt status. Setting this bit to ‘0’ has no effect.

8.1.3.2.5 LCD Interrupt Status Parameters (Address = 0x0010)

Table 8-6. LCD Interrupt Status Parameters (Address = 0x0010)

Bit	Name	Type	Description
3	IntBusErr	R	AXI/AHB master error status This bit is set when the AMBA AXI/AHB master encounters a bus error response from a slave.
2	IntVstatus	R	Vertical comparison When this bit is set, one of the four vertical durations is reached. This duration can be selected via the LCD control registers.
1	IntNxtBase	R	Frame buffer base address update When this bit is set, the current base address registers are successfully updated by the next address registers.
0	IntFIFOUdn	R	FIFO under-run When this bit is set, the DMA FIFO is a read access for an under-run condition caused by empty.

8.1.3.2.6 Frame Buffer Parameters (Address = 0x0014)

Table 8-7. Frame Buffer Parameters (Address = 0x0014)

Bit	Name	Type	Description
[15:14]	-	-	Reserved
[13:12]	-	-	Reserved
[11:10]	Im1ScalDown	R/W	Scaling down for image1 The register is valid when PoP or TV enable without PiP. The image from “LCDImage1FrameBase” can be scaled down depending on the value. 00: Disable 01: Image1 will be scaling down to 1/2x1/2. 10: Image1 will be scaling down to 1/2x1. 11: Reserved Please note that this register field can only be set to ‘00’ under the following conditions: • VirtualScreenEn or LCM_En is set. • PiP has a chance to be turned-on when TV is enabled.

Bit	Name	Type	Description
[9:8]	Im0ScalDown	R/W	Scaling down for image0 The register is valid when PoP or TV enable without PiP. The image from “LCDImage0FrameBase” can be scaled down depending on the value. 00: Disable 01: Image0 will be scaling down to 1/2x1/2. 10: Image0 will be scaling down to 1/2x1. 11: Reserved Please note that this register field can only be set to ‘00’ under the following conditions: • VirtualScreenEn or LCM_En is set. • PiP has a chance to be turned-on when TV is enabled.
[7:0]	-	-	Reserved

8.1.3.2.7 LCD Panel Image0 Frame0 Base Address (Address = 0x0018)

Table 8-8. Image0 Frame Base Address (Address = 0x0018)

Bit	Name	Type	Description
[31:2]	LCDImage0FrameBase	R/W	LCD frame0 base address of image0 This is the start address of the frame data in a memory. Two default value of the LSB bits is 0. The total occupied memory address range is: Starting address: {LCDImageFrame0Base, 0, 0} If the system is AXI bus, the base address must be set as 4-kbyte alignment.

8.1.3.2.8 LCD Panel Image1 Frame0 Base Address (Address = 0x0024)

Table 8-9. Image1 Frame Base Address (Address = 0x0024)

Bit	Name	Type	Description
[31:2]	LCDImage1FrameBase	R/W	LCD frame0 base address of image1 This is the start address of the frame data in a memory. The default value of the LSB bits is 0. The total occupied memory address range is: Starting address: {LCDImage1Frame0Base, 0, 0} If the system is AXI bus, the base address must be set as 4-kbyte alignment.

8.1.3.2.9 PatGen Pattern Bar Distance Parameters (Address = 0x0048)

Table 8-10. PatGen Pattern Bar Distance Parameters (Address = 0x0048)

Bit	Name	Type	Description
[31:8]	-	-	Reserved
[7:6]	-	-	Reserved
[5:4]	-	-	Reserved
[3:2]	Img1PatGen	R/W	Pattern generator of image1 The LCD controller will generate different test patterns of image1, depending on the value. 00: Vertical color bar 01: Horizontal color bar 10: Single color 11: Reserved
[1:0]	Img0PatGen	R/W	Pattern Generator of image0 The LCD controller will generate different test patterns of image0, depending on the value. 00: Vertical color bar 01: Horizontal color bar 10: Single color 11: Reserved

8.1.3.2.10 FIFO Threshold Control Parameters (Address = 0x004C)

Table 8-11. FIFO Threshold Control Parameters (Address = 0x004C)

Bit	Name	Type	Description
[31:24]	-	-	Reserved
[23:16]	-	-	Reserved
[15:8]	Buf1Threshold	R/W	DMA buffer 1 threshold Buffer 1 will request a master to fetch data in when the empty count of buffer 1 is larger than this value. This threshold must be larger or equal to 4.
[7:0]	Buf0Threshold	R/W	DMA buffer 0 threshold Buffer 0 will request a master to fetch data in when the empty count of buffer 0 is larger than this value. This threshold must be larger or equal to 4.

8.1.3.2.11 Bus Bandwidth Control Parameters (Address = 0x0050)

If users turn on the VBI Enable, VBI FIFO will receive the highest priority to use the bus.

Table 8-12. Bus Bandwidth Control Parameters (Address = 0x0050)

Bit	Name	Type	Description
9	Lock_En	R/W	Enable the LOCK command When this bit is set to '1', the LCD controller will use the LOCK command to request a transfer once the image data in the LCD controller is empty. The default setting is '1'. 0: Disable the LOCK command 1: Enable the LOCK command
8	Ratio_En	R/W	Enable the bus bandwidth ratio When this bit is set to '1', the LCD controller will use round robin and bandwidth ratio to arbitrate between frame buffers. When this bit is set to '0', the LCD controller will use only round robin to perform the arbitration between image frame buffers. 0: Disable the bus bandwidth ratio 1: Enable the bus bandwidth ratio
[7:6]	BwctrlF0	R/W	Bus bandwidth control ratio for the Image0 Frame buffer When Ratio_En is set, DMA of the LCD controller will use this ratio to catch data for image0. For example, when BwctrlF0: Ratio 4, BwctrlF1: Ratio 2, BwctrlF2 = Ratio 1, and BwctrlF3 = Ratio 1, the bandwidth ratio for four frame buffers is 4:2:1:1. The sequence of bus accessing is F0, F1, F2, F3, F0, F1, F0, and F0. 00: Ratio 1 01: Ratio 2 10: Ratio 4
[5:4]	BwctrlF1	R/W	Bus bandwidth control ratio for Image1 Frame buffer When Ratio_En is set, DMA of the LCD controller will use this ratio to catch data for image1. 00: Ratio 1 01: Ratio 2 10: Ratio 4
[3:2]	-	-	Reserved
[1:0]	-	-	Reserved

8.1.3.2.12 Revision Registers (Address = 0x0058)

This register shows the current revision of the LCD controller.

Table 8-13. Revision Registers (Address = 0x0058)

Bit	Name	Type	Description
[31:24]	-	-	Reserved
[23:16]	MAJOR_REV	RO	Major revision number
[15:8]	MINOR_REV	RO	Minor revision number
[7:0]	REL_REV	RO	Release number

8.1.3.2.13 LCD Panel VBI Base Address (Address = 0x005C)

Table 8-14. VBI Base Address (Address = 0x005C)

Bit	Name	Type	Description
[31:2]	LCDVBIBase	R/W	LCD VBI base address This is the start address of the VBI data in a memory. The default value of the LSB bits is 0. The total occupied memory address range is: Starting address: {LCDVBIBase, 0, 0} If the system is AXI bus, base address must be set as 4-kbyte alignment. Note: This base address is used for VBI when TV is turned on and VBI is enabled.

8.1.3.2.14 FIFO Threshold Control Parameters (Address = 0x0060)

Table 8-15. FIFO Threshold Control Parameters (Address = 0x0060)

Bit	Name	Type	Description
[7:0]	Buf4Threshold	R/W	DMA buffer 4 threshold Buffer 4 will request a master to fetch data in when the empty count of buffer 4 (VBI FIFO) is larger than this value. This threshold must be larger than or equal to 4.

8.1.3.3 LCD Timing and Polarity Parameters

8.1.3.3.1 LCD Horizontal Timing Control Parameters (Address = 0x0100)

Table 8-16. LCD Horizontal Timing Control Parameters (Address = 0x0100)

Bit	Name	Type	Description
[31:24]	HBP	R/W	**Horizontal Back Porch** **Actual horizontal back porch = HBP + 1** This field specifies the LC_PCLK periods between the LC_HS falling edge and start of active data. The 8bit HBP field is used to specify the pixel clock periods inserted at the beginning of each line or row. After the line clock for the previous line has been de-asserted, the value in HBP will be used to count the waited pixel clocks before starting the next display line. HBP generates a delay of 1 to 256 pixel clock cycles.
[23:16]	HFP	R/W	**Horizontal Front Porch** **Actual horizontal front porch = HFP + 1** This field specifies the LC_PCLK periods between the LC_HS rising edge and end of active data. The 8bit HFP field sets the pixel clock intervals at the end of each line or row before the LCD line clock is pulsed. Once complete a line of pixels transmitted to the LCD driver, the value in HFP will be used to count the pixel clocks for waiting time before asserting the line clock. HFP generates a period of 1 to 256 pixel clock cycles.
[15:8]	HW	R/W	**Horizontal Synchronization Pulses Width** **Actual horizontal synchronization pulses width = HW + 1** HW is the width of the LC_HS signal in the LC_PCLK periods. The 8bit HW field specifies the pulse width of a line clock in the passive mode or the horizontal synchronization pulse in the active mode.
[7:0]	PL	R/W	**Pixels-per-Line** **Actual pixels-per-line = 16 * (PL + 1)** This field specifies the pixels in each line or row of the screen. When PL is 6bits, pixels are between 16 and 2048 PL. PL is used to count the pixel clocks that occur before the HFP is applied (Program the required value is divided by 16 and minus 1). Users must ensure that each line start is double-word alignment and 16 * (PL + 1) * bpp is multiples of 128bits if 64bit AXI bus is used.

8.1.3.3.2 LCD Vertical Timing Control Parameters (Address = 0x0104)

Table 8-17. LCD Vertical Timing Control Parameters (Address = 0x0104)

Bit	Name	Type	Description
[31:24]	VFP	R/W	**Vertical Front Porch** **Actual vertical front porch = VFP** VFP is the inactive lines at the end of a frame before the vertical synchronization period. The 8bit VFP field is used to specify the line clocks inserted at the end of each frame. Once complete a frame of pixels transmitted to the LCD display, the value in VFP will be used to count the line clock periods for the waiting time after the count loses the asserted vertical synchronization (LC_VS) signal. VFP generates a period of 0 to 255 line clock cycles.
[23:22]	-	-	Reserved
[21:16]	VW	R/W	**Vertical Synchronization Pulse Width** **Actual vertical synchronization pulse width = VW + 1** VW is the horizontal synchronization lines. Program the number of lines required minus one. The 6bit VW field is used to specify the width of the vertical synchronization pulse. This field programs the horizontal synchronization lines and minus one with the line clocks in the vertical synchronization period.
[15:12]	-	-	Reserved
[11:0]	LF	R/W	**Lines-per-Frame** **Actual line-per-frame = LF + 1** LF is the active lines per frame. The LF field specifies the total controlled lines or rows on the LCD panel. LF is an 11bit value which is between 1 and 2048 lines.

8.1.3.3.3 LCD Vertical Back Porch Parameters (Address = 0x0108)

Table 8-18. LCD Vertical Back Porch Parameters (Address = 0x0108)

Bit	Name	Type	Description
[7:0]	VBP	R/W	**Vertical Back Porch** **Actual vertical back porch = VBP , when the LCD controller_SHARP is turned off.** **Actual vertical back porch = VBP + 1, when the LCD controller_SHARP is turned on.** VBP is the inactive lines at the start of a frame after vertical synchronization period. The 8bit VBP field is used to specify the number of line clocks inserted at the beginning of each frame. The VBP count starts just after the vertical synchronization signal for the previous frame has been negated for the active mode. After this has occurred, the count value in VBP sets the number of the line clock periods inserted before the next frame. VBP generates from 1 to 256 extra line clock cycles.

8.1.3.3.4 LCD Polarity Control Parameters (Address = 0x010C)

Table 8-19. LCD Polarity Control Parameters (Address = 0x010C)

Bit	Name	Type	Description
[22:16]	TV_DivNo	R/W	TV panel clock divisor control (TV_PCLK) TV_PCLK is divided from TV_CLK. The actual divisor value is equal to (TV_DivNo + 1).
[14:8]	DivNo	R/W	LCD panel clock divisor control (LC_PCLK) LC_PCLK is divided from LC_CLK. The actual divisor value is equal to (DivNo + 1).
[7:5]	-	-	Reserved
4	IPWR	R/W	0: LC_PWROFF output is active high. 1: LC_PWROFF output is active low. This bit selects the active polarity of the output pin, LC_PWROFF.
3	IDE	R/W	0: LC_DE/LC_AC output pin is active high. 1: LC_DE/LC_AC output pin is active low. The invert output enable bit selects the active polarity of the output enable signal.
2	ICK	R/W	0: Data is driven on the LCD data lines on the rising edge of LC_PCLK. 1: Data is driven on the LCD data lines on the falling edge of LC_PCLK. The ICK bit selects the edge of the panel clock on which the pixel data is driven out the LCD data lines.
1	IHS	R/W	0: LC_HS/LC_LP pin is active high and inactive low. 1: LC_HS/LC_LP pin is active low and inactive high. The invert horizontal sync. bit selects the active polarity of the horizontal sync. signal.
0	IVS	R/W	0: LC_VS/LC_FLM pin is active high and inactive low. 1: LC_VS/LC_FLM pin is active low and inactive high. The invert vertical sync. bit selects the active polarity of the vertical sync. signal.

8.1.3.4 LCD Output Format Parameters

8.1.3.4.1 LCD TV Parameters (Address = 0x0204)

This parameter will be valid when TVEn is set.

Notes:

1. When TVEn is set, the scaler function must be disabled for the interlace input or the interlace output format.
2. The total line number, LF (Offset 104H) and horizontal pixel number, PL (Offset 100H) must be set.

Table 8-20. LCD TV Parameter 0 (Address = 0x0204)

Bit	Name	Type	Description
3	BusWidth	R/W	TV data width selection 0: Output data is 8bit wide. 1: Output data is 16bit wide.
2	ImgFormat	R/W	Interlace/Progress image format selection The value decides the image in the frame buffer that is the format of “Progress” or “Interlace”. 0: Input image format is “Progress”. 1: Input image format is “Interlace”. The image format setting in offset 0x031C must match the setting in this register.
1	PHASE	R/W	TVE clock phase 0: TV clock is the same as TV_PCLK. 1: TV clock is the inversion of TV_PCLK.
0	OutFormat	R/W	Interlace/Progress output format selection 0: Interlace output 1: Progressive output (When the output format is progressive, the ImgFormat bit must be set to “Progress”.)

8.1.3.4.2 TV Control Parameters (Address = 0x0208)

Figure 8-1 and Figure 8-2 are the vertical and horizontal timing diagrams of the TV interface. Users can program the parameters based on the timing diagrams. Users can program any timing parameter for BT.656 and BT.1120. As shown in Figure 8-1, the active field includes three regions. The “V_ACT0/1” region contains the real image, and the “V_BLK3” regions contain black pixel to meet the vertical resolution, which represents that the height of the raw images does not meet the TV vertical resolution.

The LCD controller can patch black lines above/below the image to meet the TV vertical resolution. Figure 8-2 shows that the active line includes three regions, the "H_ACT0" region contains the real image, and the "H_BLK1" and "H_BLK2" regions contain the black pixel to meet the horizontal resolution, which represents that the width of raw image does not meet the TV horizontal resolution. The LCD controller can patch black pixel to left/right of the image to meet the TV horizontal resolution.

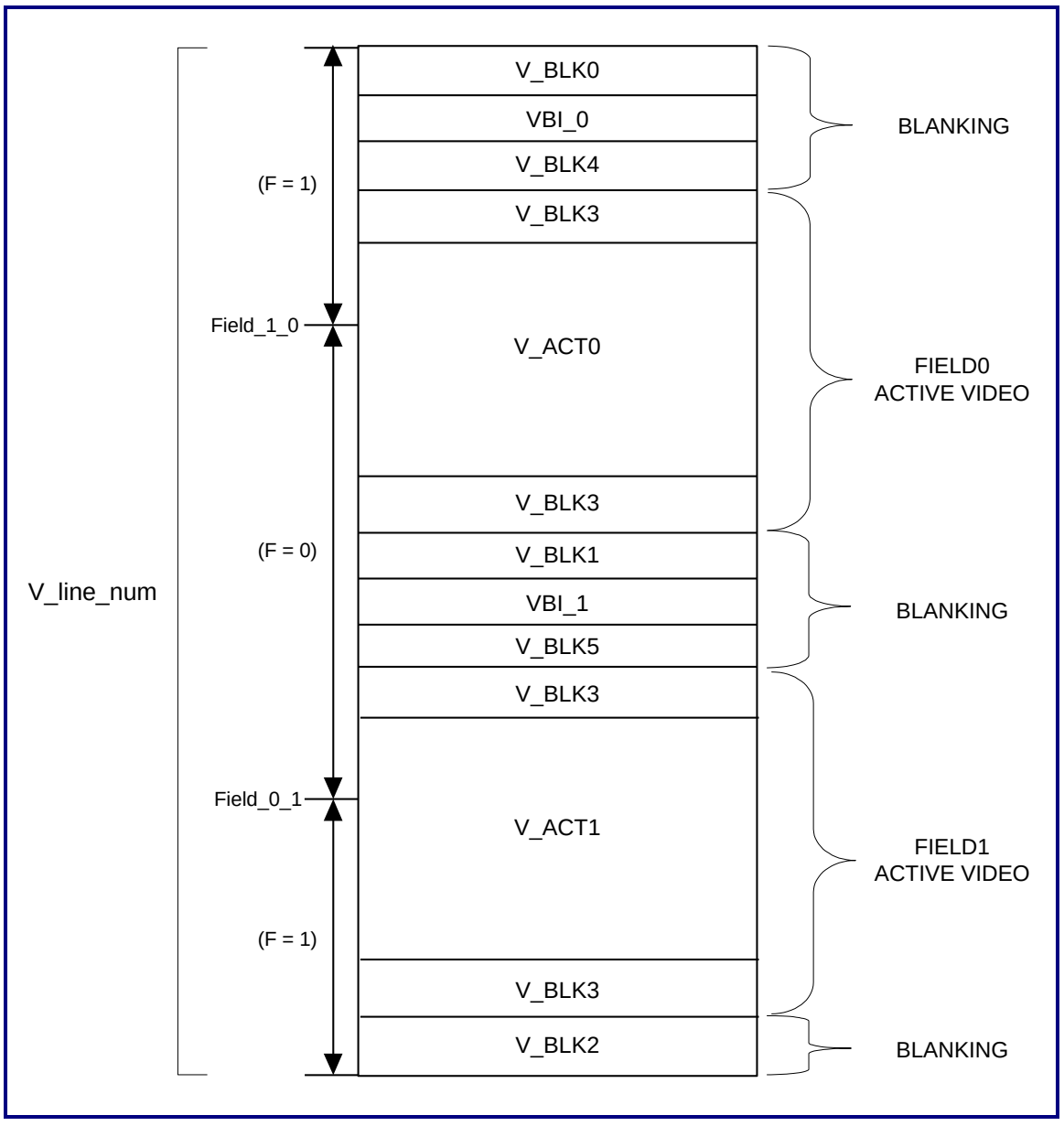


Figure 8-1. Vertical Timing of BT.656/BT.1120

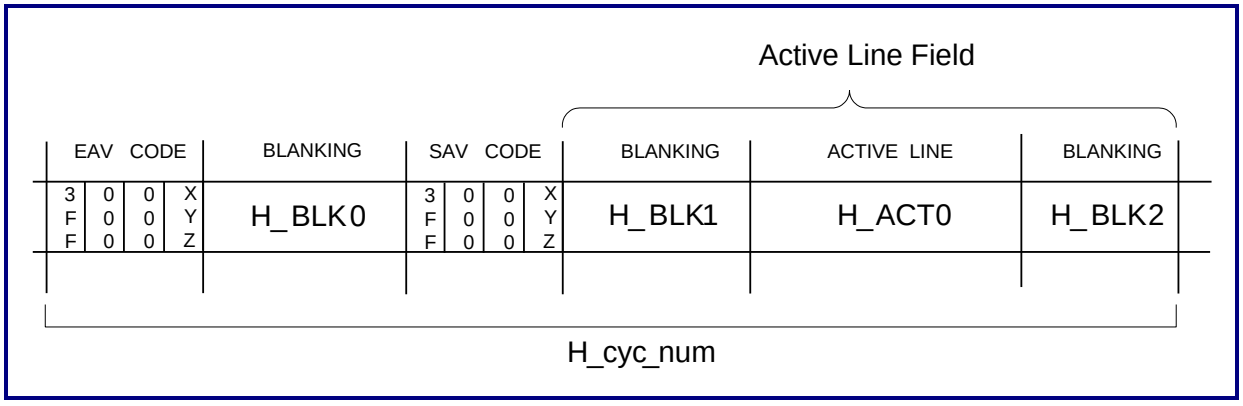


Figure 8-2. Horizontal Timing of BT.656/BT.1120

Table 8-21. TV Parameter 1 (Address = 0x0204)

Bit	Name	Type	Description
[23:12]	V_line_num	R/W	Vertical line number Take NTSC for example, the value of this parameter must be set to 525, and it must be set to 625 for the PAL system.
[11:0]	H_cyc_num	R/W	Horizontal cycle number. Take NTSC for example, the value of this parameter must be set to 1716, and it must be set to 1728 for the PAL system.

8.1.3.4.3 TV Field Polarity Parameters (Address = 0x020C)

Table 8-22. TV Field Polarity Parameter 2 (Address = 0x020C)

Bit	Name	Type	Description
[23:12]	Field_0_1	R/W	The field indicates which line the value of “F” changes from 0 to 1.
[11:0]	Field_1_0	R/W	The field indicates which line the value of “F” changes from 1 to 0.

8.1.3.4.4 TV Vertical Blank Parameters (Address = 0x0210)

Table 8-23. TV Vertical Blank Parameter 3 (Address = 0x0210

Bit	Name	Type	Description
[21:12]	V_Blk1	R/W	This field indicates the number of line of the V_BLK1 region.
[11:10]	-	-	Reserved
[9:0]	V_Blk0	R/W	This field indicates the number of line of the V_BLK0 region.

8.1.3.4.5 TV Vertical Blank Parameters (Address = 0x0214)

Table 8-24. TV Vertical Blank Parameter 4 (Address = 0x0214)

Bit	Name	Type	Description
[21:12]	V_Blk3	R/W	This field indicates the number of line of the V_BLK3 region.
[11:10]	-	-	Reserved
[9:0]	V_Blk2	R/W	This field indicates the number of line of the V_BLK2 region.

8.1.3.4.6 TV Vertical Active Parameters (Address = 0x0218)

Table 8-25. TV Vertical Active Parameter 5 (Address = 0x0218)

Bit	Name	Type	Description
[23:12]	V_Act1	R/W	This parameter indicates the active number of line of field 1.
[11:0]	V_Act0	R/W	This parameter indicates the active number of line of field 0.

8.1.3.4.7 TV Horizontal Blank Parameters (Address = 0x021C)

Table 8-26. TV Horizontal Blank Parameter 6 (Address = 0x021C)

Bit	Name	Type	Description
[21:12]	H_Blk1	R/W	This field shows the number of cycle of the H_BLK1 region.
[11:10]	-	-	Reserved
[9:0]	H_Blk0	R/W	This field shows the number of cycle of the H_BLK0 region.

8.1.3.4.8 TV Horizontal Blank Parameters (Address = 0x0220)

Table 8-27. TV Horizontal Blank Parameter 7 (Address = 0x0220)

Bit	Name	Type	Description
[31:10]	-	-	Reserved
[9:0]	H_Blk2	R/W	This field shows the number of cycle of the H_BLK2 region.

8.1.3.4.9 TV Horizontal Active Parameters (Address = 0x0224)

Table 8-28. TV Horizontal Active Parameter 8 (Address = 0x0224)

Bit	Name	Type	Description
[31:12]	-	-	Reserved
[11:0]	H_Act0	R/W	This field shows the active number of cycle of each line.

8.1.3.4.10 TV Vertical Blank Parameters (Address = 0x0228)

Table 8-29. TV Vertical Blank Parameter 9 (Address = 0x0228)

Bit	Name	Type	Description
[21:12]	V_Blk5	R/W	This field shows the number of line of the V_BLK5 region.
[11:10]	-	-	Reserved
[9:0]	V_Blk4	R/W	This field shows the number of line of the V_BLK4 region.

8.1.3.4.11 TV VBI Parameters (Address = 0x022C)

The VBI data is fetched from the frame buffer 3. The LCDImage3FrameBase register must be set to the start address of the VBI data.

Users are suggested to set the VBI frame buffer base address at the register, 03CH. The VBI data format is used with the YCbCr422 format for data packing. If the TV image input format is progressive, the VBI data must be progressive in the frame buffer. If the TV input format is interlaced, the LCD controller will treat the VBI data format as interlace.

Table 8-30. TV VBI Parameter 10 (Address = 0x022C)

Bit	Name	Type	Description
[21:12]	VBI_1	R/W	This field shows the line number of the VBI_1 region.
[11:10]	-	-	Reserved
[9:0]	VBI_0	R/W	This field shows the line number of the VBI_0 region. VBI_0 must be the same as VBI_1.

8.1.3.5 LCD Image Parameters

8.1.3.5.1 PiP Sub-Picture1 Position (Address = 0x0300)

When PiP is enabled, the following restrictions apply:

- PiPBlend < 257

The outputted pixel value can be divided into the following formula:

Output pixel value = (High_Image x PiPBlend_h + Low_Image x (256 - PiPBlend))/256

Note: The priority of Image is High_Image > Low_Image.

Table 8-31. PiP Blending Parameters (Address = 0x0300)

Bit	Name	Type	Description
[23:16]	PiPBlend_d	R/W	Alpha blending level of an image with the down priority The valid value is 0 ~ 254, and 255 represents 256.
[15:8]	PiPBlend_h	R/W	Alpha blending level of an image with a higher priority The valid value is 0 ~ 254, and 255 represents 256.
[7:0]	PiPBlend_l	R/W	Alpha blending level of an Image with the lower priority The valid value is 0 ~ 254, and 255 represents 256.

8.1.3.5.2 PiP Sub-Picture1 Position Parameters (Address = 0x0304)

When PiP is enabled, the following restrictions apply:

- PiP1HPos + PiP1HDim < Horizontal resolution of a panel
- PiP1VPos + PiP1VDim < Vertical resolution of a panel

Table 8-32. PiP Sub-Picture1 Position Parameters (Address = 0x0304)

Bit	Name	Type	Description
28	PiP_Update	W	When this bit is set, the LCD controller will start to update all PiP dimensions and position registers. This bit will be automatically cleared when registers are updated.
[26:16]	PiP1Hpos	R/W	Specify the horizontal position of the sub-picture1 of a PiP window The reference original position is the left boundary of the LCD screen. The valid value is ranging from 1 to 2047. Note: When the register bit, "En1YCbCr422", is enabled, the value must be an odd number.
[15:11]	-	-	Reserved
[10:0]	PiP1Vpos	R/W	Specify the vertical position of the sub-picture1 of a PiP window The reference original position is the up boundary of the LCD screen. The valid value is ranging from 1 to 2047. When the output format is set as interlace, this field must be set as even number.

8.1.3.5.3 PiP Sub-picture1 Dimension Parameters (Address = 0x0308)

Table 8-33. PiP Sub-picture1 Dimension Parameters (Address = 0x0308)

Bit	Name	Type	Description
[31:27]	-	-	Reserved
[26:16]	PiP1Hdim	R/W	Specify the horizontal dimension of the sub-picture1 of a PiP window Notes: When the register bit, "En1YCbCr422", is enabled, the value must be an even number. The horizontal dimension *bpp should be the multiple of 32bit or 64bit when the AXI/AHB bus is 32bits or 64bits, respectively
[15:11]	-	-	Reserved
[10:0]	PiP1Vdim	R/W	Specify the vertical dimension of the sub-picture1 of a PiP window

8.1.3.5.4 PiP Image Priority Parameters (Address = 0x0314)

The image priority register controls the priority of each image when two are overlapped.

Table 8-34. PiP Image Priority Parameters (Address = 0x0314)

Bit	Name	Type	Description
[7:6]	-	-	Reserved
[5:4]	-	-	Reserved

Bit	Name	Type	Description
[3:2]	Img1_priority	R/W	Image1 priority 00: Low priority 01: High priority 10: Reserved 11: Reserved
[1:0]	Img0_priority	R/W	Image0 priority 00: Low priority 01: High priority 10: Reserved 11: Reserved

8.1.3.5.5 PiP/PoP Image Format1 (Address = 0x0318)

The image format register controls the format of each image when two, three, or four images are appeared in the same frame.

Table 8-35. PiP/PoP Image Format1 (Address = 0x0318)

Bit	Name	Type	Description
[31:30]	-	-	Reserved
[29:28]	-	-	Reserved
[27:26]	RGBTYPE1	R/W	Image1 RGB format This field shows the Image1 input format when BppFifo1 is 100. 00: RGB 565 01: RGB 555 10: RGB 444 11: Reserved
[25:24]	RGBTYPE0	R/W	Image0 RGB format This field shows the Image0 input format when BppFifo0 is 100. 00: RGB 565 01: RGB 555 10: RGB 444 11: Reserved
[23:22]	-	-	Reserved

Bit	Name	Type	Description
[21:20]	-	-	Reserved
[19:18]	Endian1	R/W	Image1 endian control 00: Little-endian byte and little-endian pixel 01: Big-endian byte and big-endian pixel 10: Little-endian byte and big-endian pixel (WinCE) 11: Reserved
[17:16]	Endian0	R/W	Image0 endian control 00: Little-endian byte and little-endian pixel 01: Big-endian byte and big-endian pixel 10: Little-endian byte and big-endian pixel (WinCE) 11: Reserved
15	-	-	Reserved
[14:12]	-	-	Reserved
11	-	-	Reserved
[10:8]	-	-	Reserved
7	En1YCbCr422	R/W	Image1 YCbCr422 mode 0: Image0 is the RGB format. 1: Image0 is the YCbCr422 format.
[6:4]	BppFifo1	R/W	Image1 pixel format The value indicates the data format of image1 (Bit per pixel). 000: 1bpp 001: 2bpp 010: 4bpp 011: 8bpp 100: 16bpp 101: 24bpp 110: ARGB8888 111: ARGB1555
3	En0YCbCr422	R/W	Image0 YCbCr422 mode 0: Image0 is the RGB format. 1: Image0 is the YCbCr422 format.

Bit	Name	Type	Description
[2:0]	BppFifo0	R/W	Image0 pixel format The value indicates the data format of image0 (Bit per pixel). 000: 1bpp 001: 2bpp 010: 4bpp 011: 8bpp 100: 16bpp 101: 24bpp 110: ARGB8888 111: ARGB1555

8.1.3.5.6 PiP/PoP Image Format2 (Address = 0x031C)

Table 8-36. PiP/PoP Image Format2 (Address = 0x031C)

Bit	Name	Type	Description
5	-	-	Reserved
4	-	-	Reserved
3	-	-	Reserved
2	-	-	Reserved
1	Img1Format	R/W	Image1 format The value decides whether the format of image1 in the frame buffer is “Progress” or “Interlace.” This bit is effective when CCIREn is enabled. 0: Image1 format is “Progress”. 1: Image1 format is “Interlace”.
0	Img0Format	R/W	Image0 format The value decides whether the format of image0 in the frame buffer is “Progress” or “Interlace.” This bit is effective when CCIREn is enabled. 0: Image0 format is “Progress”. 1: Image0 format is “Interlace”.

8.1.3.5.7 PiP Color Key1 (Address = 0x0320)

The pixel of PiP sub-picture1 will be transparent when the pixel value equals to the value of ColorKey.

Table 8-37. PiP Color Key1 (Address = 0x0320)

Bit	Name	Type	Description
24	ColorKey1En	R/W	Color key for PiP image1 function control 0: Disable 1: Enable
[23:0]	ColorKey1	R/W	Color key for PiP image1 value

8.1.3.5.8 LCD Palette RAM Write Accessing Port (Address = 0x0A00 ~ 0x0BFC)

Example of address 0x0A00 (For index values 0 and 1)

Table 8-38. Palette RAM Write Accessing Port (Address = 0x0A00 ~ 0x0BFC)

Bit	Name	Type	Description
[31:16]	PaletteEty1	W	Not available for the YCbCr mode due to the fact that the data in the frame buffer represent the raw YCbCr components and cannot be remapped. This 16bit value is used to remap the color when the index value equals to '1'. The format is described below: D[31:27] forms MSB of the red component and the remaining LSBs will be filled with '0'. D[26:21] forms MSB of the green component and the remaining LSBs will be filled with '0'. D[20:16] forms MSB of the blue component and the remaining LSBs will be filled with '0'.
[15:0]	PaletteEty0	W	This 16bit value is used to remap the color when the index value equals to '0'. The operation is similar to the previous one. D[15:11] forms MSB of the red component and the remaining LSBs will be filled with '0'. D[10:5] forms MSB of the green component and the remaining LSBs will be filled with '0'. D[4:0] forms MSB of the blue component and the remaining LSBs will be filled with '0'.

8.1.3.6 Scaler Control Registers

The scaler cannot work when TV is enabled and the input and output format is not progressive.

If users turn on TV and the input format or output format is interlace, please use the 14H register to perform simple scaling.

8.1.3.6.1 Horizontal Resolution Register of Scaler Input (Address = 0x1100)

Table 8-39. Horizontal Input Resolution Register of Scaler Input (Address = 0x1100)

Bit	Name	Type	Description
[31:12]	-	-	Reserved
[11:0]	Hor_no_in	R/W	**Horizontal Resolution of Input Image** Actual horizontal resolution of input image = Hor_no_in + 1 This field identifies the horizontal resolution of an input image and must be programmed to a non-zero value before the scaler becomes active. The value is between 1 and 2047.

8.1.3.6.2 Vertical Resolution Register of Scaler Input (Address = 0x1104)

Table 8-40. Vertical Input Resolution of Scaler Input (Address = 0x1104)

Bit	Name	Type	Description
[31:12]	-	-	Reserved
[11:0]	Ver_no_in	R/W	**Vertical Resolution of Input Image** Actual vertical resolution of input image = Ver_no_in + 1 This field identifies the vertical resolution of an input image and must be programmed as a non-zero value before the scaler becomes active. This field is programmed to the required number of ver_no_in minus 1. The value is between 1 and 2047.

8.1.3.6.3 Horizontal Resolution Register of Scaler Output (Address = 0x1108)

Table 8-41. Horizontal Output Resolution Register of Scaler Output (Address = 0x1108)

Bit	Name	Type	Description
[31:14]	-	-	Reserved
[13:0]	Hor_no_out	R/W	**Horizontal Resolution of Output Image** Actual horizontal resolution of output image = Hor_no_out This field identifies the horizontal resolution of an output image and must be programmed as a non-zero value before the scaler becomes active. The value is between 1 and 4096.

8.1.3.6.4 Vertical Resolution Register of Scaler Output (Address = 0x110C)

Table 8-42. Vertical Output Resolution Register of Scaler Output (Address = 0x110C)

Bit	Name	Type	Description
[31:14]	-	-	Reserved
[13:0]	Ver_no_out	R/W	**Vertical Resolution of Output Image** Actual vertical resolution of output image = Ver_no_out This field identifies the vertical resolution of an output image and must be programmed as a non-zero value before the scaler becomes active. The value is between 1 and 4096.

8.1.3.6.5 Miscellaneous Control Register (Address = 0x1110)

Table 8-43. Miscellaneous Control Register (Address = 0x1110)

Bit	Name	Type	Description
[31:14]	-	-	Reserved
[8:6]	Fir_sel	R/W	Scaling ratio selection of the first-stage scaler 000: Bypass the first stage scaler 001: Scaling down to 1/2x1/2 010: Scaling down to 1/4x1/4 011: Scaling down to 1/8x1/8 100: Scaling down to 1/16x1/16 101: Scaling down to 1/32x1/32 110: Scaling down to 1/64x1/64 111: Scaling down to 1/128x1/128
5	-	-	Reserved
[4:3]	Hor_inter_mode	R/W	Horizontal interpolation mode 00: Nearly bilinear mode The following modes are only available on down-scaling: 01: Threshold nearly bilinear mode 10: Most neighborhood mode 11: Reserved

Bit	Name	Type	Description
[2:1]	Ver_inter_mode	R/W	Vertical interpolation mode 00: Nearly bilinear mode The following modes are only available on down-scaling: 01: Threshold nearly bilinear mode 10: Most neighborhood mode 11: Reserved
0	Bypass_mode	R/W	This bit identifies the second stage scaler bypass mode and bypasses the scaler. This bit must be set in the initial state and should not be changed once it is set. This bit must be cleared to activate the scaler.

8.1.3.6.6 Horizontal High Threshold Register (Address = 0x1114)

Table 8-44. Horizontal High Threshold Register (Address = 0x1114)

Bit	Name	Type	Description
[31:9]	-	-	Reserved
[8:0]	Hor_high_th	R/W	Horizontal up-scaling high threshold register The up-scaling coefficient is highly limited by this threshold value in the horizontal direction. The real value equals to Hor_high_th/256.

8.1.3.6.7 Horizontal Low Threshold Register (Address = 0x1118)

Table 8-45. Horizontal Low Threshold Register (Address = 0x1118)

Bit	Name	Type	Description
[31:9]	-	-	Reserved
[8:0]	Hor_low_th	R/W	Horizontal up-scaling low threshold register The up-scaling coefficient is low limited by this threshold value in the horizontal direction. The real value equals to Hor_low_th/256.

8.1.3.6.8 Vertical High Threshold Register (Address = 0x111C)

Table 8-46. Vertical High Threshold Register (Address = 0x111C)

Bit	Name	Type	Description
[31:9]	-	-	Reserved

Bit	Name	Type	Description
[8:0]	Ver_high_th	R/W	Vertical up-scaling high threshold register The up-scaling coefficient is high limited by this threshold value in the vertical direction. The real value equals to Ver_high_th/256.

8.1.3.6.9 Vertical Low Threshold Register (Address = 0x1120)

Table 8-47. Vertical Low Threshold Register (Address = 0x1120)

Bit	Name	Type	Description
[31:9]	-	-	Reserved
[8:0]	Ver_low_th	R/W	Vertical up-scaling low threshold register The up-scaling coefficient is low limited by this threshold value in the vertical direction. The real value equals to Ver_low_th/256.

8.1.3.6.10 Scaler Resolution Parameters (Address = 0x112C)

The parameters, scal_hor_num and scal_ver_num, are calculated as shown below, where ver_no_in/hor_no_in represents the input image resolution register value if the first scaler is bypassed or the output image resolution of the first scaler minus one if the first scaler is involved.

Scaling up:

Scal_ver_num = [(ver_no_in+1)/ver_no_out]x256

Scal_hor_num = [(hor_no_in+1)/hor_no_out]x256

If the scaling-up ratio is 2x2, Scal_ver_num and Scal_hor_num are ignored.

Scaling down:

Scal_ver_num = [mod((ver_no_in+1)/ver_no_out)]x256/ver_no_out

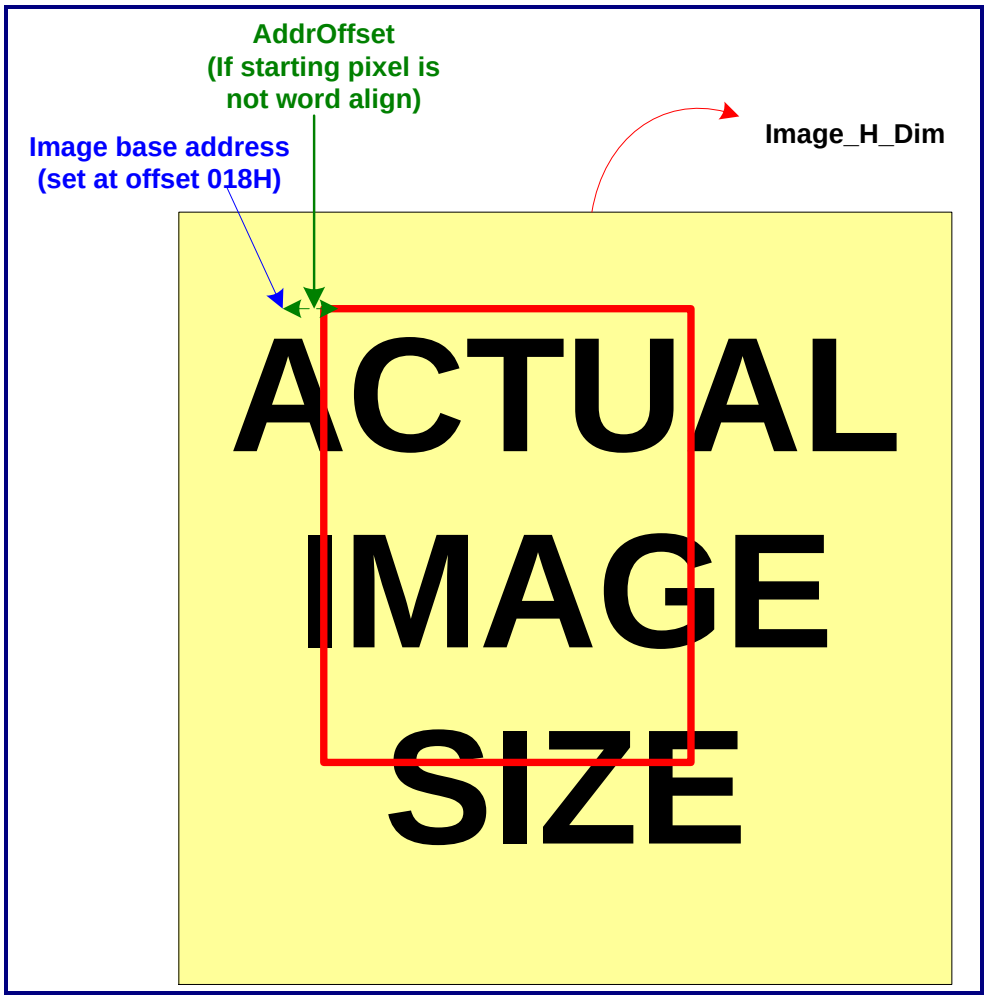
Scal_hor_num = [mod((hor_no_in+1)/hor_no_out)]x256/hor_no_out

Table 8-48. Scaler Resolution Parameters (Address = 0x112C)

Bit	Name	Type	Description
[31:16]	-	-	Reserved
[15:8]	Scal_hor_num	R/W	The second-stage initial numerator of the scaler coefficient in the horizontal direction
[7:0]	Scal_ver_num	R/W	The second-stage initial numerator of the scaler coefficient in the vertical direction

8.1.3.7 Virtual Screen

The LCD controller provides the hardware horizontal/vertical scrolling function. The image base address (018H) and address offset register (1500H) must be updated if the image is scrolled. The horizontal/vertical width of the scrolled image is set in PL (0100H)/LF (0104H). The image stored in the frame buffer should be larger than the screen size of the LCD panel.



8.1.3.7.1 Virtual Screen Control Register (Address = 0x1500)

Table 8-49. Virtual Screen Control Register (Address = 0x1500)

Bit	Name	Type	Description
[31:21]	-	-	Reserved
20	VirtualScreenEn	R/W	Virtual screen enable 0: Disable 1: Enable
[19:17]	-	-	Reserved
[16:12]	PixelOffset	R/W	In 32bit AMBA data bus, the starting address of a screen equals to “LCDImage0FrameBase” + “PixelOffset * bppfifo0”. The unit of PixelOffset is pixel.
[11:0]	Image_H_Dim	R/W	The horizontal dimension of an image stored in the horizontal frame buffer (Pixels-per-line) Users must ensure that each line start is four-word alignment.

8.1.3.7.2 Virtual Screen 1 Control Register (Address = 0x1504)

Table 8-50. Virtual Screen 1 Control Register (Address = 0x1504)

Bit	Name	Type	Description
[31:21]	-	-	Reserved
20	VirtualScreen1En	R/W	Virtual screen enable 0: Disable 1: Enable
[19:17]	-	-	Reserved
[16:12]	PixelOffset1	R/W	In 32bit AMBA data bus, the starting address of a screen equals to “LCDImage1FrameBase” + “PixelOffset1 * bppfifo1”. The unit of PixelOffset1 is pixel.
[11:0]	Image1_H_Dim	R/W	The horizontal dimension of an image stored in the horizontal frame buffer (Pixels per line)

8.2 TVE DAC

8.2.1 General Description

TVE DAC is a Digital-to-Analog Converter (DAC). Its segmented architecture can minimize the glitch energy and DNL so that its scheme of 6 thermometers plus 4 binary codes provides an excellent dynamic performance. This DAC uses the resistor and voltage reference as a pair to define the current LSB level. The current cells are capable of delivering a maximum full-scale output current at 34.67mA, which is directly fed into the load resistor without any external output buffer.

8.2.2 Features

- Guaranteed monotonic
- Supports analog current output
- Built-in power-down and standby modes
- Supports differential current output
- DNL ±1 LSB (Max.)
- INL ±2 LSB (Max.)
- Estimated output settling time: 34.5ns

8.2.3 Programming Model

Please refer to "CVBS DAC Control Register" in Chapter 10 for details.

Chapter 9

Video Processing

This chapter contains the following sections:

9.1 H.264 Encoder Engine

9.1.1 Video Encoder

The H.264 encoder is a high-profile H.264 video encoder. This encoder includes the hardware engines to accelerate the computation-intensive tasks, such as the motion estimation, forward transform, backward transform, quantization, inverse quantization, motion compensation, and entropy coding. The H.264 encoder is controlled by CPU through the AHB slave interface. By initializing the control register of the encoder, one slice of the H.264 encoding procedure is automatically done by the H.264 encoder. Thus, CPU is released from the timing-critical tasks in video encoding. To improve the quality of the video source and to reduce the bandwidth consumption of an SoC system, the de-interlace and de-noise functions are included.

9.1.1.1 Features

- Includes de-interlace and de-noise preprocessor
- Supports field coding and frame coding
- Supports 4x4 and 8x8 transform coding
- Supports 4x4 and 8x8 Intra predictions
- Supports user-defined quantization matrix
- Supports I, P, and B slice encodes
- Supports CAVLC and CABAC entropy coding
- Supports mono (4:0:0) encoding
- Supports multiple slices in one frame
- Supports EOI (Encoding interested rectangle area)
- Supports MB-level QP controlled by encoding bits and image variance

9.1.1.1.1 Motion Estimation

The Motion Estimation (ME) engine is used to find the best motion vector and best macroblock partition that have the best block match. A modified full-search algorithm is used to scan all defined search ranges in the full-pixel precision. A quarter-pixel precision refinement will be performed for the best full-pixel motion vector of each partition block in a marcoblock.

9.1.1.1.2 Intra Prediction

The intra prediction engine is used to find the best prediction mode to encode the marcoblock in the intra mode and the cost value.

9.1.1.1.3 Entropy Coding

The entropy coding unit supports the H.264 CAVLC and CABAC coding modes.

9.1.1.1.4 TQ and IQIT

This module is used to generate the quantized transform coefficient for entropy coding. It also performs the inverse quantization and inverse transform to generate reconstructed data for deblocking engine.

9.1.1.1.5 Deblocking Filter

A filter is applied to each inverse transformed macroblock for reducing the blocking distortion. The filter is applied to the vertical and horizontal edges of 4x4 or 8x8 blocks in a macroblock with five pre-defined strength modes.

9.1.1.1.6 System Controller

The local memory controller arbitrates local memory access requests from CPU, DMA, and the H.264 encoder. The local memory controller supports several addressing modes to achieve optimum utilization of the local memory bandwidth.

9.2 MPEG4/JPEG Codec Engine

9.2.1 General Description

MPEG4/JPEG is capable of accelerating the multimedia image process and the video related applications, such as MPEG4 and JPEG. This codec includes the hardware engines to accelerate the computation-intensive tasks, such as the motion estimation, DCT/IDCT, quantization/ inverse quantization, and motion compensation. it is controlled by CPU through the AHB slave interface. By initializing the control registers of the codec, the motion estimation calculation task for an entire of 16x16 or 8x8 block is automatically done by the codec. The DCT/IDCT, quantization/inverse quantization, AC/DC prediction, zigzag scan, and VLC/VLD calculation tasks are also done automatically for a macro-block by the codec. Thus, CPU is released from the timing-critical tasks in video encoding. To improve the quality of the decoded output, a deblocking post-filter is included, which improves the subjective and objective qualities, especially when the quantization step size is large. This optional feature will be turned on or off by using the hardware configuration or through the software programming. The standard-cell-based approach allows users to quickly integrate the codec into the user SoC designs. The codec includes additional features, which are described in the feature list.

The DMA controller performs the task of transferring data between the system memory and the local memory of the MPEG4 codec. The DMA controller includes one AHB master interface and one AHB slave interface. The AHB master interface allows the DMA controller to access data by the AHB bus. The AHB slave interface is used to program the control register of the DMA controller from the AHB bus.

9.2.2 Features

MPEG4/JPEG codec

Function

- Compliant with MPEG-4 (ISO/IEC 14496-2) simple profile L0 ~ L3 standards, including resolutions of Sub QCIF, QCIF, CIF, VGA, 4CIF, and D1 at 30fps with steps of 16 units
- Compliant with JPEG (ISO/IEC 10918-1) base-line standard
- Includes hardware engine for motion estimation/compensation, DCT/IDCT, quantization/ inverse quantization, AC/DC prediction, and variable length coding/decoding
- Supports local memory controller for local memory shared by DMA master and other codec blocks
- Supports DMA controller for data transfers between system memory and local memory
- Supports automatic power-down mechanism
- Motion estimation search range: -16 ~ +15.5 (Optionally -32 ~ +31.5) with half-pixel accuracy
- Supports 4 MV and unrestricted MV
- Supports constant bit rate and variable bit rate
- Error resilient tools: Encoder supports resynchronization marker and header extension code; decoder supports resynchronization marker, header extension code, data partition and RVLC
- Supports short video header (H.263 baseline)
- Supports H.263/MPEG/JPEG quantization methods
- JPEG supported features:
  - Supports 4 user-defined Huffman tables (2AC and 2DC)
  - Supports 4 programmable quantization tables
  - Supports interleave and non-interleave scans
  - Supports YUV 4:4:4, 4:2:2, and 4:2:0 formats
  - Supports image size of up to 64k×64k
- Supports full-duplex operation (e.g. video phone and video conference) by software switching encoding and task decoding on the same hardware
- Supports embedded RISC to minimize loading of host CPU (Optional)
- Supports deblocking post-filter to enhance decoded output quality (Optional)

Performance

- Supports MPEG4 simple profile encoding up to D1 at 30fps at codec clock speed under 72MHz

9.2.2.1 Motion Estimation

The Motion Estimation (ME) unit performs motion estimation for the entire search window based on a fast search algorithm. By reading commands in the local memory, the motion estimation of a macroblock will be automatically completed.

9.2.2.2 DCT/IDCT

The DCT/IDCT unit is responsible for Discrete Cosine Transform (DCT) and Inverse Discrete Cosine Transform (IDCT). The IDCT unit uses the same hardware resources as the ones for DCT unit. The results of the IDCT unit are compliant with the IEEE 1180-1990 specification; the results of DCT unit are passed to the Quantization unit at the encoding phase. The results of IDCT unit are passed to the MC unit at the decoding phase.

9.2.2.3 Quantization/Inverse Quantization

The Quantization/Inverse Quantization unit supports the H.263/MPEG/JPEG quantization methods. The Quantization results are passed to the AC/DC prediction unit at the encoding phase, and the inverse quantization results are passed to the IDCT unit at the decoding phase.

9.2.2.4 AC/DC Prediction

The AC/DC prediction unit supports both the MPEG-4 AC/DC prediction method and JPEG DC prediction method. The results of the AC/DC prediction are passed to the Zigzag Scan unit at the encoding phase and the results of the inverse AC/DC prediction are passed to the Inverse Quantization unit at the decoding phase.

9.2.2.5 Zigzag Scan

The Zigzag Scan unit supports all JPEG zigzag methods. The Zigzag results of (Run, level) pairs are passed to the VLC unit during the encoding phase and the inverse zigzag results are passed to the DC Prediction unit during the decoding phase.

9.2.2.6 Variable Length Coding/Decoding

The Variable Length Coding/Decoding (VLC/VLD) unit supports the JPEG user-defined Huffman codes. The VLC results are the final compressed bitstreams at the encoding phase and the VLD results are passed to the Zigzag unit at the decoding phase.

9.2.2.7 Motion Compensation

The Motion Compensation (MC) unit is an engine responsible for motion compensation task. At the encoding phase, this unit subtracts the interpolated block from the current block and sends the residual block to DCT. At the decoding phase, this unit adds the interpolated block to the IDCT output block to get the reconstructed block.

9.2.2.8 Local Memory Controller

The local memory controller arbitrates the local memory access requests from the CPU, DMA, and MPEG4/JPEG codec engines. A multi-bank memory architecture is used for the local memory to improve the memory bandwidth. CPU has the highest access priority, followed by DMA and then codec. The local memory controller supports the following addressing modes for codec operations:

- Transfer a linear block or 2D block from the local memory to codec
- Transfer the codec output to a linear or 2D block in the local memory
- Transfer a 2D block to the ME unit in the circular buffer addressing mode

9.3 Scalar

9.3.1 General Description

Scalar (SCAR) is in charge of resizing and enhancing the video data. By DMA, scaler reads video data on AMBA AXI, then, writes back the result. With the resizing ability, users can size-down or size-up an image to the necessary resolution or perform the aspect ratio conversion. The window clipping function can clip the interested region from the image before or after sizing down. Moreover, the luminance compensation, sharpness, and color saturation controls can help make an image more pleasing to the human eyes.

9.3.2 Features

Image Cropping (SC, Source Cropping)

- Supports front-end image source cropping function
- Supports input formats:
  YCbCr 4:2:2/RGB 555/RGB565/RGBG888/ARGB888
- Supports RGB to YCbCr transformation
- Supports maximum image size of 4096x2048

Scaling and de-noise (SD)

- Supports 1-D false color suppression
- Supports 1-D de-noise
- Supports four output resolution per scan line, including scaling-down/up function on two output resolutions and bypass another two
- Supports smooth and sharpness
- Supports ARGB888 scaling
- Supports field scaling (Top/Bottom offsets)
- Supports frame to field extraction

Mask

- Supports up to 8 windows in one frame for each channel
- User programmable 8 palette color types for mask

9.3.3 Memory Map/Register Definition

9.3.3.1 Mapping of Usual Special-purpose Registers

Channel	Symbol	Offset	Access	Description
0	Sub-channel	0x0000 ~ 0x0027	R/W	Sub-channel setting
	FCS	0x0028 ~ 0x002F	R/W	False color suppression setting
	Mask	0x0030 ~ 0x006F	R/W	Mask function setting
	DN	0x0070 ~ 0x00B7	R/W	De-noise/Scaling setting
	TC	0x0160 ~ 0x017F	R/W	Target cropping setting
	DMA	0x0180 ~ 0x01A7	R/W	DMA function setting
	LL	0x01A8 ~ 0x01AF	R/W	Link-list setting
SC		0x5100 ~ 0x5217	R/W	Source cropping setting
Palette		0x5300 ~ 0x533F	R/W	Palette setting
Global		0x5400 ~ 0x547F	R/W	global setting
Status		0x5500 ~ 0x55FF	R/W	status
Link-List		0x40000 ~ 0x47FFF	R/W	Link-list memory

9.4 3D De-interlace and De-noise

9.4.1 General Description

The de-interlacing process is an important stage, which converts ordinary TV interlaced sequences into progressive sequences for displaying on the progressive devices (Such as computers, LCD, plasma display, and projection TV).

The de-noise process is important to enhance the scene quality and reduce the encoded bit rate in network. In a real-time IP camera system, the de-noise function is very useful to eliminate the Gauss-distribution noise, so that the encoder can make lower effort to encode the source scene. On the other hand, the encoded files will be smaller and will reduce the bit rate in a network.

The 3D de-interlace and de-noise module adopts the pixel-based measurement to perform intra and inter field interpolations, depending on the motion state which the pixel belongs to the motion or stationary. Its goal is to generate a high-quality progressive frame.

9.4.2 Features

- Built-in de-noise filter to eliminate high-frequency noise of field difference
- Supports four-field and two-field motion detections
- Supports 60i to 30p/60p function
- Supports frame size of up to 1920x1080 resolution
- Supports only packet 422 format

9.5 2D Graphics Accelerator Engine

The 2D Graphics Accelerator is a graphics subsystem for the acceleration of 2D content. Graphics Accelerator is capable of targeting multiple types of surfaces, seamlessly move raster images, accelerating drawing of lines and filled rectangles and performing alpha blending in numerous blending modes.

9.5.1 Features

- 2D drawing engine
  - Pixel Drawing
  - Line Drawing (Any direction)
  - Filled Rectangles
  - Quadrilaterals
  - Triangles Gouraud Shaded

- Color formats
  - RGBA8888
  - ARGB8888
  - RGB565
  - RGBA5551
  - RGBA4444
  - Lum8 (Grayscale)
  - A8
  - A1
  - YUV (YUY2, UYVY)

- Full Alpha Blending
  - Porter-Duff Blending modes
  - DirectFB Blending Modes
  - Configurable Clipping Rectangle
  - Destination Color keying
  - Subsampling Font Rasterization

- Blitter
  - High performance DMA Blitter moves raster images in memory
  - Image format Color Conversion on-the-fly
  - Stretching on x and y axis
  - Source Color Keying
  - Supports 90°, 180°, and 270° rotations, Mirror-X Mirror-Y

9.6 Image Signal Processor (ISP)

9.6.1 General Description

ISP is a powerful image signal processor, which captures Bayer format data from sensor and outputs YCbCr data to video capture after color reproduction, noise reduction, and image enhancement processes. Besides it also provides 3A statistic and histogram for users to develop 3A algorithm.

9.6.2 Features

ISP contains the following features:

- Supports maximum resolution of up to 1920x1080
- Supports 1/2-lane MIPI, subLVDS(Note), subLVDS serial interface and 12bit parallel interface
- Supports sensor image in 8/10/12/14bits Bayer raw format
- Supports front-end image source cropping function
- Supports image processing functions as follows:
  - Compressed WDR sensor data up to 20bits
  - Local tone mapping

- Dynamic range correction
- Optical black-level cancellation
- Asymmetric lens shading compensation
- Crosstalk cancellation
- Dynamically defect pixel correction
- 2D noise reduction
- Flexible gamma look-up table
- 3x3 color correction matrix
- 3x3 programmable matrix to convert color space from RGB to YCbCr
- Brightness and contrast function
- Flexible nonlinear hue and saturation adjustment
- Powerful skin color correction
- Anti-jaggy
- Sharpness
- Chroma suppression

- Provides AE, AWB, AF statistic, and configurable histogram
- Uses YCbCr 4:2:2 data output format

Note: The specification is specified to support the subLVDS electrical signals of SONY and OV senor.

9.7 3D Noise Reduction

9.7.1 Description

Noise reduction is an important part to enhance both the human subjective quality and codec compression efficiency. However, in the spatial domain, an edge will be blurred after smoothing during the general noise reduction process. Multi-resolution noise reduction analyzes the edge information under different SNRs with the same scene. It is more robust to remove noise and keep the edge.

The ghost effort occurs during the temporal noise reduction. The local temporal noise estimator adapts the filter variance to fit different environment requirements. The temporal edge detector suppresses the filtered temporal difference under moving sharp edge scene. These two modules can ease the ghost effort while reducing the temporal noise.

3D noise reduction module will first adopt the multi-resolution noise reduction in the spatial domain; then, suppress the temporal noise with the Crawford filter. Its goal is to generate a stable noiseless video sequence with the detailed features.

9.7.2 Features

- Supports multi-resolution Noise Reduction in spatial domain
- Supports luma de-noise strength learning in time domain
- Supports luma-edge detection in time domain
- Supports frame size up to 1920x1080
- Only supports packed YCbCr 4:2:2 format

Chapter 10

Audio Interface

This chapter contains the following sections:

10.1 Audio Input/Output

The audio input/output interface of GM8136S/GM8135S is used to connect to the off-chip audio codec for playing and recording the audio data. GM8136S/GM8135S provides two audio input/output interfaces, I2S0, I2S1. I2S0 and I2S1 are used to input the audio of multiple channels.

The I^{2}S interface supports the serial data format up to 256bits in length. Both the left-/right-channel can support 128bit data. It can receive 16-channel 16bit data from the video decoder.

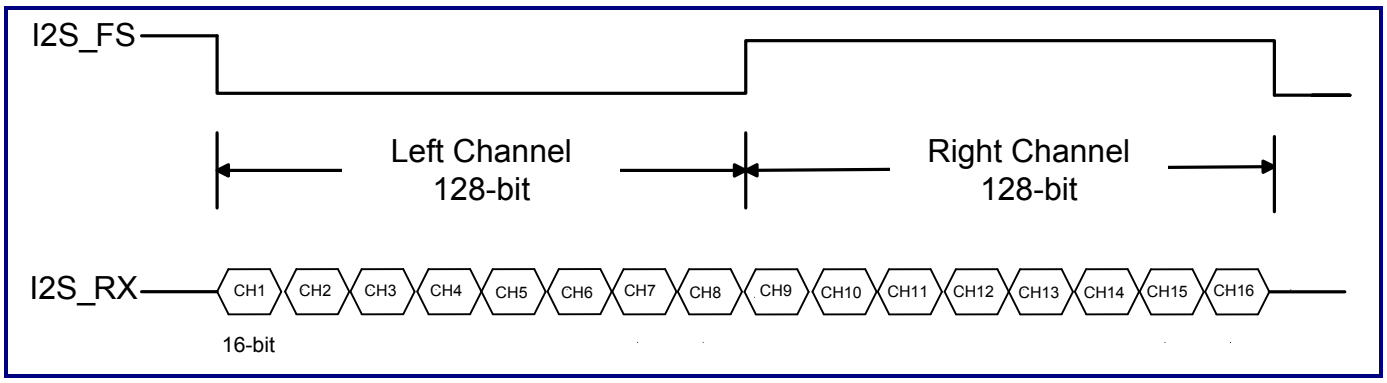


Figure 10-1. GM8136S/GM8135S I^{2}S 16-channel Audio Data Input Mode

10.2 I^{2}S Controller

10.2.1 General Description

The I^{2}S controller is a synchronous serial port interface that allows the host processor to serve as a master or a slave. It can be connected to various devices by using the serial protocol. It supports the Synchronous Serial Port (SSP) from Texas Instruments, the Serial Peripheral Interface (SPI) from Motorola, MICROWIRE from National Semiconductor, and I^{2}S from Philips. The serial data formats range from 4bits to 32bits in length. The SSP controller can directly use the on-chip DMA to transfer data between the external serial device and the system memory without being intervened by the processor.

10.2.2 Features

- Supports TI SSP, Motorola SPI, National Semiconductor MICROWIRE, and Philips I^2S
- Independent SSP clock for easy generation of bit clock
- Supports master and slave modes
- Internally or externally controlled serial bit clock
- Internally or externally controlled frame/sync.
- Programmable frame/sync. polarity
- Programmable serial bit clock polarity, phase, and frequency
- Programmable serial bit data sequence (MSB or LSB first)
- Programmable I^2S format including zero-bit padding and right/left justification
- Programmable threshold interrupt of transmit/receive FIFO
- Independently programmable interrupt enable/disable

10.2.3 Programming Model

10.2.3.1 SSP Control Registers

Table 10-1. SSP Control Registers

Offset	Type	Description	Reset Value
0x00	R/W	SSP control register 0	0x0000_010C
0x04	R/W	SSP control register 1	0x0007_8000
0x08	R/W	SSP control register 2	0x0000_0002
0x0C	R	SSP status register	0x0000_0002
0x10	R/W	SSP interrupt control register	0x0000_2100
0x14	RC	SSP interrupt status register	0x0000_0008
0x18	R/W	SSP data register	N/A

10.2.3.2 Register Descriptions

The following subsections provide the detailed information of the SSP control registers.

10.2.3.2.1 SSP Control Register 0 (Offset = 0x00)

This register controls the generation and behavior of frame/sync.

Table 10-2. SSP Control Register 0 (Offset = 0x00)

Bit	Name	Type	Description
[31:18]	-	-	Reserved and read as zeroes
17	FSFDBK	R/W	This bit indicates that fs_in is the internal feedback from fs_out_r. If the master mode is specified, the controller will need fs_in for reference. If this bit is set to '1', fs_in will be the internal feedback from fs_out_r. If this bit is set to '0', fs_in will be inputted from input pin.
16	SCLKFDBK	R/W	This bit indicates that sclk_in is the internal feedback from sclk_out_r. If the master mode is specified, the controller will need sclk_in for reference. If this bit is set to '1', sclk_in will be the internal feedback from sclk_out_r. If this bit is set to '0', sclk_in will be inputted from the input pin.
15	SPIFSPO	R/W	Frame/Sync. polarity for the SPI mode It is different to FSPO in bit 5. If this bit is set to '1', the frame/sync. will be regarded as active high. If this bit is set to '0', the frame/sync. will be regarded as active low. This bit will take effect only when the SPI frame format is specified.
[14:12]	FFMT	R/W	Frame format This field defines the pre-defined frame format according to the following encodings: 000: Texas Instrument Synchronous Serial Port (SSP) 001: Motorola Serial Peripheral Interface (SPI) 010: National Semiconductor MICROWIRE 011: Philips I^2S 100: Reserved 101: Reserved 110 ~ 111: Not defined If these bits are set, the SSP operation will be unpredictable.

Bit	Name	Type	Description
11	FLASH	R/W	This bit indicates that the current application is SPI Flash. This bit will take effect only when the SPI frame format is specified.
10	-	-	Reserved
[9:8]	FSDIST	R/W	Frame/Sync. and data distance This field will only take effect when the I^2S frame format is specified. These bits define the relationship between the frame/sync. and data bits. If these bits are set to zero, the first bit of serial data will be valid in the first cycle of frame/sync. If non-zero is specified, FSDIST defines the number of SCLK cycles between the first bit of the received valid data and frame start. For the standard I^2S frame format, these bits must be set to '1'.
7	LBM	R/W	Loop-back mode If this bit is set to '1', the transmitted data will be internally connected to the received data. This is used for self-test only. If this bit is set to '0', SSP will operate in the normal mode and the transmitted/received data are independent.
6	LSB	R/W	Bit sequence indicator If this bit is set to '0', the Most Significant Bit (MSB) of a data word will be transmitted or received first. If this bit is set to '1', the Least Significant Bit (LSB) of a data word will be transmitted or received first. This bit only takes effect when the I^2S frame format is specified.
5	FSPO	R/W	Frame/Sync. polarity If this bit is set to '0', the frame/sync. will be treated as active high. If this bit is set to '1', the frame/sync. will be treated as active low. This bit only takes effect when the I^2S or MWR frame format is specified.
4	FSJSTFY	R/W	Data justify This bit will be valid only when the I^2S frame format is specified. If this bit is set to '0' and Padding Data Length (PDL) in the control bit is not zero, the number of zeroes specified in PDL will be appended in the back of serial data. If this bit is set to '1', the padding data will be in the front of serial data.
[3:2]	OPM	R/W	Operation mode If the SSP, SPI, or MICROWIRE frame format is specified, these bits specify the operation modes as follows: 00, 01: Slave mode 10, 11: Master mode If the I^2S frame format is specified, these bits define the operation modes as follows: 11: Master stereo mode 10: Master mono mode 01: Slave stereo mode 00: Slave mono mode

Bit	Name	Type	Description
1	SCLKPO	R/W	SCLK polarity This bit will only take effect if the Motorola SPI frame format is specified. When this bit is set to ‘0’, SCLK will remain low when SSP is idle. When this bit is set to ‘1’, SCLK will remain high when SSP is idle.
0	SCLKPH	R/W	SCLK phase This bit will only take effect if the Motorola SPI frame format is specified. This bit defines the relationship between SCLK and frame/sync. When this bit is set to ‘0’, SCLK will start toggling after one SCLK cycle when frame/sync. is activated. When this bit is set to ‘1’, SCLK will start running after half an SCLK cycle when frame/sync. is activated.

10.2.3.2.2 SSP Control Register 1 (Offset = 0x04)

This register defines the clock divider and data length of a transfer. Table 10-3 shows the detailed format of SSP control register 1.

Table 10-3. SSP Control Register 1 (Offset = 0x04)

Bit	Name	Type	Description
[31:24]	PDL	R/W	Padding data length
23	-	-	Reserved and read as zeroes
[22:16]	SDL	R/W	Serial data length This field defines the bit length of a transmitted/received data word. The actual data length equals to bit number plus 1.
[15:0]	SCLKDIV	R/W	SCLK divider This field defines the serial clock divider.

When the I^2S frame format is specified, PDL will define the bit length of the padding bits in the front/back of data word. If PDL is set to ‘0’, no padding bit will be inserted. If the value specified is a non-zero value, the actual number of zeroes will be inserted/appended in the front/back of data word.

If the National Semiconductor MICROWIRE frame format is specified, PDL will define the bit length of the first phase (Transmitted in the master mode and received in the slave mode), and the actual data length will equal to this register plus 1. The maximum value of this register should not exceed 31 if the National Semiconductor MICROWIRE is specified. The minimum value of this register should be more than 1.

The minimum value of SDL should not be less than 4. The maximum value of this register refers to the width of the configured FIFO.

The frequency of the internally generated serial clock is controlled by SCLKDIV[15:0]. The frequency of the internally generated serial clock is determined by the following formula:

$$f_{SCLK} = \frac{f_{SSPCLK}}{2 \times (SCLKDIV + 1)}$$

Note: f_{SSPCLK} is SSP0_CLK/SSP1_CLK/SSP2_CLK.

10.2.3.2.3 SSP Control Register 2 (Offset = 0x08)

Table 10-4. SSP Control Register 2 (Offset = 0x08)

Bit	Name	Type	Description
[31:12]	-	-	Reserved and read as zeroes
[11:10]	FSOS	R/W	Frame/Sync. Output Select When the SPI frame format is specified and these two bits are set as: 00: Frame/Sync. output from the output port, fs_out_r 01: Frame/Sync. output from the output port, fs1_out_r 10: Frame/Sync. output from the output port, fs2_out_r 11: Frame/Sync. output from the output port, fs3_out_r These bits will take effect only if the SPI frame format is specified.
9	FS	R/W	Frame Sync. output This bit controls the frame/sync. output level in the SPI frame format. If this bit is set to '0', the FS output will be "LOW". If this bit is set to '1', the FS output will be "HIGH".

Bit	Name	Type	Description
8	TXEN	R/W	Transmit function enable When the I2S, SPI, or MICROWIRE frame format is specified, the transmit and receive functions will work independently. This bit controls the transmit function. 1: Transmit function is enabled. 0: Transmit function is disabled. For the SPI and MICROWIRE frames format, this bit can only be changed when the SSP controller is idle.
7	RXEN	R/W	Receive function enable When the I2S, SPI, or MICROWIRE frames format is specified, the transmit and receive functions will work independently. This bit controls the receive function. 1: Receive function is enabled. 0: Transmit function is disabled. For the SPI and MICROWIRE frames format, this bit can only be changed when the SSP controller is idle.
6	SSPRST	R/W	SSP reset The state machine for software reset of the SSP controller Writing ‘1’ to this bit will reset of the SSP controller. Writing ‘0’ has no effect.
5	ACCRST	R/W	AC-link cold reset enable If this bit is written as ‘1’ and the AC-link frame format is specified, the ac97_resetn_r bit will be active to reset the codec. Once the reset is completed, this bit will be cleared to ‘0’. Writing ‘1’ to this bit will clear the SSP_EN bit.
4	ACWRST	R/W	AC-link warm reset enable If this bit is written as ‘1’ and the AC-link frame format is specified, the SSP controller will assert FS high to enter a warm reset when SSP is disabled. Once the reset is completed, this bit will be cleared to ‘0’. Writing ‘1’ to this bit when the SSP controller is enabled takes no effect.
3	TXFCLR	W	Transmit FIFO clear This is a write-only bit. Writing ‘1’ to this bit will clear all data in the transmit FIFO. Writing ‘0’ has no effect. Reading this bit always gets a value of zero.
2	RXFCLR	W	Receive FIFO clear This is a write-only bit. Writing ‘1’ to this bit will clear all data in the receive FIFO. Writing ‘0’ has no effect. Reading this bit always gets a value of zero.

Bit	Name	Type	Description
1	TXDOE	R/W	Transmit data output enable This bit will only be valid when the SSP slave mode is specified. In the multi-slave system, it is possible for an SSP master to broadcast a message to all slaves within the system while ensuring that only one slave drives data on its serial output line. In such a system, the rxd lines from the multi-slave system could be tied together. To operate in such system, TXDOE may be cleared if the SSP slave is not supposed to drive the txd_r line. When TXDOE is set, SSP will drive data on the transmit data line. When TDOE is not set, SSP will not drive data on the transmit data line. If the I^2S frame format is specified, this bit defines the receiving (Recording) mode (0) or the simultaneous transmitting/receiving (Playing/Recording) mode (1).
0	SSPEN	R/W	SSP enable If this bit is set to '1', the SSP controller will start transmitting/receiving data, if possible. If this bit is set to '0', the serial data will stop toggling. When the SSPEN bit is cleared in an active operation, SSP will stop transmitting or receiving data after the current word had been transmitted or received.

Writing '1' to SSPRST will reset the SSP state machine even if the normal transmission or reception is in progress. Usually, this bit will be written when software stops the SSP controller by setting SSPEN to '0'. For the slave mode of the SSP controller, the serial clock is controlled by the external master and will be stopped under certain circumstance, and caused the SSP state machine to halt. Writing '1' to this bit puts the SSP controller back to the idle state when carrying out the operations.

10.2.3.2.4 SSP Status Register (Offset = 0x0C)

Table 10-5. SSP Status Register (Offset = 0x0C)

Bit	Name	Type	Description
[31:18]	-	-	Reserved and read as zeroes
[17:12]	TFVE	R	Transmit FIFO valid entries This field indicates the number of entries in the transmit FIFO waiting to be transmitted.
[11:10]	-	R	Reserved and read as zero
[9:4]	RFVE	R	Receive FIFO valid entries This field indicates the number of entries in the receive FIFO waiting for DMA or host processor to be read.
3	-	R	Reserved and read as zero
2	BUSY	R	Busy indicator If this bit is set to '1', it indicates that SSP is transmitting and/or receiving data. If this bit is set to '0', it indicates that SSP is idle or disabled.

Bit	Name	Type	Description
1	TFNF	R	Transmit FIFO not full This bit will be ‘1’ when the transmit FIFO is available for DMA or the host processor writes. TFNF will be cleared to ‘0’ when FIFO is completely full.
0	RFF	R	Receive FIFO full This bit will be set to ‘1’ when the receive FIFO is full. RFF will be cleared to ‘0’ when the DMA controller or host processor reads data from the receive FIFO.

10.2.3.2.5 Interrupt Control Register (Offset = 0x10)

Table 10-6. Interrupt Control Register (Offset = 0x10)

Bit	Name	Type	Description
[31:17]	-	R	Reserved and read as zeroes
[16:12]	TFTHOD	R/W	Transmit FIFO threshold If the valid data in transmit FIFO is equal to or less than the actual threshold, the DMA request and/or the interrupt will be asserted. If this bit is set to ‘0’, the interrupt will occur when the transmit FIFO is completely shifted out.
[11:7]	RFTHOD	R/W	Receive FIFO threshold If the valid data in the receive FIFO is equal to or greater than the actual threshold, the DMA request and/or the interrupt will be asserted. If this bit is set to ‘0’, the interrupt will be disabled.
6	-	-	Reserved
5	TFDMAEN	R/W	Transmit DMA request enable If this bit is set to ‘1’, the DMA request will be issued when the transmit FIFO threshold (TFLETH is set to ‘1’) is hit. If this bit is set to ‘0’, no DMA request will be issued.
4	RFDMAEN	R/W	Receive DMA request enable If this bit is set to ‘1’, the DMA request will be issued when the receive FIFO threshold (RFGETH is set to ‘1’) is hit. If this bit is set to ‘0’, no DMA request will be issued.
3	TFTHIEN	R/W	Transmit FIFO threshold interrupt enable If this bit is set to ‘1’, the interrupt will be issued when the valid entries in the transmit FIFO are less than or equal to the threshold value.
2	RFTHIEN	R/W	Receive FIFO threshold interrupt enable If this bit is set to ‘1’, the interrupt will be issued when the valid entries in the receive FIFO are greater than or equal to the threshold value.
1	TFURIEN	R/W	Transmit FIFO underrun interrupt enable If this bit is set to ‘1’, the transmit FIFO underrun will cause SSP to assert interrupt. If this bit is set to ‘0’, the interrupt will be masked even when the transmit FIFO underrun happens.

Bit	Name	Type	Description
0	RFORIEN	R/W	Receive FIFO overrun interrupt enable If this bit is set to ‘1’, the receive FIFO overrun will cause SSP to assert interrupt. If this bit is set to ‘0’, the interrupt will be masked even when the receive FIFO overrun occurs.

10.2.3.2.6 Interrupt Status Register (Offset = 0x14)

Table 10-7. Interrupt Status Register (Offset = 0x14)

Bit	Name	Type	Description
[31:5]	-	R	Reserved and read as zero
4	-	-	Reserved
3	TFTHI	R/RC	Transmit FIFO threshold interrupt TFTHOD will be set to a non-zero value and this bit will be set to ‘1’ when the transmit FIFO is equal to or less than the threshold value. If the valid entries in the transmit FIFO are larger than TFTHOD, this bit will be automatically cleared. TFTHOD will be set to ‘0’ and this bit will be set to ‘1’ when the last data in the transmit FIFO is completely shifted out. This bit will never be cleared to ‘0’ until user reads this bit.
2	RFTHI	R	Receive FIFO threshold interrupt If the receive FIFO is equal to or greater than the threshold value, this bit will be set to ‘1’. This bit will be cleared when the condition is removed.
1	TFURI	RC	Transmit FIFO underrun interrupt If the transmit logic tries to retrieve data from empty transmit FIFO, this bit will be set to ‘1’. This bit will never be cleared to ‘0’ until users read this bit.
0	RFORI	RC	Receive FIFO overrun interrupt If the receive logic tries to receive data and the receive FIFO is full, this bit will be set to ‘1’. This bit will never be cleared to ‘0’ until users read this bit.

10.2.3.2.7 SSP Transmit/Receive Data Register (Offset = 0x18)

The SSP controller separates transmit FIFO and receive FIFO, which are 32bit wide and can be separately configured. The transmitted and received data occupy the same address space. The write operation writes data into the transmit FIFO and the read operation reads data from the receive FIFO. If the size of a serial data length is less than 32bits, data will be automatically right-justified at the time of reception or transmission.

10.3 Audio ADDA and Video DAC Controller

10.3.1 General Description

This controller is designed for setting the audio ADDA and Video DAC parameters.

10.3.2 Programming Model

10.3.2.1 Summary of ADDA Controller Registers

Table 10-8 provides a summary of the ADDA controller registers.

Table 10-8. Summary of ADDA Controller Registers

Offset	Name	Reset Value
+0x00	Audio Codec Control Register 0 (ACR0)	0x18FC_0000
+0x04	Audio Codec Control Register 1 (ACR1)	0x0000_0000
+0x08	Audio Codec Control Register 2 (ACR2)	0x0000_0003
+0x0C	Audio Codec Control Register 3 (ACR3)	0x0000_5000
+0x10	Audio Codec Control Register 4 (ACR4)	0x3404_0000
+0x14	CVBS DAC Control Register (DACCR)	0x0000_0007

10.3.2.2 Register Descriptions

The following subsections describe the ADDA controller registers in details.

The abbreviation below represents the access types used throughout the register descriptions:

- R/W: Read/Write

10.3.2.2.1 Audio Codec Control Register 0 (ACR0, Offset = 0x00)

Table 10-9. Audio Codec Control Register 0 (ACR0, Offset = 0x00)

Bit	Name	Type	Description
31	-	-	Reserved
30	MIC_MODE	R/W	Microphone input 1: Single-ended input 0: Pseudo differential input
29	DEPOP	R/W	Pop sound control (Test only) 1: Enable 0: Disable
28	MICPD	R/W	Microphone bias power-down control 1: Power-down mode 0: Normal mode
27	SPK_PD	R/W	Power-down control for the speaker driver 1: Power-down mode 0: Normal mode
[26:23]	-	-	Reserved
22	DACPD	R/W	DAC power-down control 1: Power-down mode 0: Normal operation
21	-	-	Reserved
20	ADCPD	R/W	ADC power-down control 1: Power-down mode 0: Normal operation
19	-	-	Reserved
18	FEPD	R/W	ADC front-end power-down control 1: Power-down mode 0: Normal operation
17	CPS	R/W	Chopper stabilized control 1: Chopper stabilized on 0: Chopper stabilized off
16	MICSEL	R/W	ADC digital microphone control 1: Enable the digital microphone 0: Disable the digital microphone

Bit	Name	Type	Description
[15:14]	DAIMODE	R/W	Digital audio interface mode 00: Parallel data mode 01: I^2S serial data mode 10: Right-justified serial data mode (For DAC) 11: Left-justified serial data mode (For DAC)
[13:12]	DEEMPH	R/W	DAC digital de-emphasis filter control signal 11: Set the 48kHz de-emphasis filter on with MCLK = 256x 10: Set the 44.1kHz de-emphasis filter on with MCLK = 256x 01: Set the 32kHz de-emphasis filter on with MCLK = 384x 00: Set the de-emphasis filter off
[11:8]	MCLK DAC Mode	R/W	Define the ratio of MCLK frequency and DAC sampling rate 1111: MCLK = 256 x sampling clock 1110: MCLK = 128 x sampling clock 1101: MCLK = 272 x sampling clock 1100: MCLK = 136 x sampling clock 1011: MCLK = 1024 x sampling clock 1010: MCLK = 512 x sampling clock 1001: MCLK = 1088 x sampling clock 1000: MCLK = 544 x sampling clock 0111: MCLK = 384 x sampling clock 0110: MCLK = 250 x sampling clock 0101: MCLK = 500 x sampling clock 0100: MCLK = 1500 x sampling clock 0011: MCLK = 1536 x sampling clock 0010: MCLK = 768 x sampling clock 0001: MCLK = 750 x sampling clock 0000: MCLK = 375 x sampling clock
[7:4]	MCLK ADC Mode	R/W	Define the ratio of MCLK frequency and ADC sampling rate 1111: MCLK = 256 x sampling clock 1110: MCLK = 128 x sampling clock 1101: MCLK = 272 x sampling clock 1100: MCLK = 136 x sampling clock 1011: MCLK = 1024 x sampling clock 1010: MCLK = 512 x sampling clock

Bit	Name	Type	Description
			1001: MCLK = 1088 x sampling clock
			1000: MCLK = 544 x sampling clock
			0111: MCLK = 384 x sampling clock
			0110: MCLK = 250 x sampling clock
			0101: MCLK = 500 x sampling clock
			0100: MCLK = 1500 x sampling clock
			0011: MCLK = 1536 x sampling clock
			0010: MCLK = 768 x sampling clock
			0001: MCLK = 750 x sampling clock
			0000: MCLK = 375 x sampling clock
3	REBACK	R/W	The DAC input digital data is back to the ADC digital output. In the serial I^2S mode: 1: Enable the reback function 0: Normal mode
2	TESTPD	R/W	DAC anti-pop sound scheme control 0: Enable the anti-pop sound mechanism 1: Disable the anti-pop sound mechanism
1	MASTER	R/W	Audio interface control 0: Slave mode 1: Master mode
0	ADDA_RESET	R/W	This bit resets the ADDA audio codec Active high

10.3.2.2.2 Audio Codec ADC Control Register 1 (ACR1, Offset = 0x04)

Table 10-10. Audio Codec ADC Control Register 1 (ACR1, Offset = 0x04)

Bit	Name	Type	Description
[31:25]	-	-	Reserved
[24:18]	ADV	R/W	ADC 7bit digital volume gain control ADV[6:0] equals: 1111111: +30dB 0101111: -50dB (Minimum gain) Each step is 1.0-dB resolution. For 0 dB, ADV[6:0] = 7'b1100001 X: Reserved

Bit	Name	Type	Description
[17:12]	-	-	Reserved
[11:6]	IV	R/W	ADC gain control 6bit line input volume control 111111: +36dB 000000: -27dB There are a total of 64 steps, each step is 1.0dB resolution. For 0 dB, IV[5:0] = 011011
5	-	-	Reserved
4	IM	R/W	ADC input mute 1: Enable the input mute 0: Normal operation
3	-	-	Reserved
2	MICBT	R/W	MICIN boost control 1: +20dB gain 0: 0dB gain
1	-	-	Reserved
0	INSEL	R/W	ADC input signal selection 1: Select MICIN 0: Select LINEIN

10.3.2.2.3 Audio Codec Control Register 2 (ACR2, Offset = 0x08)

Table 10-11. Audio Codec Control Register 2 (ACR2, Offset = 0x08)

Bit	Name	Type	Description
[31:28]	ALCHOLT	R/W	Set the ALC hold time before gain increased 0000: 0μs 0001: 250μs 0010: 500μs 0011: 1.0ms (Time is doubled with every step.)
[27:24]	ALCDCYT	R/W	Set the ALC decay time (1.0dB per step) 0000: 104μs 0001: 208μs 0010: 416μs (Time is doubled with every step.)

Bit	Name	Type	Description
[23:20]	ALCATKT	R/W	Set the ALC attack time (1.0dB per step) 0000: 63.3μs 0001: 125μs 0010: 250μs (Time is doubled with every step.)
[19:16]	ALCTL	R/W	Set the ALC target level 1111: -4.0dBFS 1110: -6.0dBFS (2.0dB per step) 0000: -34dBFS
[15:13]	ALCMIN	R/W	Set the PGA minimum gain 111: -6.0dB 110: -9.0dB 101: -12dB 100: -15dB 011: -18dB 010: -21dB 001: -24dB 000: -27dB
[12:10]	ALCMAX	R/W	Set the PGA maximum gain 111: +36dB 110: +30dB 101: +24dB 100: +18dB 011: +12dB 010: +6.0dB 001: +0dB 000: -6.0dB
[9:7]	ALCNGTH	R/W	ALC noise gate threshold 111: -36dB 110: -42dB 101: -48dB (6dB per step) 000: -78dB
6	ALCZC	R/W	ALC update gain on zero-cross function control 1: Enable 0: Disable

Bit	Name	Type	Description
5	ALC	R/W	ALC function select 1: Enable 0: Disable
4	HPF	R/W	ADC high pass filter control 1: Enable 0: Disable
3	ADCDWA	R/W	ADC modulator DWA mode control (When ADCDEM = '0x0', this register can be controlled.) 1: Enable 0: Disable
2	ADCDEM	R/W	ADC modulator DEM mode control 1: Enable 0: Disable
[1:0]	ADCGAIN	R/W	ADC digital modulator gain setting in the test mode 00: +1.0dB 01: +1.5dB 10: +2.0dB 11: +4.0dB (Default)

10.3.2.2.4 Audio Codec Control Register 3 (ACR3, Offset = 0x0C)

Table 10-12. Audio Codec Control Register 3 (ACR3, Offset = 0x0C)

Bit	Name	Type	Description
[31:27]	-	-	Reserved
[26:25]	SPV	R/W	Speaker output power compensation (Only for test) 00: +0dB (Default) 01: +0.5dB 10 : +1.0dB 11 : +1.5dB
[24:20]	-	-	Reserved
[19:18]	PWM	R/W	Debug data input pin (For test only)
17	-	-	Reserved

Bit	Name	Type	Description
16	HM	R/W	DAC digital mute SCF DAC is mute. 1: Enable 0: Disable
15	SDMDWA	R/W	DAC modulator DWA mode control (When SDMDEM = '0', this bit is valid.) 1: Enable 0: Disable
14	SDMDEM	R/W	DAC modulator DEM mode control 1: Enable 0: Disable
[13:12]	SDMGAIN	R/W	DAC digital modulator gain attenuation in the test mode 00: -1.0dB 01: -2.0dB (Default) 10: -3.0dB 11: -6.0dB
11	-	-	Reserved
[10:9]	CLASDGAIN	R/W	DAC speaker gain attenuation in the test mode 00: -0.1dB 01: -0.13dB (Default) 10: -0.2dB 11: -0.3dB
8	LOPD	R/W	Power-down control for the LOUT driver 1: Power-down mode 0: Normal operation
7	LOM	R/W	Line-out mute control of DAC 1: Enable mute 0: Normal mode
[6:0]	DAV	R/W	7bit DAC digital volume control 1111111: +30dB 0101111: -50dB There are a total of 81 steps, and each step is 1.0dB resolution. For the values smaller than 0101111, the gain is set to -50dB.

10.3.2.2.5 Audio Codec Control Register 4 (ACR4, Offset = 0x10)

Table 10-13. Audio Codec Control Register 4 (ACR4, Offset = 0x10)

Bit	Name	Type	Description
[31:30]	-	-	Reserved
29	PSW_PDn	R/W	Digital core power control 0: Power-down digital core power, VCC11K 1: Normal operation (The ISO_ENABLE must be logic '0'.)
28	ISO_ENABLE	R/W	Isolation cell control 1: Enable the isolation cell to prevent floating signals between the analog and digital interfaces when only powered down the digital circuit. 0: Normal operation
27	-	-	Reserved
[26:24]	IRSEL_SPK	R/W	100: Normal setting Others: Reserved (Only for the RD test)
[23:21]	-	-	Reserved
20	BIAS_SEL	R/W	Bias mode control (Only for RD test) 1: Disable the temperature compensation 0: Enable the temperature compensation (Default)
19	SCFPD	R/W	Power-down control for SCF in LOUT 1: Power-down mode 0: Normal operation
[18:16]	IRSEL	R/W	Reference current control 100: Normal setting
[15:14]	-	-	Reserved
[13:12]	SPKSEL	R/W	SCF DAC to speaker control 00: Mute 01: Reserved 10: DAMIXER = 1, DAMIXINV[1:0] = 00 SCF DAC to BTL speaker is enabled. 11: Reserved
11	-	-	reserved
[10:9]	HPFGATE	R/W	It is for test only. Default: 00
[8:6]	HPFCUT	R/W	High-Pass filter cut-off frequency control Please refer to Table 10-14.

Bit	Name	Type	Description
5	HPFAPP	R/W	High pass filter control. 0: Filter is the first order with a cut-off frequency of 3.7 Hz 1: Filter is the second order with a selectable cut-off frequency via the HPFCUT register
4	CM_EN	R/W	Class-D speaker integrator type selection (Only for testing) , change to 1'b1 for function mode. 1: For common-mode integrator 0: For differential-mode integrator
[3:2]	DT	R/W	Dead-time control (Only for testing), change to 2'b01 for function mode. 00: For testing 01: Normal setting 10: For testing 11: For testing
[1:0]	HYS	R/W	Hysteresis voltage control (Only for testing) , change to 2'b10 for function mode. 00: For testing 01: For testing 10: Normal setting 11: For testing

Table 10-14. ADC HPFCUT Settings

Sample Freq.	8	11.025	12	16	22.05	24	32	44.1	48
000:	82	113	123	82	113	123	82	113	123
001:	102	141	153	102	141	153	102	141	153
010:	131	180	196	131	196	156	131	180	196
011:	163	225	245	163	225	245	163	225	245
100:	204	281	306	204	281	306	204	281	306
101:	261	360	392	261	360	392	261	360	392
110:	327	450	490	327	450	490	327	450	490
111:	408	563	612	408	563	612	408	563	612

10.3.2.2.0 CVBS DAC Control Register (DACCR, Offset = 0x14)

Table 10-15. CVBS DAC Control Register (DACCR, Offset = 0x14)

Bit	Name	Type	Description
[31:11]	-	-	Reserved
[10:8]	CVBS_RSETCTRL	R/W	Internal RSET resistor control RSETSEL[3:0] will be ranging from '1111' to '0000' RSET is ranging from 1.6kΩ to 4.225kΩ.
[7:6]	-	-	Reserved
5	CVBS_RSETSEL	R/W	RSET selection 0: RSET is the internal resistor (3kΩ) 1: RSET is the external resistor
4	CVBS_DET_EN	R/W	Enable auto CVBS plug-in/-out detect function 0: Disable 1: Enable
3	CVBS_STBY	R/W	Standby mode control 0: Normal operation 1: Standby mode
2	CVBS_PSW_PD	R/W	Power switch control for VCC11K Logic high: Power-down mode (Turn off the power switch) Logic low: Normal operation (Turn on the power switch)
1	CVBS_PD	R/W	Power-down control for CVBS 0: Normal operation 1: Power-down mode
0	CVBS_ISO_EN	R/W	Isolation control Logic high: Isolated mode Logic low: Normal operation

Chapter 11

Intelligent Video Engine

This chapter contains the following sections:

11.1 Introduction

11.1.1 Overview

The IVS (Intelligent Video Surveillance) engine is a hardware accelerator for generic intelligent video processing. The engine supports the color space conversion, SAD (Sum of Absolute Differences) computation, integral image generation, image convolution, morphological operations, and image histogram. It is programmed through the APB slave interface. A built-in DMA uses the AXI master interface for data access between the external and internal memories.

11.1.2 Features

- Supports YCbCr to HSI conversion
- Supports YCbCr to RGB conversion
- Supports de-interleaving 4:2:2 input image into two separate planes
- Supports generating integral image
- Supports computing SAD (Sum of Absolute Differences) of 1x1/3x3/5x5 block sizes
- Supports image threshold operations
- Supports any binary operation between two images
- Supports generating image histogram
- Supports 5x5 erosion/dilation
- Supports 3x3 opening/closing
- Supports 5x5 convolution
- Supports 5x5 gradient operators

11.1.3 Modes of Operation

The IVS engine provides two function operation modes. The single-frame mode is used to read one source image and write out at most of six operation results through the AXI bus. Only data path 0 can be activated in this mode. Two-frame mode is used to read two source images and write out one operation result through the AXI bus. Only data path 1 can be activated in this mode.

11.2 Memory Map/Register Definition

11.2.1 Summary of Control Registers

Table 11-1. Summary of Control Registers

Offset	R/W Type	Description	Reset Value
0x00	R/W	Control register	0x0000_0000
0x04	R/W	Parameter register	0x0000_0000
0x08	R/W	Resolution register	0x0000_0000
0x10	R/W	Read channel 0 start address register	0x0000_0000
0x14	R/W	Read channel 0 address offset register	0x0000_0000
0x18	R/W	Read channel 1 start address register	0x0000_0000
0x1C	R/W	Read channel 1 address offset register	0x0000_0000
0x20	R/W	Write channel 0 start address register	0x0000_0000
0x24	R/W	Write channel 1 start address register	0x0000_0000
0x28	R/W	Write channel 2 start address register	0x0000_0000
0x2C	R/W	Write channel 3 start address register	0x0000_0000
0x30	R/W	Write channel 4 start address register	0x0000_0000
0x34	R/W	Write channel 5 start address register	0x0000_0000
0x38	R/W	Write channel 6 start address register	0x0000_0000
0x3C	R/W	Write channel 7 start address register	0x0000_0000
0x40	R/W	Write channel address offset register	0x0000_0000
0x60	RO	Write channel status register	0x0000_0000
0x68	RO	Read channel 0 status register	0x0000_0000
0x6C	RO	Read channel 1 status register	0x0000_0000
0x74	RO	IVS status register	0x0000_0001
0x78	R/W1C	Interrupt status register	0x0000_0000
0x7C	RO	Revision register	0x0002_0000
0x080 ~ 0x098	R/W	Kernel template 0	-
0x0A0 ~ 0x0B8	R/W	Kernel template 1	-
0x0C0 ~ 0x0D8	R/W	Kernel template 2	-
0x0E0 ~ 0x0F8	R/W	Kernel template 3	-
0x100 ~ 0x118	R/W	Kernel template 4	-
0x120 ~ 0x138	R/W	Kernel template 5	-

Offset	R/W Type	Description	Reset Value
0x140 ~ 0x158	R/W	Kernel template 6	-
0x160 ~ 0x178	R/W	Kernel template 7	-
0x0180 ~0x037F	R/W	Stage threshold	-
0x2000 ~0x3FFF	R/W	Weak value	-
0x4000 ~0x7FFF	WO	Feature threshold	-
0x8000 ~0xBFFF	WO	Unit rectangle information	-

11.2.2 Register Definitions

11.2.2.1 Control Register (Offset = 0x00)

Table 11-2. Control Register (Offset = 0x00)

Bit	Name	Type	Description
31	sys_rst	R/W	Software reset 0: No operation/Recover from reset state 1: Reset hardware engine The length of reset interval is completely determined by software.
[30:29]	-	-	Reserved
28	int_mask	R/W	Interrupt mask 0: Enable interrupt 1: Disable interrupt
[27:20]	rch_num	R/W	Select read channel 0 or read channel 1 Bit 20: Corresponding operator of y_en and cbcr_en Bit 22: Corresponding operator of h_en, s_en and i_en Bit 25: Corresponding operator of ii_en and si_en 0: Select read channel 0 1: Select read channel 1 This value takes effect only if rch1_en = 1.
[19:18]	-	-	Reserved
17	cc_en	R/W	Enable the cascaded classifier 0: No operation 1: Generate the cascaded classifier output
[16:14]	-	-	Reserved

Bit	Name	Type	Description
13	lgc_en	R/W	Enable the logic operation 0: No operation 1: Generate the logic operation output
12	sad_en	R/W	Enable SAD 0: No operation 1: Generate SAD
11	-	-	-
10	si_en	R/W	Enable the squared integral image 0: No operation 1: Generate the squared integral image
When casc = 0			
9	ii_en	R/W	Enable the integral image 0: No operation 1: Generate the integral image
When casc = 1			
9	ii_en	R/W	Enable the integral image and cascaded classifier 0: No operation 1: Generate the cascaded classifier output
When yuv_int_n = 0 and rgb_en = 0			
8	i_en		Enable the HSI conversion 0: No operation 1: Generate the intensity in HSI
7	s_en		Enable the HSI conversion 0: No operation 1: Generate the saturation in HSI
6	h_en		Enable the HSI conversion 0: No operation 1: Generate the hue in HSI
When yuv_int_n = 0 and rgb_en = 1			
8	i_en		Enable the RGB conversion 0: No operation 1: Generate the blue value in RGB

Bit	Name	Type	Description
7	s_en		Enable the RGB conversion 0: No operation 1: Generate the green value in RGB
6	h_en		Enable the RGB conversion 0: No operation 1: Generate the red value in RGB
When yuv_int_n = 1 and morp_en = 0			
8	i_en		Enable the convolution 0: No operation 1: Generate the magnitude of the gradient vector
7	s_en		Enable the convolution 0: No operation 1: Generate the Y component of the gradient vector
6	h_en		Enable the convolution 0: No operation 1: Generate the X component of the gradient vector or shifted the convolution sum
When yuv_int_n = 1 and morp_en = 1			
8	i_en		Enable the morphology 0: No operation 1: Generate the morphological operator output
[7:6]	-		Reserved
When hist_en = 0 and yuv_int_n = 0			
5	cbcr_en		Enable the de-interleaving 4:2:2 0: No operation 1: Generate the semi-planar chrominance component
When hist_en = 1 or yuv_int_n = 1			
5	cbcr_en		Enable the histogram 0: No operation 1: Generate the image histogram
When yuv_int_n = 0			
4	y_en		Enable the de-interleaving 4:2:2 0: No operation 1: Generate the semi-planar luminance component

Bit	Name	Type	Description
When yuv_int_n = 1			
4	-		Reserved
3	-	-	Reserved
2	endian		Swap endianess 0: Keep the original data format 1: Swap byte order
1	rch1_en		Enable read channel 1 0: Only read channel 0 can be used. 1: Both read channel 0 and read channel 1 can be used. This bit must be set to '1' when sad_en = '1', lgc_en = '1', cc_en = '1', or rch_num is configured with a non-zero value.
0	ii_start		Start the IVS engine 0: No operation 1: Start the IVS engine This bit will automatically be cleared to '0' after set.

	Byte 0							Byte 7
Original	0x01	0x23	0x45	0x67	0x89	0xAB	0xCD	0xEF
	LSB							MSB
endian = 0	0x01	0x23	0x45	0x67	0x89	0xAB	0xCD	0xEF
endian = 1	0xEF	0xCD	0xAB	0x89	0x67	0x45	0x23	0x01

Figure 11-1. Example of Endian

11.2.2.2 Parameter Register (Offset = 0x04)

Table 11-3. Parameter Register (Offset = 0x04)

Bit	Name	Type	Description
When morp_en = 1			
[31:25]	-	-	Reserved
[24]	lt2min	R/W	Select thresholding output 0: Pixels that are above or equal to the threshold are set to '1' in the output image. Pixels that are less than the threshold are set to '0' in the output image. 1: Pixels that are above or equal to the threshold are written to the output unmodified. Pixels that are less than the threshold are set to '0' in the output image.
[23:16]	threshold	R/W	Threshold for binarization of the pixel
When morp_en = 0			
[31:16]	threshold	R/W	Threshold for binarization of the SAD or gradient magnitude
When lgc_en = 1			
[15:12]	rop	R/W	Raster operation code
When h_en = 1 or s_en = 1 or i_en =1			
[15:12]	tl_num	R/W	Select the kernel template to use
11	casc	R/W	Cascade integral image and classifier function The detailed configuration is listed in Table 11-2.
10	morp_en	R/W	Select morphology The detailed configuration is listed in Table 11-2.
9	rgb_en	R/W	Select RGB conversion The detailed configuration is listed in Table 11-2.
8	hist_en	R/W	Select histogram The detailed configuration is listed in Table 11-2.
7	binary	R/W	Bits per pixel of the source images 0: None or both source images are gray-scale. 1: Only one source image is binary.
6	tile	R/W	Perform convolution by dividing into small image tiles 0: Single pass. Entire source window is processed at once. 1: Multiple pass. Original source window is divided into overlapping image tiles, and processed consecutively.

Bit	Name	Type	Description
5	yuv_int_n	R/W	Source data format selection 0: 4:2:2 YCbCr 1: Planar
When morp_en = 0			
4	-	-	Reserved
[3:2]	sad_range	R/W	SAD search range 0: 1x1 block 1: 3x3 block 2: 5x5 block
When morp_en = 1			
[4:2]	op	R/W	Types of the morphological operator 0: Dilation 1: Closing 2: Erosion 3: Opening 7: NOP (Simple thresholding)
1	swap_cbcr	R/W	Swap Cb/Cr byte order 0: Keep the original data format 1: Swap order of byte 4*n and byte (4*n + 2), 0 ≤ n ≤ 1
0	swap_yc	R/W	Swap luminance/chrominance byte order 0: Keep the original data format 1: Swap order of byte 2*n and byte (2*n + 1), 0 ≤ n ≤ 3

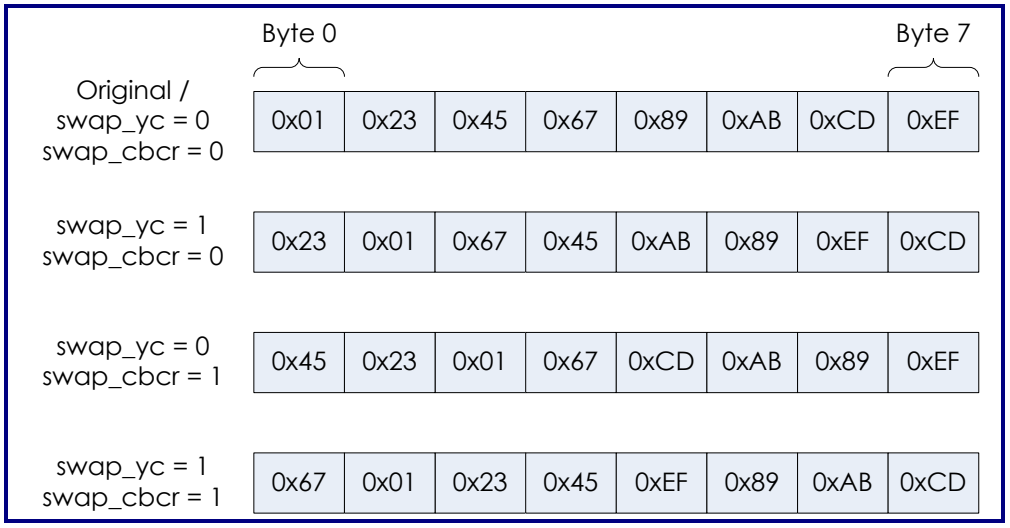


Figure 11-2. Example of swap_yc and swap_cbcr

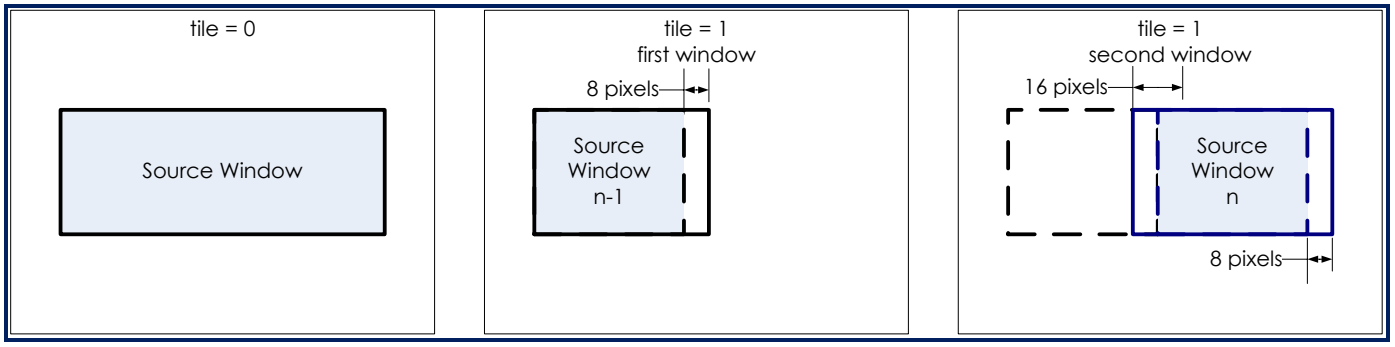


Figure 11-3. Example of Image Tiles

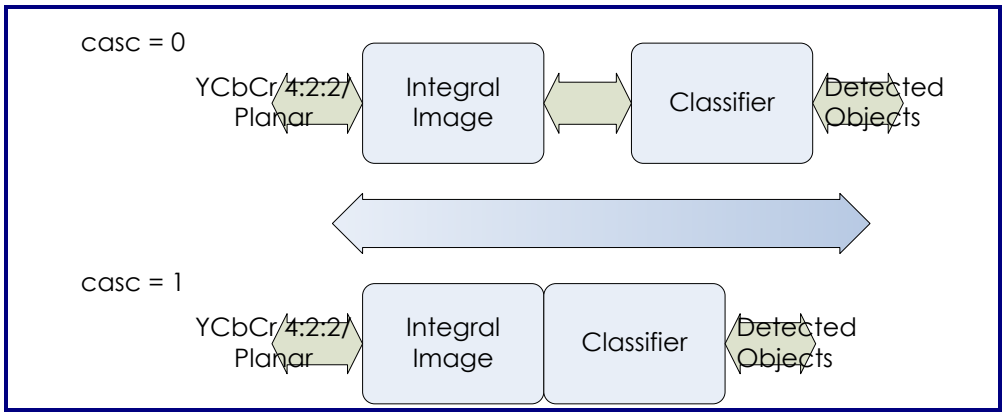


Figure 11-4. Example of casc

11.2.2.3 Resolution Register (Offset = 0x08)

Table 11-4. Resolution Register (Offset = 0x08)

Bit	Name	Type	Description
[31:16]	height	R/W	Source/Destination height in unit of pixel
When binary = 0			
[15:3]	width	R/W	Source/Destination width in unit of 8bytes
When binary = 1			
[15:3]	width	R/W	Source/Destination width in unit of 8pixels
[2:0]	-	-	Reserved

Table 11-5. Maximum Source Width

Operation	Maximum Source Width
De-interleaving YC	Any
HSI conversion	Any
Integral image	2048
SAD	2048
Histogram	Any
Convolution	2048
Morphology	2048
Raster operation	Any
Classifier	320

11.2.2.4 Face Detection Parameter Register (Offset = 0x0C)

Table 11-6. Face Detection Parameter Register (Offset = 0x0C)

Bit	Name	Type	Description
[31:20]	-	-	Reserved
[19:16]	fd_max	R/W	Detect the maximum number of objects 0: 2^{15} ($0 \le fd\_cnt \le 2^{15}$) 1: 2^{14} ($0 \le fd\_cnt \le 2^{14}$) … n: $2^{(15-n)}$ ($0 \le fd\_cnt \le 2^{(15-n)}$)

Bit	Name	Type	Description
[15:8]	-	-	Reserved
[7:4]	yshift	R/W	Displacement in the vertical direction between adjacent windows 0: 1pixel 1: 2pixels … n: (n+1)pixels
[3:0]	xshift	R/W	Displacement in the horizontal direction between adjacent windows 0: 1pixel 1: 2pixels … n: (n+1)pixels

11.2.2.5 Read Channel 0 Start Address Register (Offset = 0x10)

Table 11-7. Read Channel 0 Start Address Register (Offset = 0x10)

Bit	Name	Type	Description
[31:3]	reg_raddr0	R/W	Read channel 0 start address in unit of 8bytes
[2:0]	-	-	Reserved

11.2.2.6 Read Channel 0 Address Offset Register (Offset = 0x14)

Table 11-8. Read Channel 0 Address Offset Register (Offset = 0x14)

Bit	Name	Type	Description
[31:17]	-	-	Reserved
[16:3]	reg_raoft0	R/W	Read channel 0 address offset in unit of 8bytes This value equals to B * ((image pitch) - (source width)). B: Bytes per pixel
[2:0]	-	-	Reserved

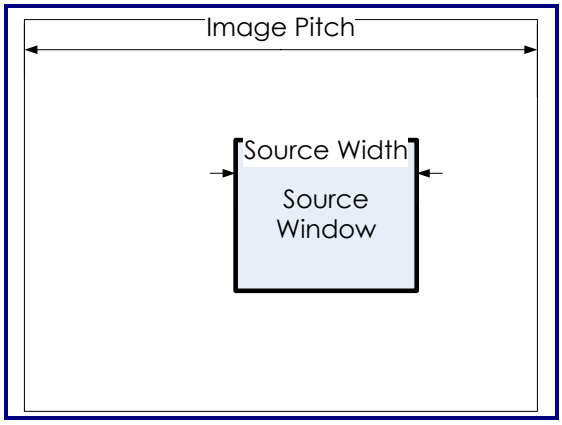


Figure 11-5 Source Window

11.2.2.7 Read Channel 1 Start Address Register (Offset = 0x18)

Table 11-9. Read Channel 1 Start Address Register (Offset = 0x18)

Bit	Name	Type	Description
[31:3]	reg_raddr1	R/W	Read channel 1 start address in unit of 8bytes This channel is initiated by setting rch1_en = '1'.
[2:0]	-	-	Reserved

11.2.2.8 Read Channel 1 Address Offset Register (Offset = 0x1C)

Table 11-10. Read Channel 1 Address Offset Register (Offset = 0x1C)

Bit	Name	Type	Description
[31:17]	-	-	Reserved
[16:3]	reg_raoft1	R/W	Read channel 1 address offset in unit of 8bytes This value equals to B * ((image pitch) - (source width)). B: Bytes per pixel
[2:0]	-	-	Reserved

11.2.2.9 Write Channel 0 Start Address Register (Offset = 0x20)

Table 11-11. Write Channel 0 Start Address Register (Offset = 0x20)

Bit	Name	Type	Description
[31:3]	reg_waddr0	R/W	Write channel 0 start address in unit of 8bytes This channel is initiated by setting y_en or sad_en to '1'.
2	-	-	Reserved
[1:0]	reg_wainc0	R/W	Write channel 0 address increment 0: 1byte per pixel 1: 2bytes per pixel 2: 4bytes per pixel 3: 8bytes per pixel

11.2.2.10 Write Channel 1 Start Address Register (Offset = 0x24)

Table 11-12. Write Channel 1 Start Address Register (Offset = 0x24)

Bit	Name	Type	Description
[31:3]	reg_waddr1	R/W	Write channel 1 start address in unit of 8bytes. This channel is initiated by setting cbcr_en or lgc_en to '1'.
2	-	-	Reserved
[1:0]	reg_wainc1	R/W	Write channel 1 address increment 0: 1byte per pixel 1: 2bytes per pixel 2: 4bytes per pixel 3: 8bytes per pixel

11.2.2.11 Write Channel 2 Start Address Register (Offset = 0x28)

Table 11-13. Write Channel 2 Start Address Register (Offset = 0x28)

Bit	Name	Type	Description
[31:3]	reg_waddr2	R/W	Write channel 2 start address in unit of 8bytes This channel is initiated by setting h_en to '1'.
2	-	-	Reserved

Bit	Name	Type	Description
[1:0]	reg_wainc2	R/W	Write channel 2 address increment 0: 1byte per pixel 1: 2bytes per pixel 2: 4bytes per pixel 3: 8bytes per pixel

11.2.2.12 Write Channel 3 Start Address Register (Offset = 0x2C)

Table 11-14. Write Channel 3 Start Address Register (Offset = 0x2C)

Bit	Name	Type	Description
[31:3]	reg_waddr3	R/W	Write channel 3 start address in unit of 8bytes This channel is initiated by setting s_en to '1'.
2	-	-	-
[1:0]	reg_wainc3	R/W	Write channel 3 address increment 0: 1byte per pixel 1: 2bytes per pixel 2: 4bytes per pixel 3: 8bytes per pixel

11.2.2.13 Write Channel 4 Start Address Register (Offset = 0x30)

Table 11-15. Write Channel 4 Start Address Register (Offset = 0x30)

Bit	Name	Type	Description
[31:3]	reg_waddr4	R/W	Write channel 4 start address in unit of 8bytes This channel is initiated by setting i_en to '1'.
2	-	-	-
[1:0]	reg_wainc4	R/W	Write channel 4 address increment 0: 1byte per pixel 1: 2bytes per pixel 2: 4bytes per pixel 3: 8bytes per pixel

11.2.2.14 Write Channel 5 Start Address Register (Offset = 0x34)

Table 11-16. Write Channel 5 Start Address Register (Offset = 0x34)

Bit	Name	Type	Description
[31:3]	reg_waddr5	R/W	Write channel 5 start address in unit of 8bytes This channel is initiated by setting ii_en or cc_en to '1'.
2	-	-	-
[1:0]	reg_wainc5	R/W	Write channel 5 address increment 0: 1byte per pixel 1: 2bytes per pixel 2: 4bytes per pixel 3: 8bytes per pixel

11.2.2.15 Write Channel 6 Start Address Register (Offset = 0x38)

Table 11-17. Write Channel 6 Start Address Register (Offset = 0x38)

Bit	Name	Type	Description
[31:3]	reg_waddr6	R/W	Write channel 6 start address in unit of 8bytes This channel is initiated by setting si_en to 1.
2	-	-	Reserved
[1:0]	reg_wainc6	R/W	Write channel 6 address increment 0: 1byte per pixel 1: 2bytes per pixel 2: 4bytes per pixel 3: 8bytes per pixel

11.2.2.16 Write Channel 7 Start Address Register (Offset = 0x3C)

Table 11-18. Write Channel 7 Start Address Register (Offset = 0x3C)

Bit	Name	Type	Description
[31:0]	-	-	-

11.2.2.17 Write Channel Address Offset Register (Offset = 0x40)

Table 11-19. Write Channel Address Offset Register (Offset = 0x40)

Bit	Name	Type	Description
[31:17]	-	-	Reserved
[16:3]	reg_waoft	R/W	Write channel address offset in unit of 8pixels This value equals to (image pitch) - (destination width).
[2:0]	-	-	Reserved

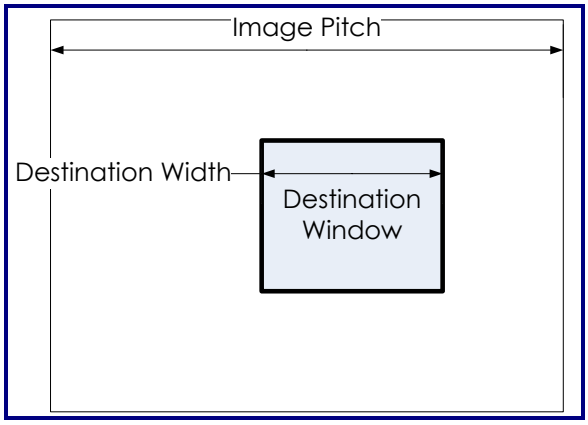


Figure 11-6 Destination Window

11.2.2.18 Write Channel Status Register (Offset = 0x60)

Table 11-20. Write Channel 2 Start Address Register (Offset = 0x60)

Bit	Name	Type	Description
[31:27]	-	-	Reserved
[26:24]	fifo6_cnt	RO	Number of entities in FIFO of write channel 6
23	-	-	Reserved
[22:20]	fifo5_cnt	RO	Number of entities in FIFO of write channel 5
19	-	-	Reserved
[18:16]	fifo4_cnt	RO	Number of entities in FIFO of write channel 4
15	-	-	Reserved
[14:12]	fifo3_cnt	RO	Number of entities in FIFO of write channel 3
11	-	-	Reserved

Bit	Name	Type	Description
[10:8]	fifo2_cnt	RO	Number of entities in FIFO of write channel 2
7	-	-	Reserved
[6:4]:	fifo1_cnt	RO	Number of entities in FIFO of write channel 1
3	-	-	Reserved
[2:0]	fifo0_cnt	RO	Number of entities in FIFO of write channel 0

11.2.2.19 Read Channel 0 Status Register (Offset = 0x68)

Table 11-21. Read Channel 0 Status Register (Offset = 0x68)

Bit	Name	Type	Description
[31:16]	ypos	RO	y position of read channel 0
[15:3]	xpos	RO	x position of read channel 0
2	-	-	Reserved
[1:0]	rch_cs	RO	Current state of read channel 0

11.2.2.20 Read Channel 1 Status Register (Offset = 0x6C)

Table 11-22. Read Channel 1 Status Register (Offset = 0x6C)

Bit	Name	Type	Description
[31:16]	ypos	RO	y position of read channel 1
[15:3]	xpos	RO	x position of read channel 1
2	-	-	Reserved
[1:0]	rch_cs	RO	current state of read channel 1

11.2.2.21 Face Detection Status Register (Offset = 0x70)

Table 11-23. Face Detection Status Register r (Offset = 0x70)

Bit	Name	Type	Description
[31:16]	-	-	Reserved
[15:0]	fd_cnt	RO	Number of detected objects

11.2.2.22 IVS Status Register (Offset = 0x74)

Table 11-24. Face Detection Status Register (Offset = 0x74)

Bit	Name	Type	Description
[31:1]	-	-	Reserved
0	ii_done	RO	Status of the IVS engine 0: Busy 1: Idle

11.2.2.23 Interrupt Status Register (Offset = 0x78)

Table 11-25. Interrupt Status Register (Offset = 0x78)

Bit	Name	Type	Description
[31:1]	-	-	Reserved
0	ii_ints	R/W1C	Interrupt status of the IVS engine This bit is set when a job is completed. 0: Interrupt does not occur. 1: Interrupt occurs. Write '1' to this bit clears the interrupt.

11.2.2.24 Filter Mask 0 of Kernel Template n (Offset = 0x80 + 0x20 * n)

Table 11-26. Filter Mask 0 of Kernel Template n (Offset = 0x80 + 0x20 * n)

Bit	Name	Type	Description
[31:24]	mask30	R/W	8bit signed coefficient of mask(3, 0)
[23:16]	mask20	R/W	8bit signed coefficient of mask(2, 0)
[15:8]	mask10	R/W	8bit signed coefficient of mask(1, 0)
[7:0]	mask00	R/W	8bit signed coefficient of mask(0, 0)

11.2.2.25 Filter Mask 1 of Kernel Template n (Offset = 0x80 + 0x20 * n + 0x04)

Table 11-27. Filter Mask 1 of Kernel Template n (Offset = 0x80 + 0x20 * n + 0x04)

Bit	Name	Type	Description
[31:24]	mask21	R/W	8bit signed coefficient of mask(2, 1)
[23:16]	mask11	R/W	8bit signed coefficient of mask(1, 1)
[15:8]	mask01	R/W	8bit signed coefficient of mask(0, 1)
[7:0]	mask40	R/W	8bit signed coefficient of mask(4, 0)

11.2.2.26 Filter Mask 2 of Kernel Template n (Offset = 0x80 + 0x20 * n + 0x08)

Table 11-28. Filter Mask 2 of Kernel Template n (Offset = 0x80 + 0x20 * n + 0x08)

Bit	Name	Type	Description
[31:24]	mask12	R/W	8bit signed coefficient of mask(1, 2)
[23:16]	mask02	R/W	8bit signed coefficient of mask(0, 2)
[15:8]	mask41	R/W	8bit signed coefficient of mask(4, 1)
[7:0]	mask31	R/W	8bit signed coefficient of mask(3, 1)

11.2.2.27 Filter Mask 3 of Kernel Template n (Offset = 0x80 + 0x20 * n + 0x0C)

Table 11-29. Filter Mask 3 of Kernel Template n (Offset = 0x80 + 0x20 * n + 0x0C)

Bit	Name	Type	Description
[31:24]	mask03	R/W	8bit signed coefficient of mask(0, 3)
[23:16]	mask42	R/W	8bit signed coefficient of mask(4, 2)
[15:8]	mask32	R/W	8bit signed coefficient of mask(3, 2)
[7:0]	mask22	R/W	8bit signed coefficient of mask(2, 2)

11.2.2.28 Filter Mask 4 of Kernel Template n (Offset = 0x80 + 0x20 * n + 0x10)

Table 11-30. Filter Mask 4 of Kernel Template n (Offset = 0x80 + 0x20 * n + 0x10)

Bit	Name	Type	Description
[31:24]	mask43	R/W	8bit signed coefficient of mask(4, 3)
[23:16]	mask33	R/W	8bit signed coefficient of mask(3, 3)
[15:8]	mask23	R/W	8bit signed coefficient of mask(2, 3)
[7:0]	mask13	R/W	8bit signed coefficient of mask(1, 3)

11.2.2.29 Filter Mask 5 of Kernel Template n (Offset = 0x80 + 0x20 * n + 0x14)

Table 11-31. Filter Mask 5 of Kernel Template n (Offset = 0x80 + 0x20 * n + 0x14)

Bit	Name	Type	Description
[31:24]	mask34	R/W	8bit signed coefficient of mask(3, 4)
[23:16]	mask24	R/W	8bit signed coefficient of mask(2, 4)
[15:8]	mask14	R/W	8bit signed coefficient of mask(1, 4)
[7:0]	mask04	R/W	8bit signed coefficient of mask(0, 4)

11.2.2.30 Filter Mask 6 of Kernel Template n (Offset = 0x80 + 0x20 * n + 0x18)

Table 11-32. Filter Mask 6 of Kernel Template n (Offset = 0x80 + 0x20 * n + 0x18)

Bit	Name	Type	Description
[31:16]	-	-	Reserved
[15:8]	shift	R/W	Number of shift amount of the convolution sum This value should be in the range of 0 to 21, inclusive.
[7:0]	mask44	R/W	8bit signed coefficient of mask(4, 4)

11.2.2.31 Stage Threshold of Haar-Like Features (Offset = 0x0180 ~ 0x037F)

Table 11-33. Stage Threshold of Haar-Like Features (Offset = 0x0180 ~ 0x037F)

Bit	Name	Type	Description
[31:16]	ts1	R/W	Stage threshold of stage 2*n + 1, 0 ≤ n ≤ 127
[15:0]	ts0	R/W	Stage threshold of stage 2*n, 0 ≤ n ≤ 127

11.2.2.32 Weak Value of Haar-Like Features (Offset = 0x2000 ~ 0x3FFF)

Table 11-34. Weak Value of Haar-Like Features (Offset = 0x2000 ~ 0x3FFF)

Bit	Name	Type	Description
[31:24]	-	-	Reserved
[23:16]	a2	R/W	Weak value 2 of weak classifier n, 0 ≤ n ≤ 2047
[15:8]	a1	R/W	Weak value 1 of weak classifier n, 0 ≤ n ≤ 2047
[7:0]	a0	R/W	Weak value 0 of weak classifier n, 0 ≤ n ≤ 2047

11.2.2.33 Feature threshold of Haar-Like Features (Offset = 0x4000 ~ 0x7FFF)

Table 11-35. Feature Threshold of Haar-Like Features (Offset = 0x4000 ~ 0x7FFF)

Bit	Name	Type	Description
[31:27]	-	-	Reserved
26	slast	WO	This bit indicates the last stage of the cascade classifier. 0: This feature is not at the last stage. 1: This feature is at the last stage.
25	clast	WO	This bit indicates the last weak classifier at any stage. 0: More weak classifiers accumulated at this stage. 1: This stage ends after this weak classifier.
24	skin	WO	This bit selects the decision rule for hypothesis testing. The detailed diagram is shown in Figure 11-14.
[23:21]	-	-	Reserved
[20:18]	type	WO	Types of features The supported types of features are depicted in Figure 11-7.
[17:0]	tf	WO	Feature threshold

11.2.2.34 Unit Rectangle Information of Haar-Like Features (Offset = 0x8000 ~ 0xBFFF)

Table 11-36. Unit Rectangle Information of Haar-Like Features (Offset = 0x8000 ~ 0xBFFF)

Bit	Name	Type	Description
[31:29]	-	-	Reserved
[28:24]	dy	WO	Height of the unit rectangle This value should be in a range from 1 to 20, inclusive. The unit rectangle is depicted in Figure 11-7.
[23:21]	-	-	Reserved
[20:16]	dx	WO	Width of the unit rectangle This value should be in a range from 1 to 20, inclusive. The unit rectangle is depicted in Figure 11-7.
[15:13]	-	-	Reserved
[12:8]	y	WO	Top-most corner of the unit rectangle This value should be in a range from 0 to 19, inclusive. The unit rectangle is depicted in Figure 11-7.
[7:5]	-	-	Reserved
[4:0]	x	WO	Left-most comer of the unit rectangle This value should be in a range from 0 to 19, inclusive. The unit rectangle is depicted in Figure 11-7.

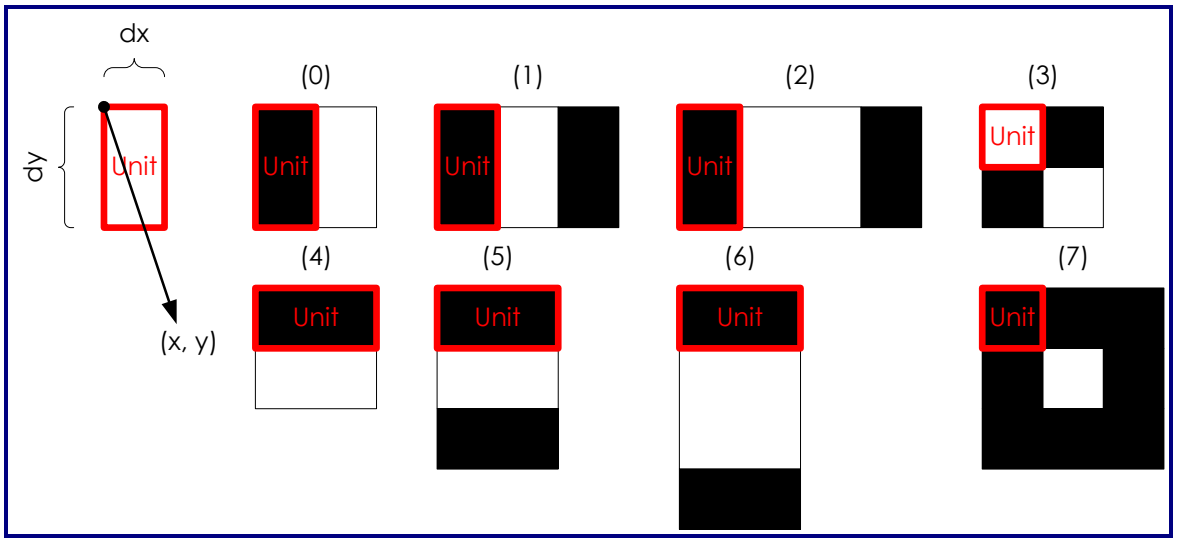


Figure 11-7. Types of Supported Features

Table 11-37. Restrictions on Unit Rectangle

Feature Type	Constraint on (x, dx)	Constraint on (y, dy)
0	x + 2 * dx ≤ 20	y + dy ≤ 20
1	x + 3 * dx ≤ 20	y + dy ≤ 20
2	x + 4 * dx ≤ 20	y + dy ≤ 20
3	x + 2 * dx ≤ 20	y + 2 * dy ≤ 20
4	x + dx ≤ 20	y + 2 * dy ≤ 20
5	x + dx ≤ 20	y + 3 * dy ≤ 20
6	x + dx ≤ 20	y + 4 * dy ≤ 20
7	x + 3 * dx ≤ 20	y + 3 * dy ≤ 20

11.3 Function Description

11.3.1 Source Data Format

The source image is stored in the raster scan, in the packed or planar data format. A list of the source data format is provided in Table 11-38. Figure 11-8 and Figure 11-9 depict the pixel orders on a little-endian machine.

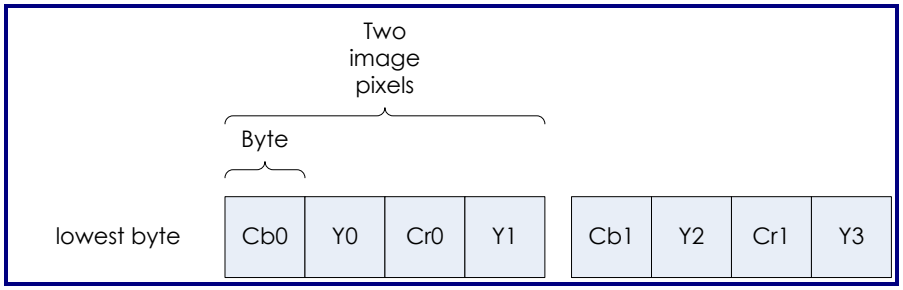


Figure 11-8. Pixel Order in Packed Data Format

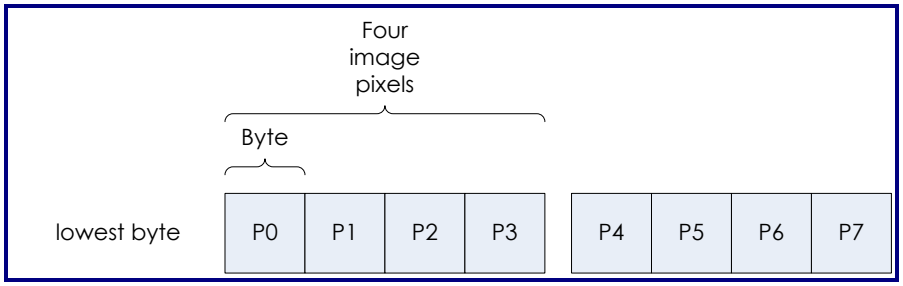


Figure 11-9. Pixel Order in Planar Data Format

Table 11-38. Source Data Format

Operation	Source Data Format
De-interleaving YC	Packed YCbCr 4:2:2
HSI conversion	Packed YCbCr 4:2:2
Integral image	Packed YCbCr 4:2:2/Planar
SAD	Planar
Histogram	Packed YCbCr 4:2:2/Planar
Convolution	Planar
Morphology	Planar
Raster operation	Planar

11.3.2 Destination Data Format

The IVS engine stores most destination images in the planar data format and little-endian byte order, except for the de-interleaved chrominance component and the outcome of the cascade classifier, which are multiplexed in the semi-planar format. Table 11-39 lists the pixel unit of the resultant images.

Table 11-39. Destination Pixel Unit

Operation	Bits per Pixel
De-interleaving YC	8 (Y)
	16 (C)
HSI conversion	16 (H)
	8 (S)
	8 (I)
RGB conversion	8 (R)
	8 (G)
	8 (B)
Integral image	32
	32 (Squared)
SAD	16
Histogram	32
Convolution	8
	16 (Gx)
	16 (Gy)
	16 (Magnitude)

Operation	Bits per Pixel
Morphology	1 or 8
Raster operation	1 or 8

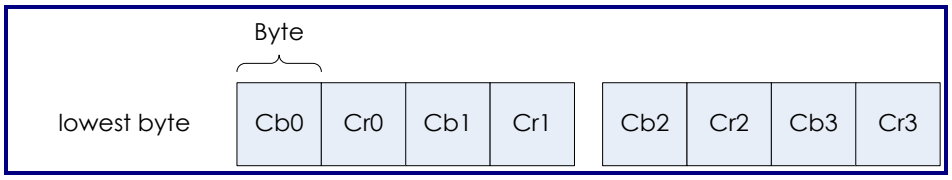


Figure 11-10. Semi-Planar Format of De-interleaved Chrominance

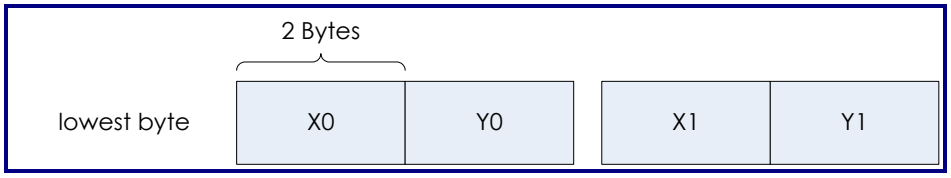


Figure 11-11. Semi-Planar Format of Cascade Classifier

11.3.3 HSI Conversion

Pixels in the YCbCr domain are first converted to the gamma-corrected RGB (R'G'B') data by using the following equations (BT.601).

$$R' = 1.164(Y - 16) + 1.596(Cr - 128)$$

$$G' = 1.164(Y - 16) - 0.813(Cr - 128) - 0.392(Cb - 128)$$

$$B' = 1.164(Y - 16) + 2.017(Cb - 128)$$

BT.601 defines Y to have a nominal range of 16 ~ 235 (Black-White); Cb and Cr are defined to have a nominal range of 16 ~ 240, with 128 corresponding to zero. The resulting RGB values have a nominal range of 0 ~ 255 (Black-White). The H, S, and I components are then obtained by,

$$H = \begin{cases} \theta & \text{if } B \le G \\ 2\pi - \theta & \text{if } B > G \end{cases}$$

$$\theta = \cos^{-1}\left(\frac{(R-G)+(R-B)}{2\sqrt{(R-G)^2+(R-B)(G-B)}} \right)$$

$$S = 128(1 - 3 \times \min(R,G,B)/(R + G + B))$$

$$I = (R + G + B)/3$$

The H, S, and I values are in ranges of [0,360], [0, 128], and [0, 255], respectively.

11.3.4 Integral Image

Each pixel of the integral image is the sum of all the pixels inside the rectangle bounded by the upper left corner of the image and the pixel of interest. For an intensity image i(x, y), its integral image denoted by ii(x, y) is defined as,

$$ii(x, y) = \Sigma\, i(x', y')$$
$$x' \le x,\ y' \le y$$

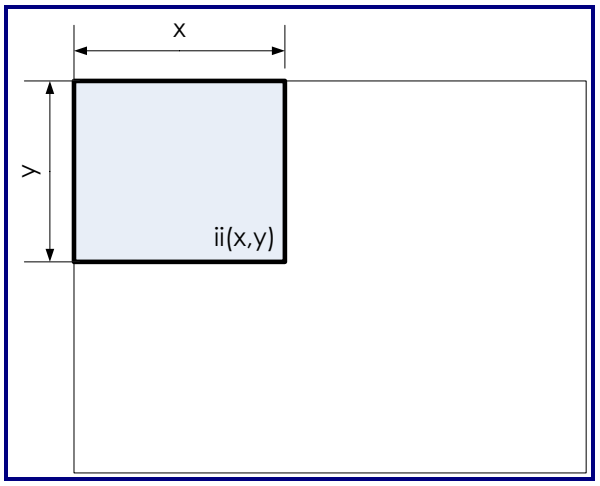


Figure 11-12. Example of Integral Image

11.3.5 Sum of Absolute Difference (SAD)

SAD adds up the absolute differences between the corresponding elements in a designated region of two video frames. Its value is given by,

$$SAD(x,y) = \sum_{j=-n}^{n} \sum_{i=-n}^{n} \left| A(x+i, y+j) - B(x+i, y+j) \right|$$

where n ϵ {0, 1, 2}.

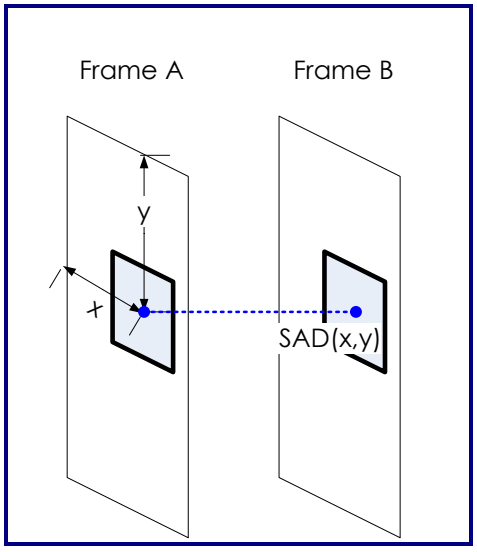


Figure 11-13. Example of SAD

11.3.6 Raster Operation (ROP)

Raster operation performs the Boolean operations on operands from two source images according to the 4bit ROP value specified by users. The truth table of ROP is given in Table 11-40.

Table 11-40. Truth Table of Raster Operation

Operand from Read Channel 1	Operand from Read Channel 0	ROP Value
0	0	Bit0
0	1	Bit1
1	0	Bit2
1	1	Bit3

An example shows on how to use the ROP value are listed in Table 11-41.

Table 11-41. Example of Raster Operation Code

Destination Data	ROP Value
Read Channel 0	0b1010
Read Channel 1	0b1100
Read Channel 0 AND Read Channel 1	0b1000
Read Channel 0 OR Read Channel 1	0b1110

11.3.7 Histogram

An image histogram is a count of the intensity levels in an image. This function processes either image data in planar form or the luminance component from an interleaved 4:2:2 video stream, and generates an image histogram consisting of 256 bins corresponding to the 256 possible pixel intensities. Each bin contains a count of the number of pixels in the image that have that particular intensity value. The maximum number of pixels that can be profiled in each bin is 2^{32} - 1 in the main histogram.

11.3.8 Convolution

The convolution function is used to apply generic filters to an input image. It performs a point by point multiplication of 5 by 5 masks with the input image. The masks are provided as 8bit signed values. The results of 25 multiplications are then summed to produce a 21bit convolution sum. If user specifies the shift value (not equal to zero), the convolution sum is shifted down to the byte range and saturated to the range of 0 and 255, inclusive. If the shift value is defined to be zero, the convolution sum is saturated to the range of -2^{15} and 2^{15} - 1, inclusive. The result being stored is given by,

$$sum = \left(\sum_{j=-2}^{2} \sum_{i=-2}^{2} i(x+i, y+j) \times mask(i+2, j+2) \right) >> shift$$

where i(x, y) is the input image. For applications of the gradient filter, this function generates the gradient vector ($\nabla \mathbf{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix}$) by,

$$G_x = \sum_{j=-2}^{2}\sum_{i=-2}^{2} i(x+i, y+j) \times mask(i+2, j+2)$$

$$G_y = \sum_{j=-2}^{2}\sum_{i=-2}^{2} i(x+i, y+j) \times mask(j+2, i+2)$$

, as well as the magnitude of this vector, which is given by $\nabla f = |G_x| + |G_y|$.

11.3.9 Morphology

The morphological operators are used to perform the dilation, erosion, opening, and closing operations on the binary images. The structuring elements of 3x3 and 5x5 are supported for erosion and dilation. The structuring element of 3x3 is supported for opening and closing. The structuring elements are provided by programming the mask(i, j) to the binary values (0s or 1s). A non-zero threshold should be provided along with the gray-scale images such that this function will perform binarization, in which the pixels value of greater than or equal to the threshold are binarized to 1s while others to 0s, prior to the morphological operation. The morphological operation is bypassed by setting the operator to NOP (7), which results in only simple thresholding with the gray-scale images.

11.3.10 Cascade Classifier

Object detection procedure classifies images based on the values of simple features. This function reads the results from the integral image and then propagates the computed feature values into multiple weak classifiers trained using AdaBoost. The pixel data within a window of 20x20 can be arbitrarily selected to form a rectangle feature. If any classifier in the cascade rejects a hypothesis, the detection window will be considered as a negative example. Otherwise, its top-left X-Y coordinates will be stored in 2bytes, respectively. The detection window is moved one column at a time, advancing the detection window over the entire image until the entire width is covered.

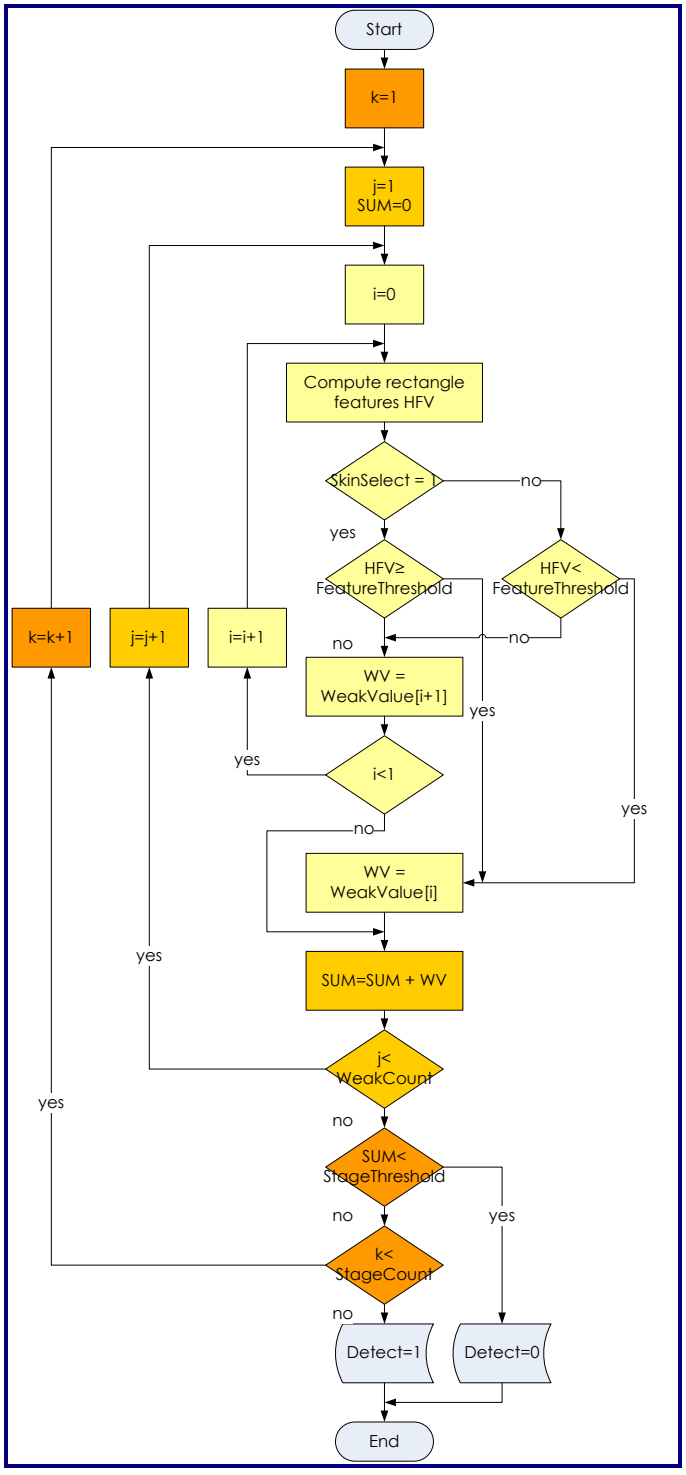


Figure 11-14. Flow Chart of Cascade Classifier

Chapter 12

High-Speed I/O

This chapter contains the following sections:

12.1 High-Speed I/O Interfaces

The high-speed I/O interfaces of GM8136S/GM8135S have the RMII Ethernet controllers, SD 3.0 host controller, USB 2.0 OTG, and OTG 1.1 device mode.

12.2 RMII Ethernet Controller (RMII MAC)

12.2.1 General Description

RMII MAC is a high-quality Ethernet controller with the DMA function. It includes the AHB wrapper, DMA engine, on-chip memories (TX FIFO and RX FIFO), MAC, and RMII interfaces.

RMII MAC is an Ethernet controller that provides the AHB master capability and is fully compliance with the IEEE 802.3 specification for the 10/100Mbps Ethernet. The RMII MAC Ethernet controller with the DMA function handles all data transfers between the system memory and the on-chip memories. With the DMA engine, this controller can reduce the CPU loading, maximize the performance, and minimize the FIFO size. RMII MAC has the on-chip memories for buffering so that the external local buffer memory will not be needed. The RMII interfaces support two specific data rates, 10Mbps and 100Mbps.

To reduce the processing load of the host CPU, RMII MAC implements the TCP, UDP, and IP V4 checksum generations and validations to support the VLAN tagging. For the QoS and CoS requirements, RMII MAC supports the high-priority queues to reduce the processing load of the host CPU for transmitting packets.

12.2.2 Features

- Supports DMA engine for transmitting and receiving packets
- Programmable DMA burst size
- Supports transmit and receive interrupt mitigations
- Supports zero-copy data transfer
- Supports IP, TCP, and UDP checksum generations
- Supports high-priority transmit priority queues for QoS and CoS applications
- Independent TX/RX FIFOs
- Supports half duplex and full duplex

- Supports flow control for full duplex and backpressure for half duplex
- Supports RMII interfaces
- Supports Jumbo packets (9Kbytes)

12.2.3 Register Descriptions

The following abbreviations are used for register definitions:

- R/W: Read/Write
- RC: Read Clear
- RO: Read Only
- R/W1C: Read/Write 1 Clear

The RMII MAC controller registers are shown in Table 12-1.

Table 12-1. Summary of RMII MAC Controller Registers

Offset	Type	Description	Reset Value
0x00 ~ 0x03	R/W1C	Interrupt Status Register, ISR	0x0
0x04 ~ 0x07	R/W	Interrupt Enable Register, IME	0x0
0x08 ~ 0x0B	R/W	MAC Most Significant Address Register, MAC_MADR	0x0
0x0C ~ 0x0F	R/W	MAC Least Significant Address Register, MAC_LADR	0x0
0x10 ~ 0x13	R/W	Multicast Address Hash Table 0 Register, MAHT0	0x0
0x14 ~ 0x17	R/W	Multicast Address Hash Table 1 Register, MAHT1	0x0
0x18 ~ 0x1B	WO	Normal Priority Transmit Poll Demand Register, NPTXPD	0x0
0x1C ~ 0x1F	WO	Receive Poll Demand Register, RXPD	0x0
0x20 ~ 0x23	R/W	Normal Priority Transmit Ring Base Address Register, NPTXR_BADR	0x0
0x24 ~ 0x27	R/W	Receive Ring Base Address Register, RXR_BADR	0x0
0x28 ~ 0x2B	WO	High Priority Transmit Poll Demand Register, HPTXPD	0x0
0x2C ~ 0x2F	R/W	High Priority Transmit Ring Base Address Register, HPTXR_BADR	0x0
0x30 ~ 0x33	R/W	Interrupt Timer Control Register, ITC	0x0
0x34 ~ 0x37	R/W	Automatic Polling Timer Control Register, APTC	0x0
0x38 ~ 0x3B	R/W	DMA Burst Length and Arbitration Control Register, DBLAC	0x0
0x3C ~ 0x3F	RO	DMA/FIFO State Register, DMAFIFOS	0x0
0x40 ~ 0x43	RO	Revision Register, REVR	0x0
0x44 ~ 0x47	RO	Feature Register, FEAR	0x0
0x48 ~ 0x4B	R/W	Transmit Priority Arbitration and FIFO Control Register, TPAFCR	0x0
0x4C ~ 0x4F	R/W	Receive Buffer Size Register, RBSR	0x640
0x50 ~ 0x53	R/W	MAC Control Register, MACCR	0x0
0x54 ~ 0x57	R/W1C	MAC Status Register, MACSR	0x0
0x58 ~ 0x5C	R/W	Test Mode Register, TM	0x0
0x60 ~ 0x63	R/W	PHY Control Register, PHYCR	0x0
0x64 ~ 0x67	-	PHY Data Register, PHYDATA	0x0
0x68 ~ 0x6B	-	Flow Control Register, FCR	0x0000A400
0x6C ~ 0x6F	R/W	Back Pressure Register, BPR	0x00000400
0x90 ~ 0x93	RO	Normal Priority Transmit Ring Pointer Register, NPTXR_PTR	0x0
0x94 ~ 0x97	RO	High Priority Transmit Ring Pointer Register, HPTXR_PTR	0x0
0x98 ~ 0x9B	RO	Receive Ring Pointer Register, RXR_PTR	0x0

Offset	Type	Description	Reset Value
0xA0 ~ 0xA3	RO	TPKT_CNT Counter Register	0x0
0xA4 ~ 0xA7	RO	TXMCOL_CNT and TXSCOL_CNT Counter Register	0x0
0xA8 ~ 0xAB	RO	TXECOL_CNT and TXFAIL_CNT Counter Register	0x0
0xAC ~ 0xAF	RO	TXLCOL_CNT and TXUNDERUN_CNT Counter Register	0x0
0xB0 ~ 0xB3	RO	RPKT_CNT Counter Register	0x0
0xB4 ~ 0xB7	RO	BROPKT_CNT Counter Register	0x0
0xB8 ~ 0xBB	RO	MULPKT_CNT Counter Register	0x0
0xBC ~ 0xBF	RO	RPF_CNT and AEP_CNT Counter Register	0x0
0xC0 ~ 0xC3	RO	RUNT_CNT Counter Register	0x0
0xC4 ~ 0xC7	RO	CRCER_CNT and FTL_CNT Counter Register	0x0
0xC8 ~ 0xCB	RO	RCOL_CNT and RLOST_CNT Counter Register	0x0

12.2.3.1 Interrupt Status Register (ISR, Offset = 0x0)

Table 12-2. Interrupt Status Register (ISR, Offset = 0x0)

Bit	RW Type	Default Value	Name	Description
[31:11]	-	-	-	Reserved
10	R/W1C	0	HPTXBUF_UNAVA	High-priority transmit buffer is unavailable.
9	R/W1C	0	PHYSTS_CHG	PHY link status change
8	R/W1C	0	AHB_ERR	AHB bus error
7	R/W1C	0	TPKT_LOST	Packets transmitted to Ethernet are lost due to late collision, excessive collision, or underrun.
6	R/W1C	0	NPTXBUF_UNAVA	Normal priority transmit buffer is unavailable.
5	R/W1C	0	TPKT2F	TXDMA has moved data to TX FIFO.
4	R/W1C	0	TPKT2E	Packets are successfully transmitted to Ethernet.
3	R/W1C	0	RPKT_LOST	Received packet are lost due to RX FIFO full.
2	R/W1C	0	RXBUF_UNAVA	Receive buffer is unavailable.
1	R/W1C	0	RPKT2F	Packets are successfully received by RX FIFO.
0	R/W1C	0	RPKT2B	RXDMA has successfully received packets to the RX buffer.

12.2.3.2 Interrupt Enable Register (IME, Offset = 0x4)

Table 12-3. Interrupt Enable Register (IME, Offset = 0x4)

Bit	RW Type	Default Value	Name	Description
[31:11]	-	-	-	Reserved
10	R/W	0	HPTXBUF_UNAVA_EN	Interrupt enable of ISR [10]
9	R/W	0	PHYSTS_CHG_EN	Interrupt enable of ISR [9]
8	R/W	0	AHB_ERR_EN	Interrupt enable of ISR [8]
7	R/W	0	TPKT_LOST_EN	Interrupt enable of ISR [7]
6	R/W	0	NPTXBUF_UNAVA_EN	Interrupt enable of ISR [6]
5	R/W	0	TPKT2F_EN	Interrupt enable of ISR [5]
4	R/W	0	TPKT2E_EN	Interrupt enable of ISR [4]
3	R/W	0	RPKT_LOST_EN	Interrupt enable of ISR [3]
2	R/W	0	RXBUF_UNAVA_EN	Interrupt enable of ISR [2]
1	R/W	0	RPKT2F_EN	Interrupt enable of ISR [1]
0	R/W	0	RPKT2B_EN	Interrupt enable of ISR [0]

12.2.3.3 MAC Most Significant Address Register (MAC_MADR, Offset = 0x8)

Table 12-4. MAC Most Significant Address Register (MAC_MADR, Offset = 0x8)

Bit	RW Type	Default Value	Name	Description
[31:16]	-	-	-	Reserved
[15:0]	R/W	0	MAC_MADR	Most significant two bytes of the MAC address

12.2.3.4 MAC Least Significant Address Register (MAC_LADR, Offset = 0xC)

Table 12-5. MAC Least Significant Address Register (MAC_LADR, Offset = 0xC)

Bit	RW Type	Default Value	Name	Description
[31:0]	R/W	0	MAC_LADR	Least significant four bytes of the MAC address

12.2.3.5 Multicast Address Hash Table 0 Register (MAHT0, Offset = 0x10)

Table 12-6. Multicast Address Hash Table 0 Register (MAHT0, Offset = 0x10)

Bit	RW Type	Default Value	Name	Description
[31:0]	R/W	0	MAHT0	Multicast address hash table bytes 3 ~ 0 (Hash table 31:0)

12.2.3.6 Multicast Address Hash Table 1 Register (MAHT1, Offset = 0x14)

Table 12-7. Multicast Address Hash Table 1 Register (MAHT1, Offset = 0x14)

Bit	RW Type	Default Value	Name	Description
[31:0]	R/W	0	MAHT1	Multicast address hash table bytes 3 ~ 0 (Hash table 63:32)

12.2.3.7 Normal Priority Transmit Poll Demand Register (NPTXPD, Offset = 18h ~ 1Bh)

Table 12-8. Normal Priority Transmit Poll Demand Register (NPTXPD, Offset = 18h ~ 1Bh)

Bit	RW Type	Default Value	Name	Description
[31:0]	WO	0	TXPD	When writing a value to this register, RMII MAC will read the normal-priority transmit descriptor and check the txdma_own bit. If txdma_own = '1', it will move the transmit buffer data to TX FIFO. The read value of the register is always 0.

12.2.3.8 Receive Poll Demand Register (RXPD, Offset = 0x1C)

Table 12-9. Receive Poll Demand Register (RXPD, Offset = 0x1C)

Bit	RW Type	Default Value	Name	Description
[31:0]	WO	0	RXPD	When writing a value to this register, RMII MAC will read the receive descriptor and check the rxdma_own bit. If rxdma_own = '1', it will move the receive packet data from RX FIFO to the receive buffer in the system memory. The read value of the register is always 0.

12.2.3.9 Normal Priority Transmit Ring Base Address Register (NPTXR_BADR, Offset = 0x20)

Table 12-10. Normal Priority Transmit Ring Base Address Register (NPTXR_BADR, Offset = 0x20)

Bit	RW Type	Default Value	Name	Description
[31:0]	R/W	0	TXR_BADR	Base address of the normal-priority transmit ring. The base address must be 16byte aligned. RMII MAC treats base address bits 3 ~ 0 as 0 when reading the descriptors if bits 3 ~ 0 are not zeroes.

12.2.3.10 Receive Ring Base Address Register (RXR_BADR, Offset = 0x24)

Table 12-11. Receive Ring Base Address Register (RXR_BADR, Offset = 0x24)

Bit	RW Type	Default Value	Name	Description
[31:0]	R/W	0	RXR_BADR	Base address of the receive ring. The base address must be 16byte aligned. RMII MAC treats base address bits 3 ~ 0 as 0 when reading the descriptors if bits 3 ~ 0 are not zeroes.

12.2.3.11 High Priority Transmit Poll Demand Register (HPTXPD, Offset = 0x28)

Table 12-12. High Priority Transmit Poll Demand Register (HPTXPD, Offset = 0x28)

Bit	RW Type	Default Value	Name	Description
[31:0]	WO	0	HPTXPD	When writing a value to the register, RMII MAC will read the high-priority transmit descriptor process and check the txdma_own bit. If txdma_own = '1', it will move the transmit buffer data into TX FIFO. The read value of the register is always 0.

12.2.3.12 High Priority Transmit Ring Base Address Register (HPTXR_BADR, Offset = 0x2C)

Table 12-13. High Priority Transmit Ring Base Address Register (HPTXR_BADR, Offset = 0x2C)

Bit	RW Type	Default Value	Name	Description
[31:0]	R/W	0	HPTXR_BADR	Base address of the high-priority transmit ring. The base address must be 16byte aligned. RMII MAC treats base address bits 3 ~ 0 as 0 when reading the descriptors if bits 3 ~ 0 are not zeroes.

12.2.3.13 Interrupt Timer Control Register (ITC, Offset = 0x30)

Table 12-14. Interrupt Timer Control Register (ITC, Offset = 0x30)

Bit	RW Type	Default Value	Name	Description
[31:30]	-	-	-	Reserved
[29:28]	R/W		RXINT_THR_UNIT	This field defines the unit of RXINT_THR. 00 → Unit is 1 packet. 01 → Unit is 4 packets. 10 → Unit is 16 packets. 11 → Unit is 64 packets.
[27:25]	-	-	-	Reserved
[24:16]	R/W	0	RXINT_CNT_H	RXINT_CNT higher bits This field and RXINT_CNT_L define the maximum waiting time to issue the receive interrupt after a packet has been received by RMII MAC. The time unit is 1 RX cycle time. When RXINT_CNT = '0', the function is disabled. If RXINT_THR = '0' and RXINT_CNT = '0', a receive interrupt will be issued when a packet is received by GMAC.
15	R/W	0	TXINT_TIME_SEL	This field defines the period of TX cycle time. When set, the TX cycle times are: 100Mbps mode → 81.92µs 10Mbps mode → 819.2µs When cleared, the TX cycle times are: 100Mbps mode → 5.12µs 10Mbps mode → 51.2µs
[14:12]	R/W	0	TXINT_THR	This field defines the maximum number of the transmit interrupts that are pending before an interrupt is generated. When TXINT_THR is not equal to 0, RMII MAC will issue a transmit interrupt if the transmit packet number transmitted by RMII MAC reaches TXINT_THR. When TXINT_THR = '0' and TXINT_CNT = '0', RMII MAC will issue a transmit interrupt or will not depend on TXIC in TXDES1.

Bit	RW Type	Default Value	Name	Description
[11:8]	R/W	0	TXINT_CNT	This field defines the maximum wait time to issue the transmit interrupt after a packet has been transmitted by RMII MAC. The time unit is 1 TX cycle time. When TXINT_CNT = '0', the function will be disabled. When TXINT_THR = '0' and TXINT_CNT = '0', RMII MAC will issue a transmit interrupt or will not depend on TXIC in TXDES1.
7	R/W	0	RXINT_TIME_SEL	This field defines the period of RX cycle time. When set, the RX cycle times are: 100Mbps mode → 81.92µs 10Mbps mode → 819.2µs When cleared, the RX cycle times are: 100Mbps mode → 5.12µs 10Mbps mode → 51.2µs
[6:4]	R/W	0	RXINT_THR	This field defines the maximum number of the receive interrupts that are pending before an interrupt is generated. When RXINT_THR is not equal to 0, RMII MAC will issue a receive interrupt if the receive packet number received by RMII MAC reaches RXINT_THR. If RXINT_THR = '0' and RXINT_CNT = '0', a receive interrupt will be issued when RMII MAC finishes receiving a receive packet.
[3:0]	R/W	0	RXINT_CNT_L	RXINT_CNT lower bits This field and RXINT_CNT_H define the maximum wait time to issue the receive interrupt after a packet has been received by RMII MAC. The time unit is 1 RX cycle time. When RXINT_CNT = '0', the function is disabled. If RXINT_THR = '0' and RXINT_CNT = '0', a receive interrupt will be issued when a packet is received by RMII MAC.

Recommended value = 0x0000_1010

The Interrupt Timer Control Register allows the software driver to reduce the number of transmit interrupts (ISR[4]) and receive interrupts (ISR[0]) by setting the register. This lowers the CPU utilization for handling a large number of interrupts.

The register defines two threshold values for the receive packet number and transmit packet number, and two associated timers. The threshold value defines the maximum number of receive or transmit interrupts that can be pending before an interrupt is generated. The timer defines the maximum wait time to issue the transmit/receive interrupt after a packet has been transmitted/received by RMII MAC. The threshold value and timer combination allow batching of several packets into a single interrupt with a limit for how long it will be pending. The combination prevents throughput from being impeded in heavy traffic, and the time limit prevents resources from being held for too long in the low traffic.

The mitigation mechanism is similar for both receive and transmit interrupts. There is a counter (TXPKT_CNT) in RMII MAC to count the packets transmitted by RMII MAC. When the counter reaches TXINT_THR and TXINT_THR is not equal to 0, RMII MAC will issue the transmit interrupt. There is also a counter (RXPKT_CNT) in RMII MAC to count the packets received by RMII MAC. When the counter reaches RXINT_THR and RXINT_THR is not equal to 0, RMII MAC will issue a receive interrupt. TXPKT_CNT will be cleared when the transmit interrupt is issued. RXPKT_CNT will be cleared when the receive interrupt is issued.

The following table shows the condition for RMII MAC to issue a transmit interrupt.

TXINT_THR = 0	TXINT_CNT = 0	RMII MAC Action
True	True	• The transmit interrupt will be issued after a packet is transmitted and TXIC of the packet is set. • Clear TXPKT_CNT
True	False	• The transmit interrupt will be issued after a packet is transmitted and timer reaches the value of TXINT_CNT. • Clear TXPKT_CNT
False	True	• The transmit interrupt will be issued if TXPKT_CNT = TXINT_THR. • Clear TXPKT_CNT
False	False	• The transmit interrupt will be issued if the following condition holds: ○ TXPKT_CNT = TXINT_THR ○ TXPKT_CNT = 1 and timer reaches the value of TXINT_CNT. • Clear TXPKT_CNT

The following table shows the condition for RMII MAC to issue a receive interrupt.

RXINT_THR = 0	{RXINT_CNT_H,RXINT_CNT_L} = 0	RMII MAC Action
True	True	• The receive interrupt will be issued after a packet is received by RMII MAC. • Clear RXPKT_CNT
True	False	• The receive interrupt will be issued after a packet is received by RMII MAC and timer reaches the value of RXINT_CNT. • Clear RXPKT_CNT
False	True	• The receive interrupt will be issued if RXPKT_CNT = RXINT_THR. • Clear TXPKT_CNT
False	False	• The receive interrupt will be issued if the following conditions hold: ○ RXPKT_CNT = RXINT_THR ○ RXPKT_CNT = '1' and timer reaches the value of RXINT_CNT • Clear RXPKT_CNT

12.2.3.14 Automatic Polling Timer Control Register (APTC, Offset = 0x34)

This register allows RMII MAC to automatically poll the descriptors. This will lower the CPU utilization. If the transmit automatic poll function is enabled, RMII MAC will automatically poll the transmit descriptor until the transmit automatic poll timer expires. If the function is disabled, software will write the Transmit Poll Demand register (Offset: 0x18) to trigger RMII MAC for reading the transmit descriptors after software has prepared the transmit packets in the transmit buffers.

If the receive automatic poll function is enabled, RMII MAC will automatically poll the receive descriptor until the receive automatic poll timer expires. If the function is disabled, software will write the Receive Poll Demand register (Offset: 0x1C) to trigger RMII MAC for reading the receive descriptors after software has released the receive descriptors to RMII MAC.

Table 12-15. Automatic Polling Timer Control Register (APTC, Offset = 0x34)

Bit	RW Type	Default Value	Name	Description
[31:13]	-	-	-	Reserved
12	R/W	0	TXPOLL_TIME_SEL	This bit defines the period of TX poll time. When set, the TX poll times are: 100Mbps mode → 81.92µs 10Mbps mode → 819.2µs When cleared, the TX poll times are: 100Mbps mode → 5.12µs 10Mbps mode → 51.2µs
[11:8]	R/W	0	TXPOLL_CNT	This field defines the period of the transmit automatic poll time. The unit is 1 TX poll time. When TXPOLL_CNT is not equal to 0, RMII MAC will automatically poll the transmit descriptor. If TXPOLL_CNT = '0', RMII MAC will not automatically poll the transmit descriptor.
[7:5]	-	-	-	Reserved
4	R/W	0	RXPOLL_TIME_SEL	This field defines the period of RX poll time. When set, the RX poll times are: 100Mbps mode → 81.92µs 10Mbps mode → 819.2µs When cleared, the RX poll times are: 100Mbps mode → 5.12µs 10Mbps mode → 51.2µs
[3:0]	R/W	0	RXPOLL_CNT	This field defines the period of the receive automatic poll time. The unit is 1 RX poll time. When RXPOLL_CNT is not equal to 0, RMII MAC will automatically poll the receive descriptor. If RXPOLL_CNT = '0', RMII MAC will not automatically poll the receive descriptor.

Recommended value = 0x0000_0001

12.2.3.15 DMA Burst Length and Arbitration Control Register (DBLAC, Offset = 0x38)

Table 12-16. DMA Burst Length and Arbitration Control Register (DBLAC, Offset = 0x38)

Bit	RW Type	Default Value	Name	Description
[31:24]	-	-	-	Reserved

Bit	RW Type	Default Value	Name	Description
23	R/W	0	IFG_INC	IFG (Inter-Frame Gap) increase The bit defines the increase or decrease of IFG in Ethernet. When IFG_INC = '1', IFG will increase. When IFG_INC = '0', IFG will decrease.
[22:20]	R/W	0	IFG_CNT	IFG (Inter-Frame Gap) count The field defines the increase or decrease number of IFG in Ethernet. When IFG_INC = '1', FG will increase. When IFG_INC = '0', IFG will decrease. The unit is 1 transmit clock in Ethernet (40ns in the 100Mbps mode, and 400ns in the 10Mbps mode).
[19:16]	R/W	2	TXDES_SIZE	Transmit descriptor size This field defines the transmit descriptor size. Writing 0 to this field is illegal. The unit is 8bytes.
[15:12]	R/W	2	RXDES_SIZE	Receive descriptor size This field defines the receive descriptor size. Writing 0 to this field is illegal. The unit is 8bytes.
[11:10]	R/W	3	TXBST_SIZE	TXDMA maximum burst size per TXDMA burst This field sets the maximum size of the TXDMA burst. The burst sizes are as follows: 00: 64bytes 01: 128bytes 10: 256bytes 11: 512bytes

Bit	RW Type	Default Value	Name	Description
[9:8]	R/W	3	RXBST_SIZE	RXDMA maximum burst size per RXDMA burst This field sets the maximum size of RXDMA burst. The burst sizes are as follows: 00: 64bytes 01: 128bytes 10: 256bytes 11: 512bytes
7	-	-	-	Reserved
6	R/W	0	RX_THR_EN	Enable the RX FIFO threshold arbitration
[5:3]	R/W	0	RXFIFO_HTHR	RX FIFO high threshold value for arbitration When the used space in RX FIFO is larger than or equal to the RX FIFO high threshold value, RXDMA will have higher priority over TXDMA by using the DMA channel. RXDMA keeps the higher priority until the used space in RX FIFO is less than or equal to the RX FIFO low threshold value. Then, TXDMA gets higher priority over RXDMA. Consequently, software must set RXFIFO_HTHR to be larger than RXFIFO_LTHR to keep GMAC work correctly. 0: Threshold = 0 1: Threshold = 1/8 space of RX FIFO 2: Threshold = 2/8 space of RX FIFO 3: Threshold = 3/8 space of RX FIFO 4: Threshold = 4/8 space of RX FIFO 5: Threshold = 5/8 space of RX FIFO 6: Threshold = 6/8 space of RX FIFO 7: Threshold = 7/8 space of RX FIFO
[2:0]	R/W	0	RXFIFO_LTHR	RX FIFO low threshold value for arbitration When the used space in RX FIFO is less than or equal to the RX FIFO low threshold value, TXDMA will have higher priority over RXDMA by using the DMA channel. 0: Threshold = 0 1: Threshold = 1/8 space of RX FIFO 2: Threshold = 2/8 space of RX FIFO 3: Threshold = 3/8 space of RX FIFO 4: Threshold = 4/8 space of RX FIFO 5: Threshold = 5/8 space of RX FIFO 6: Threshold = 6/8 space of RX FIFO 7: Threshold = 7/8 space of RX FIFO

Recommended value = 0x0002_2F72

12.2.3.16 DMA/FIFO State Register (DMAFIFOS, Offset = 0x3C)

Table 12-17. DMA/FIFO State Register (DMAFIFOS, Offset = 0x3C)

Bit	RW Type	Default Value	Name	Description
31	RO	0	TXD_REQ	TXDMA request
30	RO	0	RXD_REQ	RXDMA request
29	RO	0	DARB_TXGNT	TXDMA grant
28	RO	0	DARB_RXGNT	RXDMA grant
27	RO	1	TXFIFO_EMPTY	TX FIFO is empty.
26	RO	1	RXFIFO_EMPTY	RX FIFO is empty.
[25:22]	-	-	-	Reserved
[21:18]	RO	0	TXDMA3_SM	TXDMA 3 state machine The state machine is in charge of the read data flow from TX FIFO to TX pre-buffer.
[17:16]	RO	0	TXDMA2_SM	TXDMA 2 state machine The state machine is in charge of the burst read/write of transmit descriptor and buffer.
[15:12]	RO	0	TXDMA1_SM	TXDMA 1 state machine The state machine is in charge of the read/write of transmit descriptor and buffer.
11	-	-	-	Reserved
[10:8]	RO	0	RXDMA3_SM	RXDMA 3 state machine The state machine is in charge of RXDMA PVCI interface read/write.
[7:4]	RO	0	RXDMA2_SM	RXDMA 2 state machine The state machine is in charge of the burst read/write of receive descriptor and buffer.
[3:0]	RO	0	RXDMA1_SM	RXDMA 1 state machine The state machine is in charge of the read/write of receive descriptor and buffer.

12.2.3.17 Revision Register (REVR, Offset = 0x40)

Table 12-18. Revision Register (REVR, Offset = 0x40)

Bit	RW Type	Default Value	Name	Description
[31:24]	-	-	-	Reserved

Bit	RW Type	Default Value	Name	Description
[23:16]	RO	-	REV_B1	First digit of the revision tag For example, if the revision tag is version_1_2_r3, this filed will be 1.
[15:8]	RO	-	REV_B2	Second digit of the revision tag For example, if the revision tag is version_1_2_r3, this filed will be 2.
[7:0]	RO	-	REV_B3	Third digit of the revision tag For example, if the revision tag is version_1_2_r3, this filed will be 3.

Please note that this register is only for Faraday internal use.

12.2.3.18 Feature Register (FEAR, Offset = 0x44)

Table 12-19. Feature Register (FEAR, Offset = 0x44)

Bit	RW Type	Default Value	Name	Description
[31:6]		-	-	Reserved
[5:3]	RO	-	TFIFO_RSIZE	TX FIFO real size The FIFO sizes are as follows: 0: 2K 1: 4K 2: 8K 3: 16K 4: 32K 5 ~ 7: Reserved
[2:0]	RO	-	RFIFO_RSIZE	RX FIFO real size The FIFO sizes are as follows: 0: 2K 1: 4K 2: 8K 3: 16K 4: 32K 5 ~ 7: Reserved

12.2.3.19 Transmit Priority Arbitration and FIFO Control Register (TPAFCR, Offset = 0x48)

Table 12-20. Transmit Priority Arbitration and FIFO Control Register (TPAFCR, Offset = 0x48)

Bit	RW Type	Default Value	Name	Description
[31:30]	-	-	-	Reserved
[29:27]	R/W	0	TFIFO_SIZE	TX FIFO size Software can program this field to decide the TX FIFO size. Before programming this field, software must read the actual TX FIFO size in the FEATURE register. It is not allowed to program a value of larger than the actual TX FIFO size. The FIFO sizes are as follows: 0: 2K 1: 4K 2: 8K 3: 16K 4: 32K 5 ~ 7: Reserved
[26:24]	R/W	0	RFIFO_SIZE	RX FIFO Size Software can program this field to decide the RX FIFO size. Before programming this field, software must read the actual RX FIFO size in the FEATURE register. It is not allowed to program a value of larger than the actual RX FIFO size. The FIFO sizes are as follows: 0: 2K 1: 4K 2: 8K 3: 16K 4: 32K 5 ~ 7: Reserved

Bit	RW Type	Default Value	Name	Description
[23:16]	R/W	0	EARLY_TXTHR	Early Transmit Threshold This field specifies the threshold level in TX FIFO to begin transmission. When the byte count of the data in TX FIFO reaches the threshold or there is at least one packet in TX FIFO, hardware would begin to transmit the packet to network. Writing 0 to this field indicates that hardware should begin to transmit the packet after one whole packet has been saved in TX FIFO. The value software programs in this field should be less than the TX FIFO size. The unit is 64bytes.
[15:8]	R/W	0	EARLY_RXTHR	Early Receive Threshold This field specifies the threshold level in RX FIFO to move packet data to system memory. When the byte count of the data in RX FIFO reaches the threshold or there is at least one packet in RX FIFO, hardware would begin to move the packet from RX FIFO to system memory. Writing 0 to this field indicates that hardware should begin to move the packet after one whole packet has been stored in RX FIFO. The value software programs in this field should be less than the RX FIFO size. The unit is 64bytes.
[7:4]	R/W	0xF	HPKT_THR	High Priority Transmit Packet Threshold When the packet number of TXDMA moving from the high-priority transmit ring to TX FIFO is less than the threshold and the high-priority packet is still available, TXDMA would switch to service the high-priority transmit ring if TXDMA is servicing the normal-priority transmit ring. If TXDMA is servicing the high-priority transmit ring at that time, it would continue to service the high-priority transmit ring. When the packet number of TXDMA moving from the high-priority transmit ring to TX FIFO is equal to or greater than the threshold and the normal-priority packet is still available, TXDMA would switch to service the high-priority transmit ring. The hardware behavior is the same whether writing 0 or 1 to this field.

Bit	RW Type	Default Value	Name	Description
[3:0]	R/W	1	NPKT_THR	Normal Priority Transmit Packet Threshold Under the following conditions, TXDMA will switch to serve the high-priority transmit ring from the normal-priority transmit ring: When the high-priority packet number is equal to or less than HPKT_THR and the high-priority packet is available, TXDMA will switch to serve the high-priority transmit ring after it finishes serving a normal-priority transmit packet. When the packet number of TXDMA moving from the normal-priority transmit ring to TX FIFO is equal to or greater than the threshold and the high-priority packet number is available, TXDMA will switch to serve the high-priority transmit ring. When the high-priority packet is available and the normal-priority packet is not available, TXDMA will switch to serve the high-priority transmit ring. The hardware behavior is the same whether writing 0 or 1 to this field.

12.2.3.20 Receive Buffer Size Register (RBSR, Offset = 0x4C)

Table 12-21. Receive Buffer Size Register (RBSR, Offset = 0x4C)

Bit	RW Type	Default Value	Name	Description
[31:14]	-	-	-	Reserved
[13:0]	R/W	0x640	RXBUF_SIZE	Receive buffer size The unit is 1byte. The receive buffer size must be 8byte alignment.

12.2.3.21 MAC Control Register (MACCR, Offset = 0x50)

Table 12-22. MAC Control Register (MACCR, Offset = 0x50)

Bit	RW Type	Default Value	Name	Description
31	R/W	0	SW_RST	Software reset Writing 1 to this bit enables software reset. The software reset will last 175 AHB bus clocks, and then be auto-cleared.
[30:20]	-	-	-	Reserved

Bit	RW Type	Default Value	Name	Description
19	R/W	0	SPEED_100	Speed mode 1:100Mbps 0:10Mbps This field cannot be software reset.
18	R/W	0	DISCARD_CRCERR	Discard the CRC error packet if there is CRC error status in the transmit packet.
17	R/W	0	RX_BROADPKT_EN	Receive the broadcast packets
16	R/W	0	RX_MULTIPKT_EN	Receive all multicast packets
15	R/W	0	RX_HT_EN	Enable storing the incoming packet if the packet passes hash table address filtering and is a multicast packet.
14	R/W	0	RX_ALLADR	Destination address of the incoming packet is not checked.
13	R/W	0	JUMBO_LF	Jumbo Long Frame When set, packets with a length of more than 9216bytes (9220bytes for packets with VLAN tag) are treated as long frames. When cleared, packets with a length of more than 1518bytes (1522bytes for packets with VLAN tag) are treated as long frames.
12	R/W	0	RX_RUNT	Receive the incoming packet even if its length is less than 64bytes. The incoming packet length must be longer than or equal to 10bytes.
11	-	-	-	Reserved
10	R/W	0	CRC_APD	Append CRC to transmitted packets
9	R/W	0	GMAC_MODE	GMAC mode (It must be set to 0.) This field cannot be software reset.
8	R/W	0	FULLDUP	Full duplex If FULLDUP = ‘1’, RMII MAC will be in the full-duplex mode; otherwise, RMII MAC will be in the half-duplex mode.

Bit	RW Type	Default Value	Name	Description
7	R/W	0	ENRX_IN_HALFTX	Enable packet reception when transmitting packets in the half-duplex mode
6	R/W	0	LOOP_EN	Internal loopback enable This field cannot use the software reset.
5	R/W	0	HPTXR_EN	High priority transmit ring enable If HPTXR_EN = '1', software will use the high-priority transmit ring; otherwise, software will not use the high-priority transmit ring.
4	R/W	0	REMOVE_VLAN.	Remove the VLAN tag from the packets received with the VLAN tag
3	R/W	0	RXMAC_EN	RXMAC enable When set, enable RXMAC to receive the packets
2	R/W	0	TXMAC_EN	TXMAC enable When set, enable TXMAC to transmit the packets
1	R/W	0	RXDMA_EN	Enable the receive DMA channel If this bit is zero, reception will be immediately stopped.
0	R/W	0	TXDMA_EN	Enables transmit DMA channel If this bit is zero, transmission will be immediately stopped.

12.2.3.22 MAC Status Register (MACSR, Offset = 0x54)

Table 12-23. MAC Status Register (MACS, Offset = 0x54)

Bit	RW Type	Default Value	Name	Description
[31:12]	-	-	-	Reserved
11	R/W1C	0	COL_EXCEED	Collision amount exceeds 16.
10	R/W1C	0	LATE_COL	Transmitter detects late collision.
9	R/W1C	0	XPKT_LOST	Packets transmitted to Ethernet were lost due to late collision or excessive collision.
8	R/W1C	0	XPKT_OK	Packets are successfully transmitted to Ethernet.
7	R/W1C	0	RUNT	Receiver detects a runt packet.
6	R/W1C	0	FTL	Receiver detects that a frame is too long.
5	R/W1C	0	CRC_ERR	Incoming packet CRC check result is invalid, unless the CRC_DIS bit is set.
4	R/W1C	0	RPKT_LOST	Received packets were lost due to RX FIFO full.
3	R/W1C	0	RPKT_SAVE	Packets are successfully received to RX FIFO.

Bit	RW Type	Default Value	Name	Description
2	R/W1C	0	COL	Incoming packet is dropped due to collision.
1	R/W1C	0	BROADCAST	Incoming packet is for the broadcast address.
0	R/W1C	0	MULTICAST	Incoming packet is for the multicast address.

12.2.3.23 Test Mode Register (TM, Offset = 0x58)

Table 12-24. Test Mode Register (TM, Offset = 0x58)

Bit	RW Type	Default Value	Name	Description
[31:21]	-	-	-	Reserved
20	R/W	0	PTIMER_TEST	Automatic polling timer test mode
19	R/W	0	ITIMER_TEST	Interrupt timer test mode
[18:16]	-	-	-	Reserved
15	R/W	0	TEST_COL	Transmit collision test mode
[14:5]	R/W	0	TEST_BKOFF	Back-off value in the transmission collision test mode
[4:0]	R/W	0	TEST_EXSTHR	Retry upper limit in the transmit collision test mode

12.2.3.24 PHY Control Register (PHYCR, Offset = 0x60)

Table 12-25. PHY Control Register (PHYCR, Offset = 0x60)

Bit	RW Type	Default Value	Name	Description
[31:28]	-	-	-	Reserved
27	R/W	0	MIIWR	Setting this bit to 1 initializes a write sequence to PHY. This bit will be automatically cleared after the write operation is finished.
26	R/W	0	MIIRD	Setting this bit to 1 initializes a read sequence to PHY. This bit will be automatically cleared after the read operation is finished.
[25:21]	R/W	0	REGAD	PHY register address
[20:16]	R/W	0	PHYAD	PHY address
[15:8]	-	-	-	Reserved

Bit	RW Type	Default Value	Name	Description
[7:0]	R/W	0x52	MDC_CYCTHR	MDC cycle threshold This field defines the period of MDC. The MDC period = MDC_CYCTHR x system clock period. Users must set the correct value before using MDC.

12.2.3.25 PHY Data Register (PHYDATA, Offset = 0x64)

Table 12-26. PHY Data Register (PHYDATA) (Offset = 0x64)

Bit	RW Type	Default Value	Name	Description
[31:16]	RO	0	MIIRDATA	Read data from PHY
[15:0]	R/W	0	MIIWDATA	Write data to PHY

12.2.3.26 Flow Control Register (FCR, Offset = 0x68)

Table 12-27. Flow Control Register (FCR, Offset = 0x68)

Bit	RW Type	Default Value	Name	Description
[31:16]	R/W	0	PAUSE_TIME	Pause time of a pause frame The unit is 1 slot time.
[15:9]	R/W	2	FC_HIGH/ FC_LOW	RX FIFO free space high threshold A pause frame is sent with pause time = '0' when the RX FIFO free space is larger than the high threshold. The unit is 256bytes, and the default value is 5. RX FIFO free space low threshold A pause frame is sent with setting the pause time in bits 31 ~ 16 when the RX FIFO free space is less than the low threshold. The unit is 256bytes, and the default value is 2. When FC_HTHR_SEL = '1', the RX FIFO free space high threshold is selected. When FC_HTHR_SEL = '0', the RX FIFO free space low threshold is selected. The value software programs in this field should be less than the RX FIFO size.
8	R/W	0	FC_HTHR_SEL	RX FIFO free space high threshold select When set, the RX FIFO free space high threshold is selected. When cleared, the RX FIFO free space low threshold is selected.
[7:5]	-	-	-	Reserved

Bit	RW Type	Default Value	Name	Description
4	R/W1C	0	RX_PAUSE	Receive pause frame
3	RO	0	TXPAUSED	Packet transmission paused due to the receive pause frame
2	R/W	0	FCTHR_EN	Enable flow control threshold mode This bit enables the transmit pause frame for high/low threshold.
1	R/W	0	TX_PAUSE	Transmit pause frame Software can set this bit to send the pause frames. This bit is auto-cleared after the pause frame has been transmitted.
0	R/W	0	FC_EN	Flow control mode enable

12.2.3.27 Back Pressure Register (BPR, Offset = 0x6C)

Table 12-28. Back Pressure Register (BPR, Offset = 0x6C)

Bit	RW Type	Default Value	Name	Description
[31:15]	-	-	-	Reserved
[14:8]	R/W	4	BK_LOW	RX FIFO free space low threshold MAC generates a jam pattern if the RX FIFO free space is less than the low threshold when packets are incoming. The unit is 256bytes, and the default value is 2.
[7:4]	R/W	0	BKJAM_LEN	Back pressure jam length 0: 4bytes 6: 256bytes 1: 8bytes 7: 512bytes 2: 16bytes 8: 1024bytes 3: 32bytes 9: 1518bytes 4: 64bytes 10: 2048bytes 5: 128bytes Others: 4bytes
[3:2]	-	-	-	Reserved
1	R/W	0	BKADR_MODE	Back pressure address mode 1: Generate the jam pattern when packet address matches 0: Generate the jam pattern when any packet is incoming
0	R/W	0	BK_EN	Back pressure mode enable

12.2.3.28 Normal Priority Transmit Ring Pointer Register (NPTXR_PTR, Offset = 0x90)

Table 12-29. Normal Priority Transmit Ring Pointer Register (NPTXR_PTR, Offset = 0x90)

Bit	RW Type	Default Value	Name	Description
[31:0]	RO	-	NPTXR_PTR	Normal Priority Transmit Ring Pointer Register This register indicates the current value of the transmit descriptor pointer for the normal-priority transmit ring pointer register.

12.2.3.29 High Priority Transmit Ring Pointer Register (HPTXR_PTR, Offset = 0x94)

Table 12-30. High Priority Transmit Ring Pointer Register (HPTXR_PTR, Offset = 0x94)

Bit	RW Type	Default Value	Name	Description
[31:0]	RO	-	HPTXR_PTR	High Priority Transmit Ring Pointer Register This register indicates the current value of the transmit descriptor pointer for the high-priority transmit ring pointer register.

12.2.3.30 Receive Ring Pointer Register (RXR_PTR) (Offset = 0x98)

Table 12-31. Receive Ring Pointer Register (HPTXR_PTR) (Offset = 0x98)

Bit	RW Type	Default Value	Name	Description
[31:0]	RO	-	RXR_PTR	Receive Ring Pointer Register This register indicates the current value of the transmit descriptor pointer for the receive ring pointer register.

12.2.3.31 TPKT_CNT Counter Register (Offset = 0xA0)

Table 12-32. TPKT_CNT Counter Register (Offset = 0xA0)

Bit	RW Type	Default Value	Name	Description
[31:0]	RO	-	TPKT_CNT	Counter for counting packets transmitted successfully

12.2.3.32 TXMCOL_CNT and TXSCOL_CNT Counter Register (Offset = 0xA4)

Table 12-33. TXMCOL_CNT and TXSCOL_CNT Counter Register (Offset = 0xA4)

Bit	RW Type	Default Value	Name	Description
[31:16]	RO	0	TXMCOL_CNT	Counter for counting packets transmitted OK with 2 ~ 15 collisions
[15:0]	RO	0	TXSCOL_CNT	Counter for counting packets transmitted OK with single collision

12.2.3.33 TXECOL_CNT and TXFAIL_CNT Counter Register (Offset = 0xA8)

Table 12-34. TXECOL_CNT and TXFAIL_CNT Counter Register (Offset = 0xA8)

Bit	RW Type	Default Value	Name	Description
[31:16]	RO	0	TXFAIL_CNT	Counter for counting packets failed in transmission (Due to the late collision or collision count ≥ 16 or transmit underrun)
[15:0]	RO	0	TXECOL_CNT	Counter for counting packets failed in transmission (Due to the collision count ≥ 16)

12.2.3.34 TXLCOL_CNT and TXUNDERUN_CNT Counter Register (Offset = 0xAC)

Table 12-35. TXLCOL_CNT and TXUNDERUN_CNT Counter Register (Offset = 0xAC)

Bit	RW Type	Default Value	Name	Description
[31:16]	RO	0	TXUNDERUN_CNT	Counter for counting the packets failed in transmission (Due to transmit under-run)
[15:0]	RO	0	TXLCOL_CNT	Counter for counting the packets failed in transmission (Due to late collision)

12.2.3.35 RPKT_CNT Counter Register (Offset = 0xB0)

Table 12-36. RPKT_CNT Counter Register (Offset = 0xB0)

Bit	RW Type	Default Value	Name	Description
[31:0]	RO	32'h0	RPKT_CNT	Counter for counting the packets received successfully

12.2.3.36 BROPKT_CNT Counter Register (Offset = 0xB4)

Table 12-37. BROPKT_CNT Counter Register (Offset = 0xB4)

Bit	RW Type	Default Value	Name	Description
[31:0]	RO	-	BROPKT_CNT	Counter for counting the received broadcast packets

12.2.3.37 MULPKT_CNT Counter Register (Offset = 0xB8)

Table 12-38. MULPKT_CNT Counter Register (Offset = 0xB8)

Bit	RW type	Default value	Name	Description
[31:0]	RO	-	MULPKT_CNT	Counter for counting the received multicast packets

12.2.3.38 RPF_CNT and AEP_CNT Counter Register (Offset = 0xBC)

Table 12-39. RPF_CNT and AEP_CNT Counter Register (Offset = 0xBC)

Bit	RW Type	Default Value	Name	Description
[31:16]	RO	0	RPF_CNT	Receive pause frame counter
[15:0]	RO	0	AEP_CNT	Counter for counting the packets with alignment error The counter is to count packets with CRC error and no octet-boundary discarded by RMII MAC.

12.2.3.39 RUNT_CNT Counter Register (Offset = 0xC0)

Table 12-40. RUNT_CNT Counter Register (Offset = 0xC0)

Bit	RW Type	Default Value	Name	Description
[31:16]	-	-	-	Reserved
[15:0]	RO	0	RUNT_CNT	Counter for counting the received runt packets

12.2.3.40 CRCER_CNT and FTL_CNT Counter Register (Offset = 0xC4)

Table 12-41. CRCER_CNT and FTL_CNT Counter Register (Offset = 0xC4h)

Bit	RW Type	Default Value	Name	Description
[31:16]	RO	0	CRCER_CNT	CRC error packet counter The counter counts the number of the octet-boundary frames discarded due to the CRC error.
[15:0]	RO	0	FTL_CNT	Counter for counting received FTL packets

12.2.3.41 RCOL_CNT and RLOST_CNT Counter Register (Offset = 0xC8)

Table 12-42. RCOL_CNT and RLOST_CNT Counter Register (Offset = 0xC8)

Bit	RW Type	Default Value	Name	Description
[31:16]	RO	0	RLOST_CNT	Counter for counting the loss of the received packets (Due to RX FIFO full)
[15:0]	RO	0	RCOL_CNT	Receive collision counter

12.2.3.42 Advance Interrupt Timer Control Register (ITC, Offset = 0xE0)

Table 12-43. Advance Interrupt Timer Control Register (ITC, Offset = 0xE0)

Bit	RW Type	Default Value	Name	Description
[31:10]	-	-	-	Reserved
[9:0]	R/W	0	RXINT_RST	This field defines the refresh time. The reset counter will be cleared once one packet is received by RMII MAC. The time unit is 1 RX cycle time. When RXINT_CNT = ‘0’, the function will be disabled. If RXINT_THR = 0 and RXINT_CNT = ‘0’, a receive interrupt will be issued when a packet is received by RMII MAC.

12.2.4 Functional Description

12.2.4.1 Transmit Descriptors and Data Buffers

RMII MAC uses a descriptor ring to manage the transmit buffers. The transmit descriptors and data buffers are all located in the system memory. RMII MAC moves the transmit packet data from the transmit buffers in system memory to TX FIFO inside RMII MAC and then transmits the packet to Ethernet. The transmit descriptors reside in the system memory act as pointers to the transmit buffers.

Each transmit descriptor contains a transmit buffer. A transmit buffer consists of either an entire frame or part of a frame, but not multiple frames. The transmit descriptor contains the transmit buffer status and the transmit buffer that can only contain the transmit data. RMII MAC supports two descriptor rings for transmission. These descriptor rings are the normal-priority transmit ring and high-priority transmit ring. The normal-priority transmit ring is for normal packet transmission; the high-priority transmit ring is for high-priority packet transmission. Higher priority packets can be put into the high-priority transmit ring for quicker transmission.

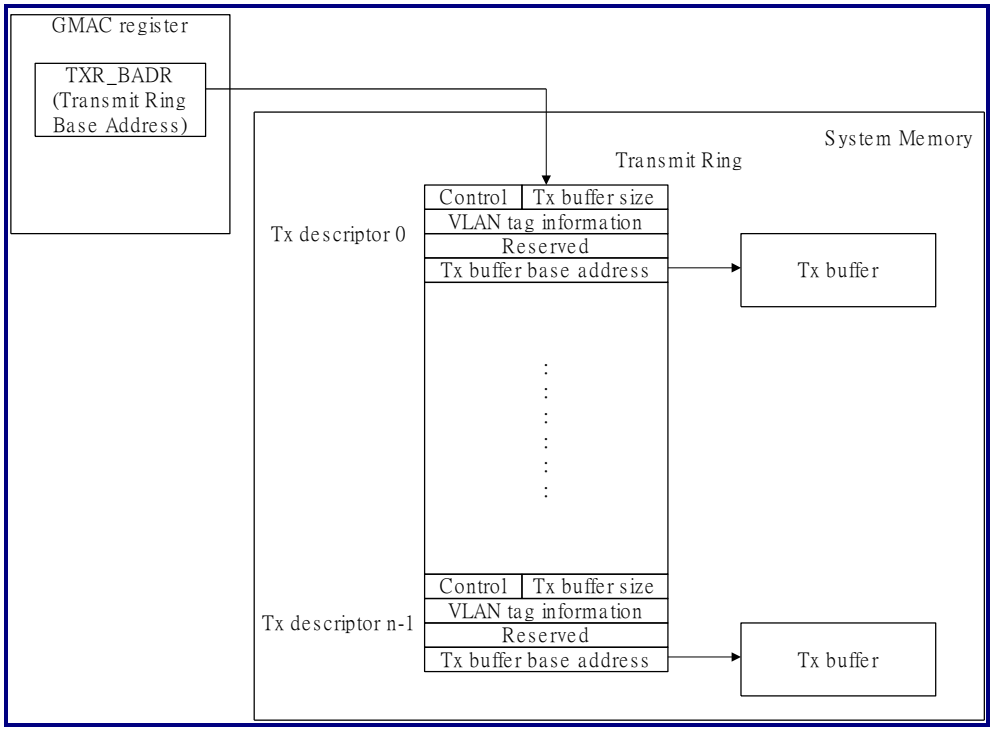


Figure 12-1. Transmit Ring Descriptor Structure

The transmit descriptor structure is as follows.

Notes:

- The start address of each transmit descriptor must be 16byte aligned.
- The maximum transmit packet size including CRC is 9216bytes (9220bytes if the VLAN tag is inserted).
- RMII MAC only supports the IPV4 checksum offload. Software must be certain that the transmit packet is an IPV4 packet when software requests RMII MAC to do checksum offload.
- LLC packet is a packet in the IEEE 802.3/802.2/SNAP format.
- RMII MAC does not support the following two packets for the checksum offload:
  - IEEE 802.3 with IEEE 802.2 packet
  - IEEE 802.3 with 802.1Q and 802.2 packets.

TXDES0	TXDMA_OWN	Control1	TX buffer size
TXDES1	VLAN Control	VLAN Tag Control Information	
TXDES2	Reserved		
TXDES3	TX buffer base address		

TXDES0 contains the control bits and transmit buffer size and descriptor ownership information.

Bit	Name	Description
31	TXDMA_OWN	TXDMA_OWN - TXDMA ownership bit When set, it indicates that the descriptor is owned by RMII MAC. When reset, it indicates that the software owns the descriptor. RMII MAC clears this bit when it completes the frame transmission.
30	-	Reserved
29	FTS	FTS - First Transmit Segment descriptor When set, it indicates that this is the first descriptor of a TX packet.
28	LTS	LTS - Last Transmit Segment descriptor. When set, it indicates that this is the last descriptor of a TX packet.
[27:20]	-	Reserved
19	CRC_ERR	CRC_ERR - CRC error When CRC_ERR = '1' and DISCARD_CRCERR, bit 18, of MAC Control Register (Offset: 0x50) = '1', TXDMA will discard the transmit packet and will not send it to Ethernet.
[18:16]	-	Reserved
15	EDOTR	EDOTR – End Descriptor of Transmit Ring When set, it indicates that the descriptor is the last descriptor of the transmit ring.

Bit	Name	Description
14	-	Reserved
[13:0]	TXBUF_SIZE	Transmit buffer size in byte The transmit buffer size cannot be zero.

TXDES1 contains the VLAN control bits and VLAN Tag Control Information.

Bit	Name	Description
31	TXIC	TXIC - Transmit Interrupt on Completion When set, RMII MAC will assert the transmit interrupt after the present frame has been transmitted. It is valid only when FTS = '1' and bits 8 ~ 14 (TXINT_THR, TXINT_CNT) of Interrupt Timer Control Register = '0'.
30	TX2FIC	TX2FIC - Transmit to FIFO Interrupt on Completion When set, RMII MAC will assert the transmit interrupt after the present frame has been moved into TX FIFO. It is valid only when FTS = '1'.
[29:23]	-	Reserved
22	LLC_PKT	LLC_PKT – LLC packet When set, RMII MAC will treat the packet as the LLC packet. It will be valid only when FTS = '1'.
[21:20]	-	Reserved
19	IPCS_EN	IPCS_EN – IP checksum offload enable When set, RMII MAC will offload the IP checksum. It will be valid only when FTS = '1'.
18	UDPCS_EN	UDPCS_EN – UDP checksum offload enable When set, RMII MAC will offload the UDP checksum. It will be valid only when FTS = '1'.
17	TCPCS_EN	TCPCS_EN – TCP checksum offload enable. When set, RMII MAC will offload the TCP checksum. It will be valid only when FTS = '1'.
16	INS_VLAN	Insert VLAN Tag When set, 0x8100 (IEEE 802.1Q VLAN Tag Type) will be inserted after the source address, and two bytes VLAN_TAGC will be inserted after the IEEE 802.1Q VLAN tag type. When clear, the packet content will not be changed when transmitting to network. It will be valid only when FTS = '1'.

Bit	Name	Description
[15:0]	VLAN_TAGC	VLAN Tag Control Information The 2byte VLAN Tag Control Information contains information, from the upper layer, of user priority, canonical format indicator, and VLAN ID. Please refer to IEEE 802.1Q for more VLAN tag information. Bits[15:13]: User priority Bit 12: CFI (Canonical Format Indicator) Bits[11:0]: VID (VLAN Identifier) It will be valid only when FTS = '1'.

TXDES2 is reserved.

Bit	Name	Description
[31:0]	-	Reserved

TXDES3 contains the transmit buffer base address.

Bit	Name	Description
[31:0]	TXBUF_BADR	Transmit buffer base address

12.2.4.2 Receive Descriptors and Data Buffers

RMII MAC uses a descriptor ring to manage the receive buffers. The receive descriptors and data buffers are all located in the system memory. RMII MAC first stores the packet received from the network in the RX FIFO and then moves the received packet data to the receive buffers in system memory. The receive descriptors reside in the system memory act as pointers to the receive buffers.

There is a descriptor ring for reception. The base address of the receive ring is in the Receive Ring Base Address Register (RXR_BADR, offset: 24h ~ 27h). Each receive descriptor contains a receive buffer. A receive buffer consists of either an entire frame or part of a frame, but not multiple frames. The receive descriptor contains the receive buffer status and the receive buffer only contains the receive packet data.

RMII MAC supports the receive buffer base address as 2byte alignment for the zero-copy feature. But there is a limitation when software program the receive buffer base address as 2byte alignment. The limitation is the receive packet can only occupy one receive buffer. This means that the receive buffer size must be greater than the receive packet length. For example, if the length of the incoming packet is always less than 1600bytes, software can program the receive buffer size as 1600bytes. Then the limitation sustained when the receive buffer base address is 2byte alignment. If software program the receive buffer base address as 8byte alignment, then the limitation does not hold.

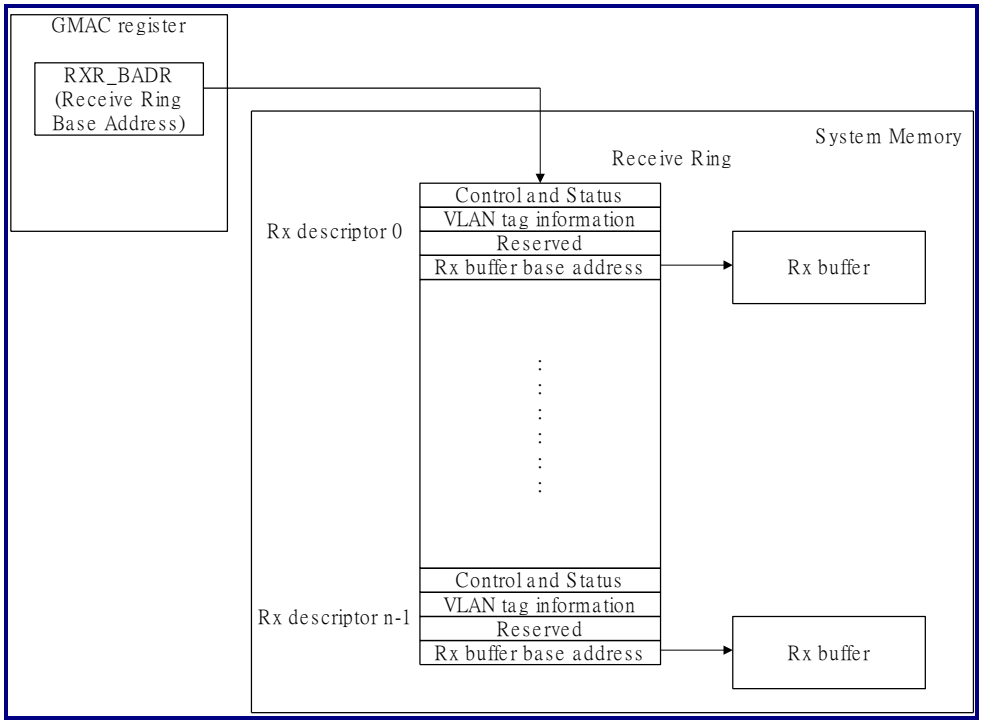


Figure 12-2. Receive Ring Descriptor Structure

The receive descriptor structure is as follows:

Notes:

- The start address of each receive descriptor must be 16byte aligned.
- The maximum receive packet size is 9216bytes (9220bytes for packets with the VLAN tag).
- RMII MAC only supports the IPV4 checksum offload. If the incoming packet is not an IPV4 packet, GMAC will not perform the checksum verification. The IPCS_FAIL, UDPCS_FAIL, and TCPCS_FAIL fields are always zeroes.
- LLC packet is a packet in the IEEE 802.3/802.2/SNAP format.
- RMII MAC does not support the following two packets for the checksum offload. They are IEEE 802.3 with IEEE 802.2 packet and IEEE 802.3 with 802.1Q and 802.2 packets.

- The receive buffer size must be greater than the receive packet length when software programs the receive buffer base address as 2byte alignment.

RXDES0	RXPKT_RDY	Status and Control
RXDES1	VLAN status	VLAN Tag Control Information
RXDES2	Reserved	
RXDES3	RX buffer base address	

RXDES0 contains the receive frame status and descriptor ownership information.

Bit	Name	Description
31	RXPKT_RDY	RXPKT_RDY – RX packet ready When cleared, it indicates that the descriptor is owned by RMII MAC. When set, it indicates that the descriptor is owned by software. RMII MAC sets this bit when it completes the frame reception or when the receive buffer of the receive descriptor is full.
30	-	Reserved
29	FRS	FRS - First Receive Segment descriptor When set, it indicates that this is the first descriptor of a received packet. Bits 25 ~ 0 are valid only when FRS = '1'.
28	LRS	LRS - Last Receive Segment descriptor When set, it indicates that this is the last descriptor of a received packet.
[27:26]	-	Reserved
25	PAUSE_FRMAE	PAUSE_FRAME – Pause frame When set, it indicates that the receive packet is a pause frame.
24	PAUSE_OPCODE	PAUSE_OPCODE – Pause frame OP code When set, it indicates that there is pause frame OP code in the receive packet.
23	FIFO_FULL	FIFO_FULL – FIFO full When set, it indicates that RX FIFO is full when the packet is received.
22	RX_ODD_NB	RX_ODD_NB - Receive Odd Nibbles When set, it indicates that a packet is received with odd nibbles.
21	RUNT	RUNT - Runt packet When set, it indicates that the received packet length is less than 64bytes.

Bit	Name	Description
20	FTL	FTL - Frame Too Long When set, it indicates that the received packet length exceeds the long frame length. If JUMBO_LF = '0', the long frame length will be 1518bytes (1522bytes for the receive packet with the VLAN tag). If JUMBO_LF = '1', the long frame length will be 9216bytes (9220bytes for the receive packet with the VLAN tag).
19	CRC_ERR	CRC_ERR - CRC error When set, it indicates that the CRC error occurs on the received packet.
18	RX_ERR	RX_ERR - Receive error When set, it indicates that receive error happens when receiving a packet.
17	BROADCAST	BROADCAST - Broadcast frame. When set, it indicates that the received packet is a broadcast frame.
16	MULTICAST	MULTICAST - Multicast frame When set, it indicates that the received packet is a multicast frame.
15	EDORR	EDORR - End Descriptor of Receive Ring When set, it indicates that the descriptor is the last descriptor of the receive ring.
14	-	Reserved
[13:0]	VDBC	VDBC - valid data byte count The field indicates the valid data in the receive buffer. The unit is 1byte.

RXDES1 contains the VLAN status bits and VLAN Tag Control Information.

Bit	Name	Description
[31:28]	-	Reserved
27	IPCS_FAIL	IPCS_FAIL - IP checksum failure When set, RMII MAC detects IP checksum failure. The field will be valid only when FRS = '1'.
26	UDPCS_FAIL	UDPCS_FAIL - UDP checksum failure When set, RMII MAC detects UDP checksum failure. The field will be valid only when FRS = '1'.
25	TCPCS_FAIL	TCPCS_FAIL - TCP checksum failure When set, RMII MAC detects TCP checksum failure. The field will be valid only when FRS = '1'.

Bit	Name	Description
24	VLAN_AVA	VLAN_AVA - VLAN Tag Available When set, the receive packet is an IEEE 802.1Q VLAN Tag Type. The field will be valid only when FRS = '1'.
23	DF	Datagram Fragment When set, the IP packet is not fragment. When clear, the IP packet may fragment. Checksum status is valid only when DF = '1'.
22	LLC_PKT	LLC_PKT - LLC packet When set, it indicates that the receive packet is the LLC packet. The field will be valid only when FRS = '1'.
[21:20]	PROTL_TYPE	PROTL_TYPE - Protocol Type These two bits indicate which protocol in the receive packet. 00: Not IP protocol 01: IP protocol 10: TCP/IP protocol 11: UDP/IP protocol The field will be valid only when FRS = '1'.
[19:16]	-	Reserved
[15:0]	VLAN_TAGC	VLAN Tag Control Information If the receive packet contains the VLAN tag, RMII MAC will extract 4bytes from the receive packet. The 4bytes data contains 0x8100 and 2bytes VLAN Tag Control Information. RMII MAC will move 2bytes VLAN Tag Control Information to this field. The 2byte VLAN Tag Control Information contains information from the upper layer of the user priority, canonical format indicator, and VLAN ID. Please refer to IEEE 802.1Q for more VLAN tag information. Bits[15:13]: User priority Bit 12: CFI (Canonical Format Indicator) Bits[11:0]: VID (VLAN Identifier) The field will be valid only when FRS = '1'.

RXDES2 is reserved.

Bit	Name	Description
[31:0]	-	Reserved

RXDES3 contains the receive buffer base address.

Bit	Name	Description
[31:0]	RXBUF_BADR	Receive buffer base address Receive buffer base address must be at least 2byte alignment. This means that RXBUF_BADR[0] must be zero.

12.2.4.3 Transmitting Packets

When software wants to transmit a packet to Ethernet, it moves the packet data into the transmit buffer first. Then, software writes the packet length and position into the transmit descriptor and triggers RMII MAC to send the packet. After the entire packet has been moved into TX FIFO, RMII MAC begins to transmit it to Ethernet. When the packet has been transmitted, RMII MAC asserts interrupt to notify software that the packet has been transmitted successfully. Higher priority packets can be put into the high priority descriptor for quicker transmission.

12.2.4.4 Receiving Packets

When there is an incoming packet, RMII MAC first saves the received packet in RX FIFO if the address check result is correct. After the incoming packet is successfully saved in RX FIFO, RMII MAC initiates Direct Memory Access (DMA) function to move the received packet data from RX FIFO to the system memory. Then, RMII MAC asserts interrupt to notify software that the packet has been received successfully.

12.2.4.5 Ethernet Address Filtering

RMII MAC can be set to recognize any Ethernet receive address group described in the following table.

- RX_BROADPKT: Bit 17 of MAC Control Register (Offset: 0x50)
- RX_MULTIPKT: Bit 16 of MAC Control Register (Offset: 0x50)
- RX_HT: Bit 15 of MAC Control Register (Offset: 0x50)
- RX_ALLADR: Bit 14 of MAC Control Register (Offset: 0x50)

RX_ ALLADR	RX_ MULTIPKT	RX_ BROADPKT	RX_HT	Group	Description
0	0	0	0	A	RMII MAC receives a frame of which the destination address exactly matches MAC_ADR (Offset: 0x8) of GMAC.
0	0	0	1	B	RMII MAC receives a frame of which the destination address exactly matches MAC_ADR (Offset: 0x8) of RMII MAC, a frame of which the destination address is a multicast address, or a frame which passes address filtering of the multicast address hash table in RMII MAC.
0	0	1	0	C	RMII MAC receives a frame of which the destination address exactly matches MAC_ADR (Offset: 0x8) of RII MAC, or a frame of which the destination address is a broadcast address.
0	0	1	1	D	RMII MAC receives a frame of which the destination address exactly matches MAC_ADR (Offset: 0x8) of RMII MAC, a frame of which the destination address is a multicast address, a frame which passes the address filtering of the multicast address hash table in RMII MAC, or a frame of which the destination address is a broadcast address.
0	1	X	X	E	RMII MAC receives a frame of which the destination address exactly matches MAC_ADR (Offset: 0x8) of RMII MAC or a frame of which the destination address is a multicast address.
1	X	X	X	F	RMII MAC supports reception of all frames on the network regardless of the destination addresses.

12.2.4.6 DMA Arbitration Scheme

The DMA arbitration scheme is decided by RX_THR_EN (Bit 6 of DMA Burst Length and Arbitration Control Register, Offset: 0x38). When RX_THR_EN = 0, the DMA arbitration scheme does a fair arbitration between TXDMA and RXDMA. If both TXDMA and RXDMA request the DMA channel at the same time, the one last using the DMA channel has lower priority to get the DMA channel.

When RX_THR_EN is set, if the used space in RX FIFO is larger than or equal to RXFIFO_HTHR (Bits 5 ~ 3 of DMA Burst Length and Arbitration Control Register, Offset: 0x38), the RXDMA has higher priority over the TXDMA by using the DMA channel. The RXDMA keeps the higher priority until the used space in RX FIFO is less than or equal to RXFIFO_LTHR (Bits 2 ~ 0 of DMA Burst Length and Arbitration Control Register, Offset: 0x38). Then, TXDMA gets higher priority over RXDMA. So, software must set RXFIFO_HTHR to be larger than RXFIFO_LTHR to keep RMII MAC working correctly.

12.2.4.7 Flow Control

RMII MAC implements flow control function. It supports IEEE802.3x flow control for full-duplex mode and backpressure for half-duplex mode.

The IEEE802.3x flow control is used in the full-duplex mode. When A and B are reciprocally transmitting and receiving packets in the full-duplex mode, if RX FIFO in B is nearly full, B sends a pause frame to A in order to avoid loss of received packet. Then A is inhibited from transmitting packets for a specified period of time. B consumes the received data during the specified period of time. A continues to send packets to B after the pause time has lapsed. Here is a brief account of the features of the flow control in the full-duplex mode:

- The software configures the pause time of the pause frame.
- The controller sends the pause frame according to the low/high threshold of RX FIFO.
- The software sends the pause frame by writing to register.

The back pressure mode is used in the half-duplex operation. When A is transmitting and receiving packets in the half-duplex mode, if RX FIFO in A is nearly full, A will send a jam pattern to generate collisions to avoid incoming packets from being saved into RX FIFO. A consumes the received data as soon as possible during the period of time. A does not send a jam pattern to receive packets again when RX FIFO is not nearly full. Here is a brief account of the features of the back pressure mode:

- Software configures the length of the jam.
- RMII MAC sends jam according to the low/high threshold of RX FIFO.

12.2.4.8 Zero-copy

With the zero-copy function of the controller, the system does not need to perform data movement for the packet header alignment. DMA of the controller will place the incoming packet data from the 2byte-aligned address if the receive buffer is programmed as the 2byte-aligned address. Figure 12-3 shows an example of the packet placement in a little-endian system.

The controller supports the 2byte-aligned receive buffer base address for the zero-copy function. But there will be a limitation when software programs the receive buffer base address as 2byte alignment. The limitation is that the receive packet can only occupy one receive buffer. This means that the size of the receive buffer must be greater than the receive packet length. For example, if the length of the incoming packet is less than 1600bytes, software can program the size of the receive buffer to 1600bytes. Then, the limitation can be achieved when the receive buffer base address is 2byte alignment. If software programs the receive buffer base address to be 8byte alignment, the limitation will not hold.

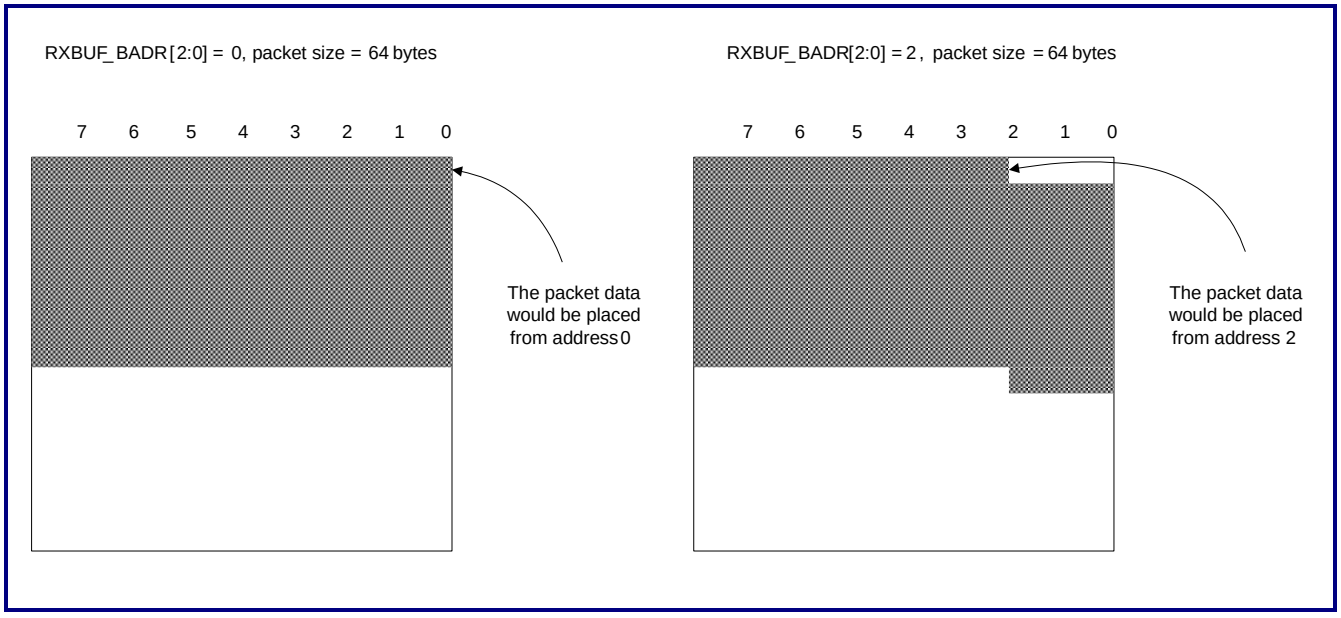


Figure 12-3. Packet Data Placement for Different Receive Buffer Address

12.2.4.9 Supported Ethernet Frame Type for Checksum Offload

Two Ethernet frame formats are supported in order to correctly identify the IP and TCP/UDP headers. The RMII MAC supports Ethernet Type II format and IEEE 802.3/802.2/SNAP format as shown below. VLAN tagging is also supported on top of these two frame formats.

Ethernet Type II

Destination Address	Source Address	Type	Data	CRC
6bytes	6bytes	2bytes	nbytes	4bytes

IEEE 802.3/802.2/SNAP

Destination Address	Source Address	Length	DSAP	SSAP	Control	OUI	Type	Data	CRC
6bytes	6bytes	2bytes	1byte	1byte	1byte	3bytes	2bytes	nbytes	4bytes
where DSAP = AA, SSAP = AA, Control = 03, and OUI = 000000									

12.2.5 Initialization/Application Information

12.2.5.1 Frame Transmitting Procedure

The frame transmitting procedure is as follows:

Initialization

- Set LOOP_EN (Offset: 0x50 bit[6]), GMAC_MODE (Offset: 0x50 bit[9]) and SPEED_100 (Offset: 0x50 bit[19]) for proper setting
- Set SW_RST (Offset: 0x50 bit[31]) = '1' to do software reset. It takes about 200 system clock cycles for hardware to finish the software reset
- Read the Feature Register (Offset: 0x44) to get the real TX/RX FIFO size in hardware
- Allocate the system memory for the transmit descriptor ring and transmit buffer
- Initialize the transmit descriptor ring
- Set the Normal Priority Transmit Ring Base Address Register (Offset: 0x20) to the base address of the normal priority transmit descriptor ring in the system memory

- Set the High Priority Transmit Ring Base Address Register (Offset: 0x2C) to the base address of the high priority transmit descriptor ring in the system memory if necessary
- Set the Interrupt Enable Register (Offset: 0x4)
- Set the MAC Address Register (Offset: 0x8)
- Set the Multicast Address Hash Table Register (Offset: 0x10)
- Set the Interrupt Timer Control Register (Offset: 0x30) to select the manner of the transmit interrupt.
- Set the Automatic Polling Timer Control Register (Offset: 0x34) to select the manner of transmit poll
- Set the Transmit Priority Arbitration and FIFO Control Register (Offset: 0x48) to set transmit priority arbitration and proper TX/RX FIFO size in use
- Set the DMA Burst Length and Arbitration Control Register (Offset: 0x38) for proper setting the TX/RX descriptor size and DMA burst length
- Set the MAC Control Register (Offset: 0x50) to set valid configuration for RMII MAC and to enable the transmit channel

Transmit procedure

- Software checks if the remained normal-priority transmit descriptors is sufficient for the next packet transmission. If not, software needs to wait until the transmit descriptors are sufficient.
- Prepare the transmit packet data to the transmit buffer.
- Set the normal priority transmit descriptor.
- Write the Normal Priority Transmit Poll Demand Register (Offset: 0x18) to trigger RMII MAC to poll the transmit descriptor if necessary when the packet is put in the normal priority transmit ring.
- Wait for interrupt.
- When interrupt occurs, software checks if it is a transmit interrupt. If ISR[4] = '1', it means that the packet has been transmitted to network successfully. If ISR[7] = '1', it means that the packet has been aborted during transmission due to late collision or excessive collision or under-run.
- Steps 1 through 6 are for the normal packets in the normal-priority transmit ring. If software wants to transmit the high-priority packets, repeat these steps for the high-priority transmit ring.

Notes:

When setting the transmit descriptor, TXDES0 must be set last. Thus, the setting procedure should be as follows:

- Set TXDES3
- Set TXDES2
- Set TXDES1
- Set TXDES0

When preparing a transmit packet which contains more than one transmit descriptors, the first transmit descriptor must be the last set descriptor of the transmit packet.

12.2.5.2 Frame Receiving Procedure

The frame receiving procedure is as follows:

Initialization

- Set LOOP_EN (Offset: 0x50 bit[6]), GMAC_MODE (Offset: 0x50 bit[9]), and SPEED_100 (Offset: 0x50 bit[19]) for proper setting
- Set SW_RST (Offset: 0x50 bit[31]) = '1' to perform software reset. It takes about 200 system clock cycles for hardware to finish the software reset
- Read the Feature Register (Offset: 0x44) to get the real TX/RX FIFO size in hardware
- Allocate the system memory for the receive descriptor ring and receive buffer
- Initialize the receive descriptor ring
- Set the Receive Ring Base Address Register (Offset: 0x24) to the base address of the receive descriptor ring in the system memory
- Set the Interrupt Enable Register (Offset: 0x4)
- Set the MAC Address Register (Offset: 0x8)
- Set the Multicast Address Hash Table Register (Offset: 0x10)
- Set the Interrupt Timer Control Register (Offset: 0x28) to select the manner of receive interrupt
- Set the Automatic Polling Timer Control Register (Offset: 0x34) to select the manner of receive poll
- Set the Transmit Priority Arbitration and FIFO Control Register (Offset: 0x48) for the transmit priority arbitration and proper TX/RX FIFO size in use
- Set the DMA Burst Length and Arbitration Control Register (Offset: 0x38) for the proper TX/RX descriptor size and DMA burst length
- Set the MAC Control Register (Offset: 0x50) to set valid configuration for RMII MAC and to

enable receive channel

- Write the Receive Poll Demand Register (Offset: 0x1C) to trigger RMII MAC for polling the receive descriptor

Receive procedures

- Wait for interrupt
- When interrupt occurs, software checks if it is a receive interrupt. If ISR[0] = '1', it means the packet has been moved to the receive buffer successfully. Then, software needs to fetch the receive descriptor to get the receive packet until the owner bit of the next receive descriptor does not belong to software.
- Software releases the receive descriptors to RMII MAC after accessing the received packet.
- If the receive automatic poll function is disabled, software needs to write Receive Poll Demand Register (Offset: 0x1C) to trigger RMII MAC to poll the receive descriptor.

12.2.5.3 Procedures to Enter and Exit Power-down Mode

The procedure for entering into the power-down mode is as follows:

1. Set TXDMA_EN (Offset: 0x50 bit[0]) = '0'
2. Poll DMA/FIFO State Register (Offset: 0x3C) to wait for empty TX FIFO
3. Set TXMAC_EN (Offset: 0x50 bit[2]) = '0' to stop transmission
4. Set RXMAC_EN (Offset: 0x50 bit[3]) = '0'
5. Poll the DMA/FIFO State Register (Offset: 0x3C) to wait for empty RX FIFO
6. Set RXDMA_EN (Offset: 0x50 bit[1]) = '0' to stop reception
7. Program Wake-up Frame CRC Register (Offset: 0x78), Wake-up Frame Byte Mask 1st Double-word Register (Offset: 0x80), Wake-up Frame Byte Mask 2nd Double-word Register (Offset: 0x84), Wake-up Frame Byte Mask 3rd Double-word Register (Offset: 0x88), and Wake-up Frame Byte Mask 4th Double-word Register (Offset: 0x8C), if software wants to support wake-up frame event in power saving mode
8. Write 0xFFFF_FFFF to clear the Wake-On-LAN Status Register (0x74)
9. Program the requested wake-up events and power state into the Wake-On-LAN Register (Offset: 0x70) to let RMII MAC enter the power-saving mode
10. Set RXMAC_EN (Offset: 0x50 bit[3]) = '1' to enable the reception

The procedure for exiting from the power-down mode is as follows:

1. Wait for occurrence of wake-up events
2. Set RXMAC_EN (Offset: 0x50 bit[3]) = '0' to stop the reception
3. Read the Wake-On-LAN Status Register (Offset: 0x74) to check which wake-up event happened
4. Program the Wake-On-LAN Register (Offset: 0x70) to let RMII MAC exit the power-saving mode
5. Set SW_RST (Offset: 0x50 bit[31]) = '1' to reset RMII MAC
6. Check if SW_RST (Offset: 0x50 bit[31]) = '0' to make sure that RMII MAC has finished reset
7. Re-initialize RMII MAC to transmit and receive packets

12.2.6 Ethernet Frame Formats

Table 12-44. Frame Structure and Frame Formats

Preamble	SFD	DA	SA	LEN	LLC	PAD	CRC
7byte	1byte	6byte	6byte	2byte	0 ~ 1500bytes	0 ~ 46bytes	4byte

- Preamble - Seven identical bytes. The bits in each byte are 0xAA, transmitted from left to right. The preamble is not required for reception. A sequence of 56bits alternating between 1 and 0 is used for synchronization. The preamble gives components in the network time to detect the presence of a signal, and read the signal before the frame data arrives.
- Start of Frame Delimiter (SFD) - One byte. This bit is 0xAB, transmitted from left to right. SFD is required for reception.
- Destination Address (DA) - Six bytes. DA may be an individual or a multicast or a broadcast address. The destination MAC address identifies the station or stations to receive the frame. The source MAC address identifies the station that originated the frame. The 802.3 standard permits these address fields to be either 2byte or 6byte in length, but virtually all Ethernet implementations use 6byte addresses. A destination address may specify either an "individual address" destined for a single station, or a "multicast address" destined for a group of stations. A destination address of 1bit refers to all stations on the LAN and is called a "broadcast address".
- Source Address (SA) - Six bytes. The source MAC address identifies the station that originated the frame.
- Length (LEN) - Two bytes. The length value indicates the number of Logical Link Control (LLC) data bytes in the data field. If the value of this field is less than or equal to 1500, then the Length/Type field indicates the number of bytes in the subsequent MAC data field. If the value of this field is greater than or equal to 1536, then the Length/Type field indicates the nature of the MAC protocol (Protocol type).
- Logical Link Control (LLC) Data - The user data
- PAD - Pad to 46bytes. If the user data from DA is less than 46bytes, PAD is needed.
- Cyclic Redundancy Check (CRC) - Four bytes. CRC is a value computed as a function of all fields except the preamble, SFD, and CRC. This CRC field contains a 4byte CRC value used for error checking. When a source station assembles a MAC frame, it performs a CRC calculation on all bits in the frame from destination MAC address through the Pad fields (All fields, except the preamble, start

of frame delimiter, and frame check sequence). The source station stores the value in this field and transmits it as part of the frame. When the destination station receives the frame, it performs an identical check. If the calculated value does not match the value in this field, the destination station assumes that an error has occurred during transmission and discards the frame.

- Inter-frame Gap - Ethernet devices must allow a minimum idle period between transmission of frames known as the Inter-frame Gap (IFG) or Inter-Packet Gap (IPG). IFG or IPG provides a brief recovery time between frames to allow devices to prepare for reception of the next frame. The minimum inter-frame gap is 96 bit times, which is 9.6 microseconds for 10Mbps Ethernet and 0.96 microseconds for 100Mbps Ethernet.

12.3 Secure Digital Controller (SDC)

12.3.1 General Description

The secure digital controller is a host controller for accessing the Secure Digital (SD) card that conforms to the standard specifications, SDIO specification, and SD memory card physical layer specification. The SD host controller provides the programmable I/O and DMA for data transfers. DMA supports the data transfers between the memory and the SD card through the AHB Master interface without being interrupted by CPU. DMA also supports the single-block transfer, multi-block transfer, and infinite transfer. The system address register points to the address of the descriptor table and sequentially accesses the data until the end of a transfer.

12.3.2 Features

- Compliant with SD host controller standard specification, version 3.0
- Supports both DMA and non-DMA data transfers
- Compliant with SD physical layer specification, version 3.0
- Supports UHS50/UHS104 card
- Supports configurable SD bus mode: 4bit mode
- Compliant with SDIO card specification, version 2.0
- Compliant with MMC card specification, version 4.3
- Supports 1K SRAM data FIFO
- Supports configurable 1bit/4bit SD card bus and 1bit/4bit MMC card bus
- Built-in generation and checking for 7bit and 16bit CRC data
- Supports Read Wait mechanism for SDIO function
- Supports Suspend/Resume mechanism for SDIO function

12.3.3 Memory Map/Register Definition

This section provides the memory maps and detailed descriptions of all registers that are accessible to the end users. The descriptions include individual bit levels and the reset state of each register.

12.3.3.1 Summary of Register

Table 12-45 lists and defines the register types.

Table 12-45. Register Types

Register Type	Description
RO	Read-only register
ROC	Read-only status Bits are initialized to zero at reset.
WO	Write-only register
R/W	Register for the Read/Write operations
RW1C	Read-only status and Write '1' to clear register
RWAC	Register for the Read/Write operations with the automatically cleared function
HwInit	Initialized by the hardware configuration Bits are read-only.

Table 12-46 lists and defines the SDXC controller registers.

Table 12-46. Summary of SDXC Controller Registers

Offset	Type	Width	Description	Reset Value
0x00	R/W	32	SDMA system address register/argument 2	0000_0000
0x04	R/W	16	Block size register	0000
0x06	R/W	16	Block count register	0000
0x08	R/W	32	Argument 1 register	0000_0000
0x0C	R/W	16	Transfer mode register	0000
0x0E	R/W	16	Command register	0000
0x10	RO	32	Response 0 register	0000_0000
0x14	RO	32	Response 1 register	0000_0000
0x18	RO	32	Response 2 register	0000_0000
0x1C	RO	32	Response 3 register	0000_0000
0x20	R/W	32	Buffer data port	0000_0000
0x24	RO	32	Present state register	0002_0000
0x28	R/W	8	Host control 1 register	00
0x29	R/W	8	Power control register	00
0x2A	R/W or RWAC	8	Block gap control register	00
0x2C	R/W	16	Clock control register	0000

Offset	Type	Width	Description	Reset Value
0x2E	R/W	8	Timeout control register	0E
0x2F	RWAC	8	Software reset register	00
0x30	R/W or RW1C	16	Normal interrupt status register	0000
0x32	RW1C	16	Error interrupt status register	0000
0x34	R/W	16	Normal interrupt status enable register	0000
0x36	R/W	16	Error interrupt status enable register	0000
0x38	R/W	16	Normal interrupt signal enable register	0000
0x3A	R/W	16	Error interrupt signal enable register	0000
0x3C	RO	16	Auto CMD12 error status register	0000
0x3E	R/W	16	Host control 2 register	0000
0x40	HwInit	32	Capabilities register	Revision specific
0x44	HwInit	32	Capabilities register	Revision specific
0x48	HwInit	32	Maximum current capabilities register	Revision specific
0x4C	HwInit	32	Maximum current capabilities register	Revision specific
0x50	WO	16	Force event for Auto CMD12 error status	-
0x52	WO	16	Force event for error interrupt status	-
0x54	RO	8	ADMA error status register	00
0x58	R/W	32	ADMA system address register (Low)	0000_0000
0x5C	R/W	32	ADMA system address register (High)	0000_0000
0x60 ~ 0x6F	RO	32	Preset value	Configurable
0xFE	HwInit	8	Specification version number register	Revision specific
0x100	R/W	32	Vendor-defined register 0	0x0000_0001
0x104	R/W	32	Vendor-defined register 1	0x0000_0000
0x108	R/W	32	Vendor-defined register 2	0x0000_0000
0x10C	R/W	32	Vendor-defined register 3	0x1F00_0808
0x110	RO	32	Vendor-defined register 4	0x0000_0000
0x114	RO	32	Vendor-defined register 5	0x0000_0000
0x118	RO	32	Vendor-defined register 6	0x0000_0001
0x11C	RO	32	Vendor-defined register 7	0x0000_0000
0x120	RO	32	Vendor-defined register 8	0x0000_0000
0x124	RO	32	Vendor-defined register 9	0x0000_0000
0x128	R/W	32	DMA Handshake Enable Register	0x0000_0000
0x178	RO	32	Hardware Attributes Register	Configuration specific
0x17C	RO	32	IP Revision Register	Revision specific

12.3.3.2 Register Description

The following subsections describe the SD host controller registers in details.

12.3.3.2.1 SDMA System Address Register (Offset = 0x00)

Table 12-47 lists the bit assignments of the SDMA system address register. This register is used to set the system memory address of an SDMA transfer or the second argument for Auto CMD23.

For the SDMA transfer

The host driver sets the register before issuing a command to start the SDMA transfer. After the SDMA buffer boundary reaches and stops an SDMA transfer, the next system address of the next contiguous data address can be read from this register. The SDMA transfer waits at every boundary specified by **sdma_buf_bound** in the block size register. When the host driver sets the next system address to this register, the host controller will start a new SDMA transfer.

For the Auto CMD23 argument

This register sets a 32bit block count to the CMD23 argument while executing Auto CMD23. If Auto CMD23 is used with ADMA, the 32bit full block will be used. If Auto CMD23 is used without ADMA, the available block count will be limited by **blk_cnt_r**. In this case, 65535 blocks will be the maximum value.

Table 12-47. SDMA System Address Register (Offset = 0x00)

Bit	Name	Type	Description
[31:0]	sdma_sys_addr_r	R/W	SDMA system address register or Auto CMD23 argument 2

12.3.3.2.2 Block Size Register (Offset = 0x04)

Table 12-48 lists the bit assignments of the block size register. This register is used to configure the number of bytes in a data block. The SDMA system address register is updated at each system memory boundary during the SDMA transfer. When the SDMA transfer reaches each boundary, the host controller will trigger the **dma_interrupt_r** interrupt to request the host driver to update the SDMA system address.

Table 12-48. Block Size Register (Offset = 0x04)

Bit	Name	Type	Description
15	-	-	Reserved
[14:12]	sdma_buf_bound	R/W	Buffer boundary of the SDMA host 3'b111: 512Kbytes 3'b110: 256Kbytes 3'b101: 128Kbytes 3'b100: 64Kbytes 3'b011: 32Kbytes 3'b010: 16Kbytes 3'b001: 8Kbytes 3'b000: 4Kbytes
[11:0]	blk_size_r	R/W	This field specifies the block size of a data transfer for CMD17, CMD18, CMD24, CMD25, and CMD53, and can be set with values ranging from 1 to the maximum buffer size. In the memory, this field should be set to 512bytes. It can be accessed only when no transaction is executed (After a transaction has stopped). The read operation during the transfer may return an invalid value and the write operation should be ignored. 12'h0800: 2048bytes ..... 12'h0200: 512bytes 12'h01FF: 511bytes …….. 12'h0002: 2bytes 12'h0001: 1byte 12'h0000: No transfer

12.3.3.2.3 Block Count Register (Offset = 0x06)

Table 12-49 lists the bit assignments of the current block count. The block count register will be set when the **blk_cnt_en** register is set to '1'. This register is used only for the multi-block transfers. The host controller will decrease the counting number during the data transfer and stop counting when it counts down to zero. When a suspend command is completed in the SDIO transfer, the remaining block counts can be determined by reading this register. Before issuing a resume command to start a re-transfer, the host driver should restore the block counts that are previously saved.

Table 12-49. Block Count Register (Offset = 0x06)

Bit	Name	Type	Description
[15:0]	blk_cnt_r	R/W	Block count of the current transfer 16'hFFFF: 65535 blocks … 16'h0002: 2 blocks 16'h0001: 1 block 16'h0000: Stop counting

12.3.3.2.4 Argument 1 Register (Offset = 0x08)

Table 12-50 lists the bit assignments of the SD command argument. This register is assigned to bits[39:8] of the command field.

Table 12-50. Argument 1 Register (Offset = 0x08)

Bit	Name	Type	Description
[31:0]	arg1_r	R/W	Command argument

12.3.3.2.5 Transfer Mode Register (Offset = 0x0C)

Table 12-51 lists the bit assignments of the transfer mode register. The host driver should set this register before issuing the data transfer command or resume command. In a SDIO transfer, the values of this register should be reserved after the suspend command and should be restored before the resume command. Table 12-52 lists the transfer command settings.

Table 12-51. Transfer Mode Register (Offset = 0x0C)

Bit	Name	Type	Description
[15:6]	-	-	Reserved
5	multi_blk_rw	R/W	Single/Multi-block selection 1: Multiple blocks 0: Single block
4	tran_dir_sel	R/W	Data transfer direction selection 1: Read from the card to host 0: Write from the host to card
[3:2]	auto_cmd_en	R/W	Auto CMD Enable There are two methods to stop the read and write operations of multiple blocks: (1) Auto CMD12 Enable When this field is set to '01', the host controller will automatically issue CMD12 when the last block transfer is completed. The Auto CMD12 error is indicated in the Auto CMD Error Status register. These bits should not be set by the host driver if CMD12 is not required by the command. (2) Auto CMD23 Enable When this field is set to '10', the host controller will automatically issue a CMD23 before issuing a command specified in the Command Register. Auto CMD23 can be used with or without ADMA. By writing the Command Register, the host controller will issue CMD23 first and then issue a command specified by the Command Index in the Command Register. If the response errors of CMD23 are detected, the second command will not be issued. A CMD23 error is indicated in the Auto CMD Error Status register. CMD23 with a block count value of 32bits will be set to the SDMA System Address register. 2'b11: Reserved 2'b10: Auto CMD23 Enable 2'b01: Auto CMD12 Enable 2'b00: Auto Command Disable

Bit	Name	Type	Description
1	blk_cnt_en	R/W	Block count enable This bit is only valid for a multi-block transfer. When this bit is set to '0', the blk_cnt_r register will be disabled. The multi-block transfer will be an infinite transfer.
0	dma_en	R/W	DMA enable This bit can enable the DMA transfer. The DMA mode can be selected by dma_type in the host control register. When a data transfer command is issued, the DMA transfer will begin the operation. 1: DMA data transfer 0: No data transfer or Non-DMA data transfer

Table 12-52. Transfer Command Settings

Single/Multi-block Select	Block Count Enable	Block Count	Transfer Type
0	Don't care	Don't care	Single transfer
1	0	Don't care	Infinite block transfer
1	1	Non-zero	Multi-block transfer
1	1	0	Stop multi-block transfer

12.3.3.2.6 Command Register (Offset = 0x0E)

Table 12-53 lists the bit assignments of the command register. The host driver should check the **Command Inhibit (CMD)** and **Command Inhibit (DAT)** bits in the present state register to determine whether the SD bus is free for a transfer.

Table 12-53. Command Register (Offset = 0x0E)

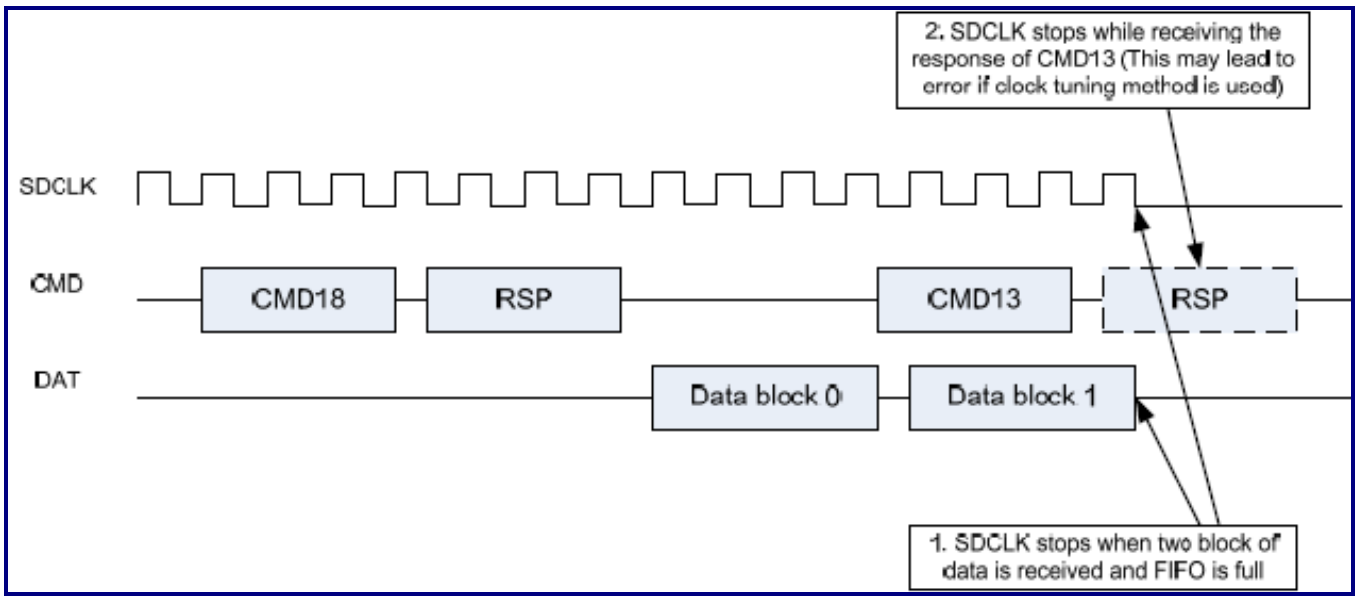
Bit	Name	Type	Description
[15:14]	-	-	Reserved
[13:8]	cmd_idx	R/W	Command Index These bits should be assigned to bits[45:40] of the command field.
[7:6]	cmd_type	R/W	Command Type 2'b11: Abort, CMD12 and CMD52 for writing "I/O Abort" in CCCR 2'b10: Reserved, Reserved for SDIO resume command 2'b01: Reserved, Reserved for SDIO suspend command 2'b00: Normal, other commands

Bit	Name	Type	Description
5	data_pres_sel	R/W	Data Present Select There are two types of special commands: Normal and Abort. Abort Command If this command is set when executing a read transfer, the Host Controller should stop reading to the buffer. If this command is set when executing a write transfer, the Host Controller should stop driving the ***DAT*** line. After issuing the Abort command, the Host Driver should issue a software reset. These bits should be set to '00b' for all other commands. This bit is set to '1' to indicate that data is presented and data transfer is enabled. This bit is set to '0' under the following conditions: (1) Commands only use the CMD line (ex. CMD52). (2) Commands with no data transfer but using the busy signal on DAT[0] line (R1b or R5b ex. CMD38) (3) Resume command
4	cmd_idx_chk_en	R/W	Command Index Check Enable If this bit is set to '1', the host controller will check the response of the index field to determine if the values in cmd_idx are the same. If they are not the same, cmd_idx_err will be triggered.
3	cmd_crc_chk_en	R/W	Command CRC Check Enable If this bit is set to '1', the host controller will check the CRC field response to determine whether CRC is correct. If an error is detected, cmd_crc_err will be triggered.
2	-	-	Reserved
[1:0]	rsp_type_sel	R/W	Response Type Select 2'b11: Response length of 48 with busy check after response 2'b10: Response length of 48 2'b01: Response length of 136 2'b00: No response

Table 12-54. Relationship between Parameters and Names of Response Types

Response Type	Command Index Check Enable	Command CRC Check Enable	Response
2'b00	0	0	No response
2'b01	0	1	R2
2'b10	0	0	R3, R4
2'b10	1	1	R1, R5, R6, R7
2'b11	1	1	R1b, R5b

When using the clock tuning method (In the SDR104/SDR50/DDR50 mode), it is forbidden to issue any command (Except CMD12) before a data transfer command is finished. This is because that if SDCLK stops during receiving a response, errors may occur in this response once the clock is resumed. Please refer to the figure below.



12.3.3.2.7 Response Registers 0 ~ 3 (Offset = 0x10 ~ 0x1C)

Table 12-55 lists the Response Registers. Table 12-56 lists the command responses of each response type. The host controller stores the Auto CMD12 response in the upper word of the Response Registers to avoid the Auto CMD12 response to be overwritten by other commands.

Table 12-55. Response Registers 0 ~ 3 (Offset = 0x10 ~ 0x1C)

Bit	Name	Type	Description
[127:0]	rsp_r	R	Command Response

Table 12-56. Response Registers of Each Response Type

Response Type	Meaning of Response	Response Field	Response Register
R1, R1b (Normal response)	Card status	R[39:8]	rsp_0_r (rsp_r[31:0])
R1b (Auto CMD12 response)	Card status for Auto CMD12	R[39:8]	rsp_3_r (rsp_r[127:96])
R1 (Auto CMD23 response)	Card status for Auto CMD23	R[39:8]	rsp_3_r (rsp_r[127:96])
R2 (CID, CSD register)	CID or CSD register	R[127:8]	rsp_0_r ~ rsp_3_r (rsp_r[119:0])
R3 (OCR register)	OCR register for memory	R[39:8]	rsp_0_r (rsp_r[31:0])
R4 (OCR register)	OCR register for I/O	R[39:8]	rsp_0_r (rsp_r[31:0])
R5, R5b	SDIO response	R[39:8]	rsp_0_r (rsp_r[31:0])
R6 (RCA response)	RCA[31:16]	R[39:8]	rsp_0_r (rsp_r[31:0])

12.3.3.2.8 Buffer Data Port Register (Offset = 0x20)

Table 12-57 lists the buffer data port register. This register uses the 32bit data port register to access the internal buffer (Always accessed through Offset address 0x20 even when the transfer size is BYTE or HALF-WORD.)

Table 12-57. Buffer Data Port Register (Offset = 0x20)

Bit	Name	Type	Description
[31:0]	data_port_r	R/W	Buffer Data Port Register

12.3.3.2.9 Present State Register (Offset = 0x24)

Table 12-58 lists the present status register of the host controller. The host driver can access the status from the 32bit read-only register.

Table 12-58. Present State Register (Offset = 0x24)

Bit	Name	Type	Description
[31:25]	-	-	Reserved
24	cmd_lin_lv	RO	Command Line Signal Level This bit is used to check the CMD line level.
[23:20]	data_lin_lv	RO	DATA[3:0] Line Signal Level This bit is used to check the DATA[3:0] line level
19	wr_prot_lv	RO	Write Protect Pin Level This bit is used to reflect the write protect pin level. 1: Write is enabled. 0: Write is protected.
18	cd_pin_lv	RO	Card Detect Pin Level This bit is used to reflect the inverse value of the card detect pin. It will be stable when sys_card_stable is set to '1'. 1: Card is detected. 0: Card is not detected.
17	sys_card_stable	RO	Card State Stable This bit is used to reflect whether the card detect pin is stable. 1: No card or card is inserted. 0: Reset or de-bounce
16	sys_card_insert	RO	Card Inserted This bit indicates whether a card has been inserted. After de-bounced by the host controller, the host driver can check the insertion of the SD card. The de-bouncing time can be determined by setting bounce_time_sel in the card detect timing control register. This bit can be changed from '0' to '1' to generate a card_insert_r interrupt in the normal interrupt status register. This bit can be changed from '1' to '0' to generate a card_remove_r interrupt in the normal interrupt status register. 1: Card is inserted. 0: Reset, de-bounce, or no card is detected.
[15:12]	-	-	Reserved

Bit	Name	Type	Description
11	buf_ren_r	ROC	Buffer Read Enable This bit is used for the non-DMA read transfer. This bit indicates that there are existing valid data in the RX buffer and is ready to be read. This bit can be changed from '1' to '0' to indicate that data in the RX buffer have been read out. This bit can be changed from '0' to '1' when all block data are ready to be read in the buffer or when the RX buffer is full and generates a buf_r_rdy_r interrupt in the normal interrupt status register. 1: Read is enabled. 0: Read is disabled. Note: The buffer read enable will be set when data equals to the FIFO depth and is ready to be read when data FIFO is implemented by the register (Please check the register offset 0x178 for the hardware attribute). When data FIFO is implemented by SRAM, this bit will be set when at least one block of data is ready to be read.
10	buf_wen_r	ROC	Buffer Write Enable This bit is used for the non-DMA write transfer. This bit indicates that the TX buffer is ready to receive data. If the bit is '1', data will be written to the TX buffer. This bit changes from '1' to '0' when all block data are written to the TX buffer or when the TX buffer is full. This bit changes from '0' to '1' when all block data are written to the TX buffer. 1: Write is enabled. 0: Write is disabled. Note: The buffer write enable will be set when space equals to the FIFO depth and is available for write if data FIFO is implemented by the register (Please check the register offset 0x178 for the hardware attribute). When data FIFO is implemented by SRAM, this bit will be set when at least one block of space is available to be written.
9	rd_tran_act_r	ROC	Read Transfer Active This bit is used to detect the completion of a read transfer. This bit is set to '1' under the following conditions: (1) After the end bit of a read command (2) cont_req in the block gap control register is set to restart a transfer. This bit is set to '0' under the following conditions: (1) All data blocks specified by the block length are transferred to the system. (2) sp_blk_gap_req in the block gap control register is set to '1' and the host controller transfers all valid data blocks to the system. The tran_complete_r interrupt will be generated when this bit changes from '1' to '0'.

Bit	Name	Type	Description
8	wr_tran_act_r	ROC	Write Transfer Active This bit indicates that a write transfer is active. This bit is set to '1' under the following conditions: (1) After the end bit of a write command (2) cont_req in the block gap control register is set to restart a transfer. This bit is set to '0' under the following conditions: (1) Get the CRC status of the last data block specified by the transfer count. (2) Get the CRC status of a block that the data transmission is stopped by sp_blk_gap_req. A blk_gap_evt_r interrupt will be generated when sp_blk_gap_req is set to '1' and this bit changes to '0.' This bit is useful in the command with a busy data line.
[7:3]	-	-	Reserved
2	data_lin_act_r	ROC	Data Line Active This bit is used to determine whether data line is in use. In a read transfer, this bit is used to check the execution of a read transfer on the bus. Changing this bit from '1' to '0' will generate a blk_gap_evt_r interrupt when sp_blk_gap_req is set to '1'. This bit is set to '1' under the following conditions: (1) After the end bit of a read command (2) cont_req in the block gap control register is set to restart a transfer. This bit is set to '0' under the following conditions: (1) The end bit of the last data block is sent from the SD bus to the host controller. (2) sp_blk_gap_req is set to '1' and a read transfer is stopped at the block gap. In a write transfer, this bit is used to check the execution of a write transfer on the bus. Changing this bit from '1' to '0' will generate a tran_complete_r interrupt in the normal interrupt status register. This bit is set to '1' under the following conditions: (1) A read command after the end bit (2) cont_req in the block gap control register is set to restart a transfer. This bit is set to '0' under the following conditions: (1) The card releases the busy signal of the last data block. (2) sp_blk_gap_req is set to '1' and the card releases the write busy at the block gap. In a command with busy data line, this bit indicates the execution of a command with the busy signal on the bus. This bit will be set after the end bit of the command with the busy signal and will be cleared when the busy signal is de-asserted or the busy signal is not detected after the end of a response.

Bit	Name	Type	Description
1	cmd_inhibit_d	ROC	Command Inhibit (DAT) This bit will be generated if data_lin_act_r or rd_tran_act_r is set to '1'. When this bit is set to '0', the host driver can issue a new command by using the data line. Command with the busy data line belongs to cmd_inhibit_d. 1: This bit cannot issue a new command to use the data line. 0: This bit can issue a new command to use the data line.
0	cmd_inhibit_c	ROC	Command Inhibit (CMD) When this bit is set to '0', the host driver will issue a new command by using the command line. When this bit is set to '1', the command register will be written. This bit will be cleared when the command response is received. Even if cmd_inhibit_d is set to '1', the commands using the command line can be issued. If this bit is set to '0', CMD0, CMD12, CMD13, and CMD52 can be issued when the data lines are in use during a data transfer. The cmd_complete_r interrupt in the normal interrupt register will be issued when this bit is changed from 1 to 0. However, if the present command suffers errors (CRC error, command index error, and so on), this bit will remain to '1'. Note: This bit will not be set by Auto CMD12. 1: Cannot issue a command 0: Issue a command only with the command line

12.3.3.2.10 Host Control 1 Register (Offset = 0x28)

Table 12-59 lists the bit assignments of the host control 1 register.

Table 12-59. Host Control 1 Register (Offset = 0x28)

Bit	Name	Type	Description
7	cd_sel	R/W	Card Detect Signal Selection This bit is used to select the source of card detection. 1: Test level for card detection 0: Card detect pin is selected.
6	cd_test_lv	R/W	Card Detect Test Level This bit is used to set the test level when cd_sel is set to '1'. 1: Card is inserted. 0: Card cannot be found.

Bit	Name	Type	Description
5	ext_data_width	R/W	Extended Data Transfer Width This bit controls the bus width supported by the embedded device, which is indicated in the Capabilities Register. When this bit is set to '1', it indicates that the 8bit bus is supported by the device. When this bit is set to '0', it indicates that the bus width is controlled by the Data Transfer Width in the Host Control 1 Register. This bit will not be effective if multiple devices are installed on a bus slot (Slot Type is set to '10b' in the Capabilities Register). In this case, each device bus width will be controlled by the Bus Width Preset field in the Shared Bus Control Register. 1: 8bit bus width 0: Bus width is selected by the data transfer width.
[4:3]	dma_type	R/W	DMA Type Select 2'b11: Reserved 2'b10: 32bit address ADMA2 is selected. 2'b01: Reserved 2'b00: SDMA is selected.
2	hi_speed	R/W	High Speed Enable If this bit is set to '0', the host controller will output command and data at the falling edge of io_sd_clk at a speed up to 25MHz. If this bit is set to '1', the host controller will output command and data at the rising edge of io_sd_clk at a speed up to 50MHz.
1	data_width	R/W	Data Width This bit can select the data width of the host controller. The data width should match the SD card bus width during a data transfer. 1: 4bit data width 0: 1bit data width
0	led_ctrl	R/W	LED Control This bit is used to notify users that the SD card has been accessed. 1: LED on 0: LED off

12.3.3.2.11 Power Control Register (Offset = 0x29)

Table 12-60 lists the bit assignments of the power control register.

Table 12-60. Power Control Register (Offset = 0x29)

Bit	Name	Type	Description
[7:4]	-	-	Reserved
[3:1]	sd_bus_vol	R/W	SD Bus Voltage Select By setting these bits, the host driver will select the voltage level for the SD card. Before setting this field, the host driver should check the Voltage Support bits in the Capabilities Register. If an unsupported voltage is selected, the host system should not supply the SD bus voltage. 3'b111: 3.3V (Typ.) 3'b110: 3.0V (Typ.) 3'b101: 1.8V (Typ.) Others: Reserved
0	sd_bus_pow	R/W	SD Bus Power This bit is used to enable the SD bus power. If no card is detected by the host controller, this bit will be set to '0'. When this bit is set to '0', io_sd_clk will be driven to '0'.

12.3.3.2.12 Block Gap Control Register (Offset = 0x2A)

Table 12-61 lists the bit assignments of the block gap control register.

Table 12-61. Block Gap Control Register (Offset = 0x2A)

Bit	Name	Type	Description
[7:4]	-	-	Reserved
3	Int_at_blk_gap	R/W	Interrupt at Block Gap This bit is only useful in the 4bit mode of the SDIO card. Setting this bit to '1' will enable the interrupt detection at the block gap for multiple block transfers. This bit should be set to '0' if the SDIO card cannot issue an interrupt during the block gap. 1: Enable the check of the interrupt at block gap 0: Disable the check of the interrupt at block gap

Bit	Name	Type	Description
2	read_wait	R/W	Read Wait Control This bit is useful for the SDIO card. When the SDIO card supports the read wait function, the host controller will use the read wait protocol to control the SDIO bus. When the card does not support the read wait function, the host controller will stop io_sd_clk to hold a read transfer. Suspend/Resume cannot be used if this bit is set to '0'. 1: Enable the read wait function 0: Disable the read wait function
1	cont_req	RWAC	Continue Request This bit is used to restart a transaction, which can be stopped by using sp_blk_gap_req. To restart a transaction, this bit should be set to '1' and sp_blk_gap_req should be set to '0'. The host controller will automatically clear this bit under the following conditions: (1) In a read transfer, data_lin_act_r will be changed from 0 to 1 to start a read transfer. (2) In a write transfer, wr_tran_act_r will be changed from 0 to 1 to start a write transfer. 1: Restart 0: No effect
0	sp_blk_gap_req	RW	Stop at Block Gap Request This bit is used to stop executing the read or write transfer at the next block gap. This bit is useful for the non-DMA and SDMA, but is not useful for ADMA. The host driver should keep the setting of this bit to '1' until the tran_complete_r interrupt is set to '1'. If this bit is set to '1', the host controller will stop at the block gap by using read_wait or stop io_sd_clk in a read transaction. If this bit is set to '0', the host controller will not write data to data_port_r. 1: Stop 0: Transfer

12.3.3.2.13 Clock Control Register (Offset = 0x2C)

Table 12-62 lists the bit assignments of the clock control register.

Table 12-62. Clock Control Register (Offset = 0x2C)

Bit	Name	Type	Description
[15:8]	low_bit_sd_clk_sel	R/W	SD clock frequency [7:0] for the 10bit divided clock mode This bit is used to select the frequency of the io_sd_clk pin. The base clock is specified by sdc_clk (Base clock = sdclk1x). 10'h3FF Base clock divided by 2046 … … N Base clock divided by 2N (Duty 50%) … … 10'h002 Base clock divided by 4 10'h001 Base clock divided by 2 10'h000 Base clock (sdclk1x)
[7:6]	Upper_bit__sd_clk_sel	R/W	SD clock frequency [9:8] for the 10bit divided clock mode
5	clk_gen_sel	ROC	This bit is always set to zero. 1: Programmable clock mode 0: 10bit divided clock mode
[4:3]	-	R/W	Reserved
2	sd_clk_en	R/W	SD Clock Enable When this bit is set to '1', io_sd_clk will be outputted to the SD card. When this bit is set to '0', io_sd_clk will be stopped at '0'.
1	clk_stable	ROC	Internal Clock Stable This bit will be set to '1' when the internal clock is stable. After inter_clk_en is set to '1', the internal clock will begin oscillating.

Bit	Name	Type	Description
0	Inter_clk_en	R/W	Internal Clock Enable When the host controller is not used by the host driver, this bit can be set to '0' to stop the internal clock at the low-power state. When this bit is set to '1', the internal clock will begin oscillating.

12.3.3.2.14 Timeout Control Register (Offset = 0x2E)

Table 12-63 lists the bit assignments of the timeout control register. The host driver should set the timeout value according to the **cap_r** register. The value of data_timer indicates the data line timeout times.

Table 12-63. Timeout Control Register (Offset = 0x2E)

Bit	Name	Type	Description
[7:4]	-	-	Reserved
[3:0]	data_timer	R/W	Data Timeout Counter Value 4'b1111 Reserved 4'b1110 sys_hclk x 2^{27} 4'b1101 sys_hclk x 2^{26} …. …. 4'b0000 sys_hclk x 2^{13}

12.3.3.2.15 Software Reset Register (Offset = 0x2F)

Table 12-64 lists the bit assignments of the software reset register. A reset pulse will be generated when this bit is set to '1'. This bit will be automatically cleared when the reset pulse is issued.

Table 12-64. Software Reset Register (Offset = 0x2F)

Bit	Name	Type	Description
[7:3]	-	-	Reserved

Bit	Name	Type	Description
2	soft_rst_dat	RWAC	Software Reset for Data Line The following registers will be cleared by resetting this bit: (1) Buffer Data Port Register data_port_r (2) Present State Registers buf_ren_r, buf_wen_r, rd_tran_act_r, wr_tran_act_r, data_lin_act_r, and cmd_inhibit_d (3) Block Gap Control Registers cont_req and sp_blk_gap_req (4) Normal Interrupt Status Registers buf_r_rdy_r, buf_w_rdy_r, dma_interrupt_r, blk_gap_evt_r, and tran_complete_r
1	soft_rst_cmd	RWAC	Software Reset for Command Line The following registers will be cleared by this reset bit: (1) Present State Register cmd_inhibit_c (2) Normal Interrupt Status Register cmd_complete_r
0	soft_rst_all	RWAC	Software Reset for All Software resets all registers, except for the card detection and capability register.

12.3.3.2.16 Normal Interrupt Status Register (Offset = 0x30)

Table 12-65 lists the bit assignments of the normal interrupt status register. The interrupt status can be latched by setting the **nor_int_st_en_r** registers to '1'.

Table 12-65. Normal Interrupt Status Register (Offset = 0x30)

Bit	Name	Type	Description
15	err_interrupt_r	ROC	Error Interrupt This bit will be set to '1' if any bit in the err_int_status_r register is set to '1'.
[14:13]	-	-	Reserved

Bit	Name	Type	Description
12	re_tuning_int_r	ROC	Re-Tuning Event This bit will be set if the re-tuning request in the Present State Register changes from ‘0’ to ‘1’. The host controller requests the host driver to perform re-tuning for the next data transfer. The current data transfer (Not the large block count) can be completed without re-tuning. 1: Re-tuning is required. 0: Re-tuning is not required.
11	Int_C_r	ROC	INT_C This bit will be set if INT_C is enabled and the INT_C# pin is at low level. Writing this bit to ‘1’ will not clear this bit. This bit can be cleared by setting the INT_C interrupt factor. Please refer to Shared Bus Control Register for details.
10	Int_B_r	ROC	INT_B This bit will be set if INT_B is enabled and the INT_B# pin is at low level. Writing this bit to ‘1’ will not clear this bit. This bit can be cleared by setting the INT_B interrupt factor. Please refer to Shared Bus Control Register for details.
9	Int_A_r	ROC	INT_A This status will be set if INT_A is enabled and INT_A# pin is at low level. Writing this bit to ‘1’ will not clear this bit. This bit can be cleared by setting the INT_A interrupt factor. Please refer to Shared Bus Control Register for details.
8	card_int_r	ROC	Card Interrupt In the 4bit mode, the card interrupt is sampled during the interrupt cycle. When this bit is set to ‘1’, the host driver will handle the interrupt. The host driver needs to set card_int_st_en to ‘0’ and clear the card_int_r status to avoid driving the interrupt signal to the host system again. If the host driver has completely handled the card interrupt. card_int_st_en, should be set to ‘1’ to restart sampling.
7	card_remove_r	RW1C	Card Remove This bit will be set if sys_card_inst in the Present State Register changes form ‘1’ to ‘0’. When the host driver writes ‘1’ to this bit, this status will be cleared.
6	card_insert_r	RW1C	Card Insert This bit will be set if sys_card_inst in the Present State Register changes from ‘0’ to ‘1’. When the host driver writes ‘1’ to this bit, this status will be cleared.
5	buf_r_rdy_r	RW1C	Buffer Read Ready This bit will be set if buf_ren_r in the Present State Register changes from ‘0’ to ‘1’.

Bit	Name	Type	Description
4	buf_w_rdy_r	RW1C	Buffer Write Ready This bit will be set if buf_wen_r in the Present State Register changes from '0' to '1'.
3	dma_interrupt_r	RW1C	DMA Interrupt This bit will be set if the host controller detects that the SDMA buffer boundary is reached during a transfer. In case of ADMA, the "int" field will be set in the descriptor line and the host controller will generate the interrupt when the descriptor line is done.
2	blk_gap_evt_r	RW1C	Block Gap Event By setting sp_blk_gap_req to '1', this bit will be set when a read or write transaction is stopped at the block gap. (1) Read Transaction This bit will be set when data_lin_act_r changes from '1' to '0'. The read wait should be supported for this function in the SDIO card. (2) Write Transaction This bit will be set when wr_tran_act_r changes from '1' to '0'.
1	tran_complete_r	RW1C	Transfer Complete A data transfer will be completed or a data transfer will be stopped at the block gap during a transmission to generate a transfer complete interrupt. (1) Read Transaction This bit will be set when rd_tran_act_r changes from 1 to 0. (2) Write Transaction This bit will be set when data_lin_act_r changes from 1 to 0. (3) Command with busy This bit will be set when the busy data line is released.
0	cmd_complete_r	RW1C	Command Complete This bit will be set when the command response is received and the CRC check is okay. Auto CMD12 and Auto CMD23 will not generate the command complete interrupt. The following table shows that the Command Timeout Error bit has higher priority than the Command Complete bit. When both bits are set to '1', it indicates that the response has not been correctly received.

Command Complete	Command Timeout Error	Status
0	0	Interrupted by another factor
don't care	1	Response not received within 64 SCLK cycles
1	0	Response received

12.3.3.2.17 Error Interrupt Status Register (Offset = 0x32)

Table 12-66 lists the bit assignments of the error interrupt status register. The interrupt status can be latched when the **err_int_st_en_r** register is set to '1'.

Table 12-66. Error Interrupt Status Register (Offset = 0x32)

Bit	Name	Type	Description
[15:11]	-	-	Reserved
10	Tuning_err_r	RW1C	Tuning Error This bit will be set when an unrecoverable error is detected in a tuning circuit, except during the tuning procedure (Occurrence of an error during the tuning procedure is indicated by the Sampling Select). By detecting the tuning error, the host driver needs to abort a command executing and perform tuning. To reset the tuning circuit, the sampling clock should be set to '0' before restarting the tuning procedure. The tuning error has higher priority than other error interrupts generated during a data transfer. By detecting the tuning error, the host driver should discard the data transfer by a current read/write command and retry the data transfer after the host controller is retrieved from the tuning circuit error.
9	adma_err_r	RW1C	ADMA Error This bit will be set to '1' when an error is detected in the ADMA error status register during an ADMA transfer. In addition, the host controller will set this interrupt when it detects the invalid descriptor data (Valid = 0).
8	auto_cmd12_err_r	RW1C	Auto CMD12 Error This bit will be set to '1' when an error is detected in the Auto CMD12 error status register.
7	cur_lim_err_r	RW1C	Current Limit Error By setting the SD Bus Power bit in the Power Control Register, the host controller will 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 supplying power to the card. In this case, this bit will indicate a failure status. Reading '1' from this bit means that the host controller is not supplying power to the SD card due to some failures. Reading '0' from 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. This bit should always be set to '0' if the host controller does not support this function.
6	data_end_bit_err_r	RW1C	Data End Bit Error This bit will be set when the host controller detects '0' at the end bit of the read data, which uses the data line or at the end bit of the write CRC status.
5	data_crc_err_r	RW1C	Data CRC Error This bit will be set when the host controller detects the CRC error during the read transfer or detects the write CRC status that is not the value of "010".

Bit	Name	Type	Description
4	data_timeout_err_r	RW1C	Data Timeout Error This bit will be set under the following conditions: (1) Wait to read data timeout (2) Wait to write CRC status timeout (3) Busy timeout after write CRC status (4) Busy timeout for R1b, R5b types
3	cmd_idx_err_r	RW1C	Command Index Error This bit will be set when the command index in the command response is different from the command.
2	cmd_end_bit_err_r	RW1C	Command End Bit Error This bit will be set when the end bit of the command response is 0.
1	cmd_crc_err_r	RW1C	Command CRC Error This bit will be set when the command CRC error is detected.
0	cmd_timeout_err_r	RW1C	Command Timeout Error This bit will be set when no response is sent from the card within 64 SD card clock.

12.3.3.2.18 Normal Interrupt Status Enable Register (Offset = 0x34)

Table 12-67 lists the bit assignment of the normal interrupt status enable register. If the corresponding bit of an interrupt source in the normal interrupt status enable register is set to '1' and the interrupt becomes active, which is latched and available for the host driver in the normal interrupt status register.

Table 12-67. Normal Interrupt Status Enable Register (Offset = 0x34)

Bit	Name	Type	Description
15	-	R	Reserved and fixed to 0 Error interrupts are controlled by the error interrupt status enable register.
[14:13]	-	-	Reserved
12	re_tuning_st_en	RW	Re-Tuning Event Status Enable
11	Int_C_st_en	R/W	INT_C Status Enable If this bit is set to '0', the Host Controller will clear the interrupt request to the system. The Host Driver may clear this bit before serving INT_C and may reset this bit after all interrupt requests to the INT_C pin are cleared to prevent the inadvertent interrupts.

Bit	Name	Type	Description
10	Int_B_st_en	R/W	INT_B Status Enable If this bit is set to '0', the Host Controller will clear the interrupt request to the system. The Host Driver may clear this bit before serving INT_B and may reset this bit after all interrupt requests to the INT_B pin are cleared to prevent the inadvertent interrupts.
9	Int_A_st_en	R/W	INT_A Status Enable If this bit is set to '0', the Host Controller will clear the interrupt request to the system. The Host Driver may clear this bit before serving INT_A and may reset this bit after all interrupt requests to the INT_A pin are cleared to prevent the inadvertent interrupts.
8	card_int_st_en	R/W	Card Interrupt Status Enable If this bit is set to '1', card_int_r will be served by the Host Driver. The Host Driver should set this bit to '0' before serving card_int_r and should reset this bit after card_int_r completes requesting card. The detection of card_int_r will be stopped when this bit is set to '0' and will be restarted when this bit is set to '1'.
7	card_remove_st_en	R/W	Card Remove Status Enable
6	card_insert_st_en	R/W	Card Insert Status Enable
5	buf_r_rdy_st_en	R/W	Buffer Read Ready Status Enable
4	buf_w_rdy_st_en	R/W	Buffer Write Ready Status Enable
3	dma_interrupt_st_en	R/W	DMA Interrupt Status Enable
2	blk_gap_evt_st_en	R/W	Block Gap Event Status Enable
1	tran_complete_st_en	R/W	Transfer Complete Status Enable
0	cmd_complete_st_en	R/W	Command Complete Status Enable

12.3.3.2.19 Error Interrupt Status Enable Register (Offset = 0x36)

Table 12-68 lists the bit assignment of the error interrupt status enable register. If the corresponding bit of the interrupt source in the Error Interrupt Status Enable Register is set to '1' and if the interrupt becomes active, this is latched and available for the host driver in this register.

Table 12-68. Error Interrupt Status Enable Register (Offset = 0x36)

Bit	Name	Type	Description
[15:11]	-	-	Reserved
10	Tuning_err_st_en	R/W	Tuning Error Status Enable
9	adma_err_st_en	R/W	ADMA Error Status Enable
8	auto_cmd12_err_st_en	R/W	Auto CMD12 Error Status Enable

Bit	Name	Type	Description
7	cur_lim_err_st_en	R/W	Current Limit Error Status Enable
6	data_end_bit_err_st_en	R/W	Data End Bit Error Status Enable
5	data_crc_err_st_en	R/W	Data CRC Error Status Enable
4	data_timeout_err_st_en	R/W	Data Timeout Error Status Enable
3	cmd_idx_err_st_em	R/W	Command Index Error Status Enable
2	cmd_end_bit_err_st_en	R/W	Command End Bit Error Status Enable
1	cmd_crc_err_st_en	R/W	Command CRC Error Status Enable
0	cmd_timeout_err_st_en	R/W	Command Timeout Error Status Enable

12.3.3.2.20 Normal Interrupt Signal Enable Register (Offset = 0x38)

Table 12-69 lists the bit assignment of the normal interrupt signal enable register. This register is used to select the interrupt status that is notified to the host system as an interrupt. These interrupt statuses share the same interrupt line.

Table 12-69. Normal Interrupt Signal Enable Register (Offset = 0x38)

Bit	Name	Type	Description
15	-	R	Reserved and fixed to '0' The error interrupts are controlled by the error interrupt signal enable register.
[14:13]	-	-	Reserved
12	re_tuning_sig_en	R/W	Re-Tuning Event Signal Enable
11	Int_C_sig_en	R/W	INT_C Signal Enable
10	Int_B_sig_en	R/W	INT_B Signal Enable
11	Int_A_sig_en	R/W	INT_A Signal Enable
8	card_int_sig_en	R/W	Card Interrupt Signal Enable
7	card_remove_sig_en	R/W	Card Remove Signal Enable
6	card_insert_sig_en	R/W	Card Insert Signal Enable
5	buf_r_rdy_sig_en	R/W	Buffer Read Ready Signal Enable
4	buf_w_rdy_sig_en	R/W	Buffer Write Ready Signal Enable
3	dma_interrupt_sig_en	R/W	DMA Interrupt Signal Enable
2	blk_gap_evt_sig_en	R/W	Block Gap Event Signal Enable
1	tran_complete_sig_en	R/W	Transfer Complete Signal Enable
0	cmd_complete_sig_en	R/W	Command Complete Signal Enable

12.3.3.2.21 Error Interrupt Signal Enable Register (Offset = 0x3A)

Table 12-70 lists the bit assignment of the error interrupt signal enable register. This register is used to select the interrupt status that is regarded by the host system as an interrupt. These interrupt statuses share the same interrupt line.

Table 12-70. Error Interrupt Signal Enable Register (Offset = 0x3A)

Bit	Name	Type	Description
[15:11]	-	-	Reserved
10	tuning_err_sign_en	R/W	Tuning Error Signal Enable
9	adma_err_sig_en	R/W	ADMA Error Signal Enable
8	auto_cmd12_err_sig_en	R/W	Auto CMD12 Error Signal Enable
7	cur_lim_err_sign_en	R/W	Current Limit Error Signal Enable
6	data_end_bit_err_sig_en	R/W	Data End Bit Error Signal Enable
5	data_crc_err_sig_en	R/W	Data CRC Error Signal Enable
4	data_timeout_err_sig_en	R/W	Data Timeout Error Signal Enable
3	cmd_idx_err_sig_em	R/W	Command Index Error Signal Enable
2	cmd_end_bit_err_sig_en	R/W	Command End Bit Error Signal Enable
1	cmd_crc_err_sig_en	R/W	Command CRC Error Signal Enable
0	cmd_timeout_err_sig_en	R/W	Command Timeout Error Signal Enable

12.3.3.2.22 Auto CMD12 Error Status Register (Offset = 0x3C)

Table 12-71 lists the status of the Auto CMD12 error status register. When the auto_cmd12_en register is set to '1' and the auto cmd12 error status register is set, the host driver will check this register to identify what kind of error happens during executing the auto cmd12. This register is valid only when the **auto_cmd12_err_r** is set to '1'.

Table 12-71. Auto CMD12 Error Status Register (Offset = 0x3C)

Bit	Name	Type	Description
[15:8]	-	-	Reserved

Bit	Name	Type	Description
7	cmd_no_ex_by_cmd12_r	ROC	Command Not Executed by Auto CMD12 Error This bit error indicates the command that follows the Auto CMD12 which is not executed due to an Auto CMD12 error (Bit 1 to bit 4) in this register. The bit is set to '0' when Auto CMD error is generated by Auto CMD23.
[6:5]	-	-	Reserved
4	auto_cmd_idx_err_r	ROC	Auto CMD Index Error
3	auto_cmd_end_bit_err_r	ROC	Auto CMD End Bit Error
2	auto_cmd_crc_err_r	ROC	Auto CMD CRC Error
1	auto_cmd_timeout_err_r	ROC	Auto CMD Timeout Error
0	auto_cmd12_no_ex_r	ROC	Auto CMD12 Not Executed If the memory multiple block data transfer is not started due to the command error, this bit will not be set because it is not necessary to issue Auto CMD12. Setting this bit to '1' means that the Host Controller cannot issue Auto CMD12 to stop the memory multiple block data transfer due to some errors. If this bit is set to '1', other error status bits (D04-D01) are meaningless. This bit is set to '0' when Auto CMD error is generated by Auto CMD23.

The relationship between Auto CMD CRC error and Auto CMD timeout error is shown in Table 12-72.

Table 12-72. Relationship between CRC Error and Timeout Error for Auto CMD

Auto CMDCRC Error	Auto CMDTimeout Error	Error Type
0	0	No Error
0	1	Response Timeout Error
1	0	Response CRC Error
1	1	CMD line conflict

12.3.3.2.23 Host Control 2 Register (Offset = 0x3E)

Table 12-73 lists the bit assignment of the Host Control 2 Register.

Table 12-73. Host Control 2 Register (Offset = 0x3E)

Bit	Name	Type	Description
15	preset_val_en	R/W	Preset Value Enable Since the operating SDCLK frequency and I/O driving strength are dependent on the Host System implementation, it is difficult to determine these parameters in the Standard Host Driver. When this bit is set, the SDCLK frequency will be automatically generated and the driver strength will be selected without considering the specific system conditions. This bit enables the functions defined in the Preset Value registers. 1: Automatic Selection by Preset Value are enabled. 0: SDCLK and Driver Strength are controlled by the host driver.
14	asyn_int_en	R/W	Asynchronous Interrupt Enable This bit can be set to '1' if a card supports the asynchronous interrupts and the Asynchronous Interrupt Support is set to '1' in the Capabilities register. If this bit is set to '1', the Host Driver will stop SDCLK during the asynchronous interrupt period to save power. During this period, the Host Controller will continue delivering the Card Interrupt to the host when it is asserted by the Card.
[13:8]	-	-	Reserved
7	sample_clk_sel	R/W	Sampling Clock Select This bit is used by the Host Controller to select the sampling clock to receive CMD and DAT. This bit is set by the tuning procedure and is valid after the completion of tuning. Setting this bit to '1' means that the tuning is completed successfully Setting this bit to '0' means that the tuning is failed. 1: Tuned clock is used to sample data. 0: Fixed clock is used to sample data.
6	execute_tuning	RWAC	Execute Tuning This bit is set to '1' to start the tuning procedure and will be automatically cleared when the tuning procedure is completed. The result of tuning is indicated to Sampling Clock Select. The tuning procedure is aborted by writing '0'. 1: Execute Tuning 0: Not Tuned or Tuning Completed

Bit	Name	Type	Description
[5:4]	driver_str_sel	R/W	Driver Strength Select This bit selects the Host Controller output driver in the 1.8V signaling. This field is not effective in the 3.3V signaling. This field can be set depends on the driver type A; type C, and type D Support Bit in the Capabilities register. 11: Driver type D is selected. 10: Driver type C is selected. 01: Driver type A is selected. 00: Driver type B is selected (Default).
3	1V8_sig_en	R/W	1.8V Signaling Enable This bit controls the voltage regulator for the I/O cell. 3.3V is supplied to the card regardless of the signaling voltage. 1: 1.8V signaling 0: 3.3V signaling
[2:0]	UHS_mode_sel	R/W	UHS Mode Select This field is used to select one of the UHS-I modes and is effective when the 1.8V Signaling Enable is set to '1'. 111: Reserved 110: Reserved 101: Reserved 100: DDR50 011: SDR104 010: SDR50 001: SDR25 000: SDR12

12.3.3.2.24 Capabilities Register (Offset = 0x40)

Table 12-74 lists the information of the Capabilities Register. The host controller may implement these values during initialization.

Table 12-74. Capabilities Register (Offset = 0x40)

Bit	Name	Type	Description
[63:56]	-	-	Reserved

Bit	Name	Type	Description
[55:48]	clk_multiplier	RO	Clock Multiplier This field indicates the clock multiplier of the programmable clock generator. The version does not support the feature and fixed to ‘00’.
[47:46]	re_tuning_mode	RO	Re-Tuning Modes This field selects the re-tuning method and limits the maximum data length. Re-Tuning Mode 1 The host controller does not have any internal logic to detect when the re-tuning needs to be performed. In this case, the Host Driver should maintain all the re-tuning timings by using a re-tuning timer. To enable the insertion of the re-tuning procedure during data transfers, the data length per read/write command should be limited up to 4MB. The version supports only mode 1 and is fixed to ‘00’.
45	tuning_SDR50	RO	Use Tuning for SDR50 If this bit is set to ‘1’, this Host controller will require the tuning to operate SDR50. The version does not require tuning to operate SDR50 and is fixed to ‘0
44	-	-	Reserved
[43:40]	timer_cnt_for_re_tuning	RO	Timer Count for Re-Tuning This field indicates an initial value of the re-tuning timer for re-tuning mode 1 to mode 3. The version does not support the feature and is fixed to ‘0000’.
39	-	-	Reserved
38	driver_D_support	HwInit	Driver Type D Support This bit indicates that driver type D is supported for 1.8V signaling.
37	driver_C_support	HwInit	Driver Type C Support This bit indicates that driver type C is supported for 1.8V signaling.
36	driver_A_support	HwInit	Driver Type A Support This bit indicates that driver type A is supported for 1.8V signaling.
35	-	-	Reserved
34	DDR50_support	RO	DDR50 Support The version supports the feature and is fixed to ‘1’.
33	-	RO	Reserved
32	SDR50_support	RO	SDR50 Support The version supports the feature and is fixed to ‘1’.

Bit	Name	Type	Description
[31:30]	Slot_type	HwInit	Slot Type This field indicates the usage of a slot by specific host system (A host controller register set is defined per slot). Embedded slot for one device (01b) indicates that only one non-removable device will be connected to a SD bus slot. Shared Bus Slot (10b) can be set if the Host Controller supports the Shared Bus Control register. The version supports “Removable Card Slot” or “Embedded Slot for One Device”. This field can only be set to 2’b00 or 2’b01.
29	async_int_support	RO	Asynchronous Interrupt Support Please refer to the SDIO Specification, Version 3.00, on the asynchronous interrupt. The version does not support the feature and is fixed to ‘0’.
28	bus_64_support	RO	64bit System Bus Support Setting this bit to ‘1’ indicates that the host controller supports 64bit address descriptor mode and is connected to 64bit address system bus. The version does not support the feature and is fixed to ‘0’.
27	-	-	Reserved
26	voltage_1_8_support	HwInit	Voltage supports 1.8V.
25	voltage_3_0_support	HwInit	Voltage supports 3.0V.
24	voltage_3_3_support	HwInit	Voltage supports 3.3V.
23	suspend_resume_support	RO	Suspend/Resume Support This bit suspends/resumes the SDIO function of the host controller. The version supports the feature and is fixed to ‘1’.
22	sdma_support	RO	SDMA Support This bit indicates whether the host controller can use SDMA to transfer data. The version supports the feature and is fixed to ‘1’.
21	hi_speed_support	RO	High Speed Support This bit indicates that the host controller can support the high-speed mode and can supply io_sd_clk from 25MHz to 50MHz. The version supports the feature and is fixed to ‘1’.
20	adma1_support	RO	ADMA1 Support This bit indicates that the host controller is capable of using legacy ADMA1. The version does not support the feature and is fixed to ‘0’.

Bit	Name	Type	Description
19	adma2_support	RO	ADMA2 Support This bit indicates that the host controller is capable of using ADMA2. The version supports the feature and is fixed to ‘1’.
18	-	RO	Reserved
[17:16]	max_blk_len	RO	Maximum Block Length This field indicates the maximum block size to be read and written by the host driver to the buffer in the host controller. The buffer should transfer this block size without wait cycles. 512bytes are defined
[15:8]	base_clk_for_SD_clk	HWInit	Base Clock Frequency For SD Clock This field indicates the base (Maximum) clock frequency for the SD clock. FFh 255MHz … … 02h 2MHz 01h 1MHz 00h Get information via another method If the real frequency is 16.5MHz, the lager value should be set to 0001 0001b (17MHz) because this value is used by the host driver to calculate the clock divider value. The value should not exceed the upper limit of the SD clock frequency. If these bits are all ‘0’, the host system has to acquire information via another method
7	timeout_clk_unit	RO	Timeout Clock Unit This bit shows the unit of the base clock frequency used to detect data_timeout_err_r. The version supports MHz and is fixed to ‘1’.
6	-	-	Reserved

Bit	Name	Type	Description
[5:0]	timeout_clk_freq	RO	Timeout Clock Frequency This field shows the base clock for detecting data_timeout_err_r. The base clock is dominated by the system clock. Users should obtain the system clock frequency. The version is “Get information via another method” and is fixed to ‘00h’.

12.3.3.2.25 Maximum Current Capabilities Register (Offset = 0x48)

Table 12-75 lists the maximum current capability for each voltage. The value will be meaningful when the Voltage Support is set in the capabilities register. If this register is set to ‘0’, the information will be supplied by the host system via another method. The current unit of this register is in 4mA steps. Table 12-76 lists the current value of the register.

Table 12-75. Maximum Current Capabilities Register (Offset = 0x48)

Bit	Name	Type	Description
[63:24]	-	-	Reserved
[23:16]	max_cur_for_vlg_1_8	HwInit	Maximum current for a voltage of 1.8V Please refer to Table 31-32 for the value setting.
[15:8]	max_cur_for_vlg_3_0	HwInit	Maximum current for a voltage of 3.0V Please refer to Table 31-32 for the value setting.
[7:0]	max_cur_for_vlg_3_3	HwInit	Maximum current for a voltage of 3.3V Please refer to Table 31-32 for the value setting.

This register measures the current in 4mA steps. The current at each voltage level is listed in Table 12-76.

Table 12-76. Definitions of Maximum Current Value

Register Value	Current Value
8’d0	Get information via another method
8’d1	4mA
8’d2	8mA
8’d3	12mA
….	….
8’d255	1020mA

12.3.3.2.26 Force Event Register for Auto CMD Error Status (Offset = 0x50)

Table 12-77 lists the Force Event Register for the error status of Auto CMD. The Force Event Register is not a physical register. It is an address at which the Auto CMD error status register can be written. The force event register is only for debugging.

Table 12-77. Force Event Register for Auto CMD Error Status (Offset = 0x50)

Bit	Name	Type	Description
[15:8]	-	-	Reserved
7	f_cmd_no_ex_by_cmd12	WO	Force event for the Command Not Executed by Auto CMD12 Error
[6:5]	-	-	Reserved
4	f_cmd_idx_err	WO	Force event for the Auto CMD Index Error
3	f_cmd_end_bit_err	WO	Force event for the Auto CMD End Bit Error
2	f_cmd_crc_err	WO	Force event for the Auto CMD CRC Error
1	f_cmd_timeout_err	WO	Force event for the Auto CMD Timeout Error
0	f_cmd12_no_ex	WO	Force event for the Auto CMD12 Not Executed

12.3.3.2.27 Force Event Register for Error Interrupt Status (Offset = 0x52)

Table 12-78 lists the Force Event Register for the error interrupt status. The Force Event Register is not a physical register. It is an address at which the error interrupt status register can be written. This Force Event Register is for debugging only. The effect of writing to this address will be reflected in the error interrupt status register if the corresponding bit of the error interrupt status enable register is set.

Table 12-78. Force Event Register for Error Interrupt Status (Offset = 0x52)

Bit	Name	Type	Description
[15:13]	Reserved	-	-
12	f_ahb_resp_err	WO	Force Event for the AHB response Error
[11:10]	Reserved	-	-
9	f_adma_err	WO	Force Event for the ADMA Error
8	f_auto_cmd_err	WO	Force Event for the Auto CMD Error
7	f_cur_lim_err	WO	Force Event for the Current Limit Error
6	f_data_end_bit_err	WO	Force Event for the Data End Bit Error
5	f_data_crc_err	WO	Force Event for the Data CRC Error
4	f_data_timeout_err	WO	Force Event for the Data Timeout Error

Bit	Name	Type	Description
3	f_cmd_idx_err	WO	Force Event for the Command Index Error
2	f_cmd_end_bit_err	WO	Force Event for the Command End Bit Error
1	f_cmd_crc_err	WO	Force Event for the Command CRC Error
0	f_cmd_timeout_err	WO	Force Event for the Command Timeout Error

12.3.3.2.28 ADMA Error Status Register (Offset = 0x54)

Table 12-79 lists the status of the ADMA Error Status Register. When the ADMA error interrupt is occurred, adma_err_st_r holds the ADMA state and the ADMA system address register holds the address around the error descriptor. When an error occurs, the host driver should watch the ADMA state to identify the error descriptor address.

Table 12-79. ADMA Error Status Register (Offset = 0x54)

Bit	Name	Type	Description
15	-	R	Reserved and fixed to '0' The error interrupts are controlled by the error interrupt signal enable register.
[7:3]	-	-	Reserved
2	adma_len_err	ROC	ADMA Length Mismatch Error This error occurs under the following two conditions: (1) When Block Count Enable is set, the total data length specified by the Descriptor table will be different from the one specified by the Block Count and Block Length. (2) The total data length cannot be divided by the block length.
[1:0]	adma_err_st_r	ROC	ADMA Error State This field indicates the state of ADMA when an error occurs during an ADMA transfer. D01 ~ D00 ADMA Error State when error occurs in the SYS_SDR register 11: ST_TFR (Transfer Data) for points next of the error descriptor 10: Never set this state (Not used) 01: ST_FDS (Fetch Descriptor) for points of the error descriptor 00: ST_STOP (Stop DMA) for points next of the error descriptor

12.3.3.2.29 ADMA System Address Register (Offset = 0x58)

Table 12-80 lists the address of the ADMA System Address register. The host controller version only supports the 32bit address ADMA function.

Table 12-80. ADMA System Address Register (Offset = 0x58)

Bit	Name	Type	Description
[31:0]	adma_lo_addr_r	R/W	Lower 32bit ADMA System Address The 32bit address descriptor only uses the lower 32bit of the ADMA system address register. ADMA ignores the lower 2bit of this register and assumes it to be 2'b00.

12.3.3.2.30 Preset Value Register (Offset = 0x60 ~ 0x6F)

Table 12-81 lists a set of preset values per card or device.

Table 12-81. Preset Value Registers (Offset = 0x60 ~ 0x6F)

Offset	Preset Value Register	Signal Voltage
060h	Preset value for initialization	3.3V or 1.8V
062h	Preset value for the default speed	3.3V
064h	Preset value for the high speed	3.3V
066h	Preset value for SDR12	1.8V
068h	Preset value for SDR25	1.8V
06Ah	Preset value for SDR50	1.8V
06Ch	Preset value for SDR104	1.8V
06Eh	Preset value for DDR50	1.8V

When Preset Value Enable in the Host Control 2 register is set to '1', SDCLK Frequency Select in the Clock Generator Select in the Clock Control register and Driver Strength Select in the Host Control 2 register will be automatically set based on the selected bus speed mode. This means that the host driver does not need to set these fields when preset is enabled. A Preset Value for Initialization (060h) will not be selected by the bus speed mode. Before starting the initialization sequence, the Host Driver needs to set a clock preset value to SDCLK Frequency Select in the Clock Control register. Preset Value Enable can be set after the initialization is completed.

Table 12-82 lists the speed mode selections used for the Preset Value register.

Table 12-82. Speed Mode Selection

Selected Bus Speed Mode	1.8V Signaling Enable (Host Control 2)	High Speed Enable (Host Control 1)	UHS-I Mode Selection (Host Control 2)
Default speed	0	0	Don't care
High speed	0	1	Don't care
SDR12	1	Don't care	000b
SDR25	1	Don't care	001b
SDR50	1	Don't care	010b
SDR104	1	Don't care	011b
DDR50	1	Don't care	100b
Reserved	1	Don't care	101b ~ 111b

Table 12-83 defines the Preset Value register based on the speed modes.

Table 12-83. Preset Value Register Based on Speed Mode

Bit	Name	Type	Description
[15:14]	driver_str_sel	HwInit	Driver Strength Select Value Driver Strength is supported by the 1.8V signaling bus speed modes. This field is meaningless for the 3.3V signaling. 2'b11: Driver type D is selected. 2'b10: Driver type C is selected. 2'b01: Driver type A is selected. 2'b00: Driver type B is selected.
[13:11]	-	-	Reserved
10	clk_gen_sel	RO	Clock Generator Select Value The current version does not support the programmable clock generator and is fixed to '0'.
[9:0]	SDCLK_freq_sel	HwInit	SDCLK Frequency Select Value The 10bit preset value for setting the SDCLK frequency select in the Clock Control Register. The register is described by a host system.

12.3.3.2.31 Host Controller Version Register (Offset = 0xFE)

Table 12-84 lists the bit assignment of the Host Controller Version Register.

Table 12-84. Host Controller Version Register (Offset = 0xFE)

Bit	Name	Type	Description
[15:8]	vendor_ver_num	HwInit	Vendor Version Number This field is reserved for the vendor version number. The host driver should not use this field.
[7:0]	spec_ver_num	RO	Specification Version Number This field shows the specification version of the supportable host controller. The current version supports the SD Host Specification, Version 3.00 and is fixed to '02h'.

12.3.3.2.32 Vendor-defined Register 0 (Offset = 0x100)

Table 12-85 lists the assignment of the Vendor-defined Register 0.

Table 12-85. Vendor-defined Register 0 (Offset = 0x100)

Bit	Name	Type	Description
[31:28]	-	-	Reserved
[27:24]	NCRC	R/W	Write CRC Status Wait Cycle The host controller is used to set five SCLK clock cycles for the specifications and round-chip effect. Users can add the wait cycle for other factors. For the SDR50 and SDR104 Specifications, the wait cycle requires at least six cycles to meet the SD Specification, Version 3.00.
[23:17]	-	-	Reserved
16	int_edge_sel	R/W	Internal SCLK Edge Selection 1: The CMD and DAT lines are outputted at the rising edge of SCLK. 0: The CMD and DAT lines are outputted at the falling edge of SCLK.
[15:14]	-	-	Reserved

Bit	Name	Type	Description
[13:8]	p_lat_off	R/W	Pulse Latch Offset When the host controller uses the pulse latch to sample the read data and response, users need to set the latch offset to correctly sample the value. The value set should be small than the SDCLK Frequency Select. 3Fh Latch value at the 63rd sdclk1x rising edge after the SCLK edge … … 01h Latch value at the 1st sdclk1x rising edge after the SCLK edge 00h Latch value at the SCLK edge
[7:0]	-	-	Reserved

12.3.3.2.33 Vendor-defined Register 1 (Offset = 0x104)

Table 12-86 lists the Vendor-defined Register 1.

Table 12-86. Vendor-defined Register 1 (Offset = 0x104)

Bit	Name	Type	Description
[31:25]	-	-	Reserved
24	cmd_conflict_en	R/W	Command Conflict Checker Enable The host controller can check the conflict error of the CMD line. Users can set this bit to '1' to enable the checker.
[23:19]	-	-	Reserved
[18:16]	NSB	R/W	NSB Timing of SD Specification The host controller is set to five SCLK clock cycles based on the specification. Users can add the busy wait cycle for other factors.
[15:12]	-	-	Reserved
[11:8]	NCR	R/W	NCR Timing of SD Specification The host controller is set to 64 SCLK clock cycles for the specification. Users can add the response wait cycle for other factors.
[7:3]	-	-	Reserved

Bit	Name	Type	Description
2	mmc_boot_ack_en	R/W	MMC Booting Mode Acknowledge Enable The MMC specification defines the Booting Acknowledge that can be disabled. Users can enable the function.
[1:0]	mmc_boot	R/W	MMC Booting Mode Selection The MMC specification defines two boot modes and bus test mode. Users need to set the function to select one of these modes. Data in the boot mode can be read from the data port or transferred by SDMA. To exit from the boot mode, “software reset for all” must be set after clearing this register to ‘0’. (In the alternative boot mode, CMD0 for terminating the boot mode should be issued after setting “software reset for all”.) 2’b11: MMC bus test mode 2’b10: MMC alternative boot mode 2’b01: MMC boot mode 2’b00: Normal mode

12.3.3.2.34 Vendor-defined Register 2 (Offset = 0x108)

Table 12-87. Vendor-defined Register 2 (Offset = 0x108)

Bit	Name	Type	Description
[31:1]	-	-	Reserved
0	clk_ctrl_sw_rst	RWAC	Clock Control Software Reset Users can reset the clock control of the host controller. When this bit is set to ‘1’, the clock control will be reset to the default value.

12.3.3.2.35 Vendor-defined Register 3 (Offset = 0x10C)

Table 12-88. Vendor-defined Register 3 (Offset = 0x10C)

Bit	Name	Type	Description
[31:29]	-	-	Reserved
[28:24]	sd_delay_sel_bound	R/W	Repeating Times of Auto-tuning Operation The default value is 31, which represents that a total of 32 times of tuning pattern reads will be executed. This value should be matched with the stage of the multiphase DLL.
[23:21]	-	-	Reserved

Bit	Name	Type	Description
[20:16]	sd_delay_val	R/W	SD Delay Value Selection The output delay value of the host controller to the multiphase DLL for delay control. Users can set the value or use auto-tuning to find the best setting value.
[15:13]	-	-	Reserved
[12:8]	crc16_error_thres	R/W	DATA CRC16 Error Threshold Value When the times of CRC16 error reaches this threshold value, the tuning_err_r status (Offset 0x32 bit 10) will be set.
[7:5]	-	-	Reserved
[4:0]	tuning_success_thres	R/W	Auto-tuning Operation Success Times Threshold If the success time of the tuning pattern read operation is greater than this threshold, the auto-tuning operation will be successful. The value is suggested to be greater than 8 when the total times of the tuning pattern read operation is 32.

12.3.3.2.36 Vendor-defined Register 4 (Offset = 0x110)

Table 12-89. Vendor-defined Register 4 (Offset = 0x110)

Bit	Name	Type	Description
[31:0]	tuning_record	RO	Auto-tuning Operation Result The auto-tuning operation result can be checked in this register. Bit[x] (x = 0 ~ 31) 1: Tuning pattern read is correct when sd_delay_val is x. 0: Tuning pattern read is incorrect when sd_delay_val is x. sd_delay_val can be adjusted by checking the result in this register.

12.3.3.2.37 Vendor-defined Register 5 (Offset = 0x114)

Table 12-90. Vendor-defined Register 5 (Offset = 0x114)

Bit	Name	Type	Description
[31:4]	-	-	Reserved

Bit	Name	Type	Description
[3:0]	db_timeout	R/W	Card Insertion De-bounce Cycle 0: 2^9 sys_hclk cycles 1: 2^{10} sys_hclk cycles … 15: 2^{24} sys_hclk cycles

12.3.3.2.38 Vendor-defined Register 6 (Offset = 0x118)

Table 12-91. Vendor-defined Register 6 (Offset = 0x118)

Bit	Name	Type	Description
[31:1]	-	-	Reserved
0	hburst_incr	R/W	AHB Bus Burst Type 1: AHB master uses INCR as the AHB burst type. 0: AHB master uses SINGLE and INCR4 as the AHB burst type.

12.3.3.2.39 Vendor-defined Register 7 (Offset = 0x11C)

Table 12-92. Vendor-defined Register 7 (Offset = 0x11C)

Bit	Name	Type	Description
[31:1]	-	-	Reserved
0	ahb_resp_err_sts	R/W1C	AHB Master Response Error Status This bit will be set when the AHB master receives an error type response.

12.3.3.2.40 Vendor-defined Register 8 (Offset = 0x120)

Table 12-93. Vendor-defined Register 8 (Offset = 0x120)

Bit	Name	Type	Description
[31:1]	-	-	Reserved
0	ahb_resp_err_sts_en	R/W	AHB Master Response Error Status Enable 1: Enable the AHB master response error status 0: Disable the AHB master response error status

12.3.3.2.41 Vendor-defined Register 9 (Offset = 0x124)

Table 12-94. Vendor-defined Register 9 (Offset = 0x124)

Bit	Name	Type	Description
[31:1]	-	-	Reserved
0	ahb_resp_err_sts	R/W	AHB Master Response Error Signal Enable 1: When the AHB master response status is set, the interrupt generation will be enabled. 0: When the AHB master response status is set, the interrupt generation will be disabled.

12.3.3.2.42 DMA Handshake Enable Register (Offset = 0x128)

Table 12-95. DMA Handshake Enable Register (Offset = 0x128)

Bit	Name	Type	Description
[31:1]	-	-	Reserved
0	dma_hsk_en	R/W	DMA Handshake Enable 1: Enable the DMA handshake protocol for the data port access 0: Disable the DMA handshake protocol for the data port access Note: When this bit is set to '1', buf_r_rdy_r and buf_w_rdy_r will not be set and become ineffective. Data are transferred by external DMA and the flow is controlled by the DMA handshake protocol.

12.3.3.2.43 Hardware Attributes Register (Offset = 0x178)

Table 12-96. Hardware Attributes Register (Offset = 0x178)

Bit	Name	Type	Description
[31:9]	-	-	Reserved
[8:0]	hw_config	RO	Bit 8 1: ASYNC, 0: SYNC (SD interface and core) Bit 7 1: 8bit SD data bus 0: 4bit SD data bus Bit 6 1: CPRM present

Bit	Name	Type	Description
			0: CPRM absent Bit 5 1: DLL present 0: DLL absent Bit 4 1: DATA FIFO is 4K SRAM. Bit 3 1: DATA FIFO is 2K SRAM. Bit 2 1: DATA FIFO is 1K SRAM. Bit 1 1: DATA FIFO is 16-entry register. Bit 0 1: DATA FIFO is 8-entry register. Note: For bit 4 to bit 0, it has only 1 bit and is set to '1'.

12.3.3.2.44 IP Revision Register (Offset = 0x17C)

Table 12-97. IP Revision Register (Offset = 0x17C)

Bit	Name	Type	Description
[31:0]	Rev_num	RO	Revision number

12.3.4 Functional Description

This chapter contains the following sections:

- Block Description
- DMA Transaction
- SD Command and Data Input/Output Timing

12.3.4.1 Block Description

The SD host controller includes the AHB master interface, AHB slave interface, Command Unit, DATA Unit, Control Register, Configurable FIFO, Clock Divider, and Configurable CPRM function.

12.3.4.1.1 AHB Master

When the DMA transfer is used, the AHB Master will initiate a read/write transfer with the memory and the minimum data size will be limited to four bytes. The DMA algorithms defined in the SD host controller standard specification include SDMA and ADMA (Advanced DMA). SDMA has a disadvantage that the DMA interrupt generated at the every page boundary will disturb CPU in reprogramming the new system address. The new ADMA uses the link-list mechanism to provide a higher data transfer speed.

12.3.4.1.2 AHB Slave

The AHB slave is implemented to read/write the host control registers by using the processor through the slave interface. The AHB slave provides the byte, half-word, or word data access. Please note that the address must be 0x20 no matter the HSIZE is byte, half-word, or word when accessing the data port though the register. The DMA handshake protocol is supported for the flow control when using the external DMA. Please note that the setting of the burst size in the DMA should match the setting of the block size if the FIFO size is in 256-word, 512-word, or 1024-word. For example, when using the SD memory card with the 512bytes block size, the burst size should be 128 (The data width is 32bits). If the FIFO size is 8-word or 16-word, no matter the block size setting, the burst size should be 8 or 16, respectively (The data width is 32bits).

12.3.4.1.3 Command Unit

The command unit is implemented to transmit command to the SD card and receive the response from the SD card. The built-in 7bit CRC check and generation are involved in the command unit.

12.3.4.1.4 Data Unit

The data unit is implemented to transmit data to the SD card and receive data from the SD card. The built-in 16bit CRC check and generation are involved in the data unit.

12.3.4.1.5 Control Register

The control register can be divided into two groups: Command register and status register. To activate a command or data transfer with the card, the command register should be programmed properly. The failure, success, or received command/data CRC status will be reported in the status register under any condition. The control register conforms to the SD Host Standard Specification described in Chapter 4.

12.3.4.1.6 Clock Divider

In the card identification mode, the maximum clock frequency is 400kHz. During the data transfer, the maximum clock frequency rate should not exceed 25MHz in the default speed mode, 50MHz in the high-speed mode for the SD card, and 52MHz for the MMC card. When the card clock is stopped, the activated command, the received response, or the transfer data may be temporarily suspended in the command unit the sufficient clock cycles re-start for the card operation.

12.3.4.2 DMA Transaction

The SD host controller is compliant with the SD host controller standard specification, Version 2.0. The DMA algorithms (Also called SDAM and ADMA) are implemented in this host controller. SDMA has a disadvantage that the DMA interrupt generated at every page boundary will disturb CPU in reprogramming the new system addresses. Since the host driver is required to reprogram the next system address to restart the transfer, the performance of the DMA transfer will be affected. The new ADMA implements the link-list mechanism to provide the higher data transfer speed. The host driver can program a list of descriptor line to transfer data without interrupting the host driver. Figure 12-4 shows the block diagram of ADMA. The descriptor table is created in the system memory by the host driver.

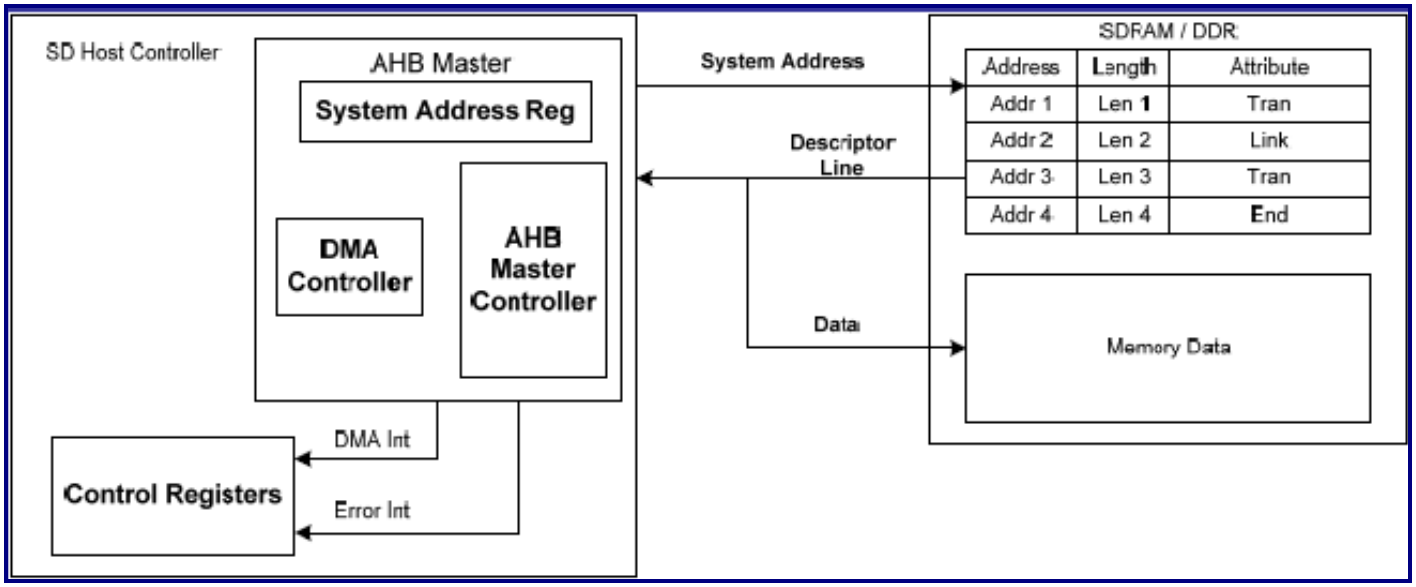


Figure 12-4. Block Diagram of ADMA

12.3.4.2.1 Data Address and Data Length

The minimum unit of an address is four bytes and the data address should be word-aligned. The maximum data length of each descriptor line should be less than 64Kbytes. The total data length should be equal to the block count multiplied by the block size. When multiple blocks are transferred with the continual descriptor lines, the data length of each descriptor line should be the multiple of the block size. The block count register is defined as a 16bit register, which will limit the maximum transfer to 65535 blocks.

(1) The address should be word-aligned (4byte).

(2) The data length of each descriptor line is less than 64Kbytes.

(3) Total data length = Len 1 + Len 2 + …… + Len n = Multiple block size

12.3.4.2.2 Descriptor Table

Figure 12-5 shows the 32bit address descriptor table. Each descriptor line requires the 64bit memory space. The address field stores the 32bit data address. The length field defines the data length of each descriptor line.

Attribute

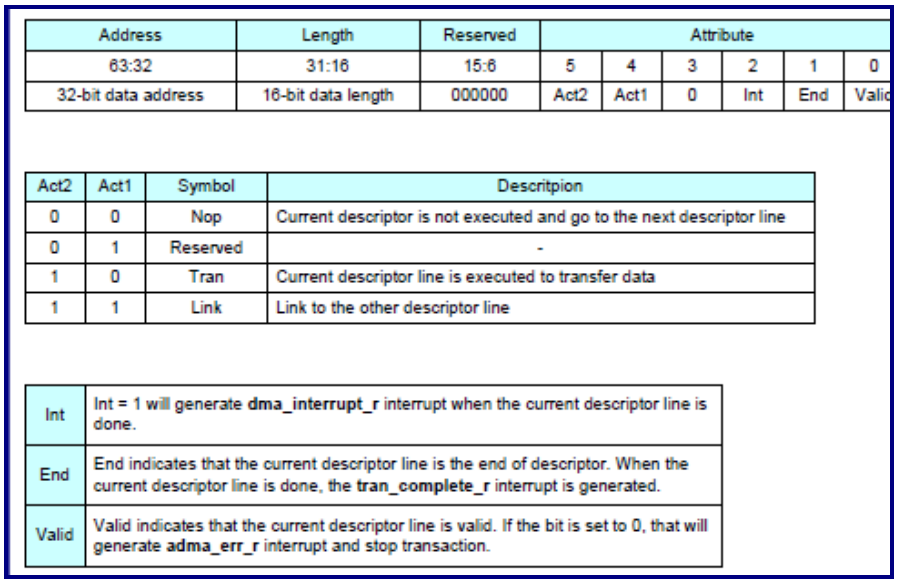
The data in the attribute field is used to control each descriptor line.

Symbol

Three action symbols are specified:

The “Nop” operation indicates that the current descriptor line is not in operation and fetches the next descriptor line. The “Tran” operation indicates that the current descriptor line is the data transfer operation, which is designated by the data address and data length. The “Link” operation indicates the connection to the other descriptor lines, including the next system address and the other system addresses. The next descriptor line of the system address is generated by the data address of the current descriptor line.

Table 12-98 lists the defined the data length according to the length field.



Address	Length	Reserved	Attribute					
63:32	31:16	15:6	5	4	3	2	1	0
32-bit data address	16-bit data length	000000	Act2	Act1	0	Int	End	Valid

Act2	Act1	Symbol	Descritpion
0	0	Nop	Current descriptor is not executed and go to the next descriptor line
0	1	Reserved	-
1	0	Tran	Current descriptor line is executed to transfer data
1	1	Link	Link to the other descriptor line

Int	Int = 1 will generate **dma_interrupt_r** interrupt when the current descriptor line is done.
End	End indicates that the current descriptor line is the end of descriptor. When the current descriptor line is done, the **tran_complete_r** interrupt is generated.
Valid	Valid indicates that the current descriptor line is valid. If the bit is set to 0, that will generate **adma_err_r** interrupt and stop transaction.

Figure 12-5. 32bit Descriptor Table

Table 12-98. ADMA Data Length Field

Length Field	Data Length
16’hFFFF	65535 bytes
….	….
16’h0002	2 bytes
16’h0001	1 bytes
16’h0000	65536 bytes

12.3.4.2.3 ADMA States

Figure 12-6 depicts the state diagram of the ADMA transfer. Four states are addressed in this state diagram: Stop transfer state, fetch descriptor state, change address state, and transfer data state. The state operations are defined in Table 12-99. Please note that the ADMA transfer does not support the suspend/resume function. The ADMA transfer can neither be stopped at the block gap nor continue the available requests. If an error occurs during the ADMA transfer, the ADMA operation may stop and generate the **adma_err_r** interrupt. The **adma_err_st_r** registers hold the ADMA state and **adma_sys_addr_r** holds the system address around the error descriptor. The host driver can read the **adma_err_st_r** registers to check the state of the ADMA operation.

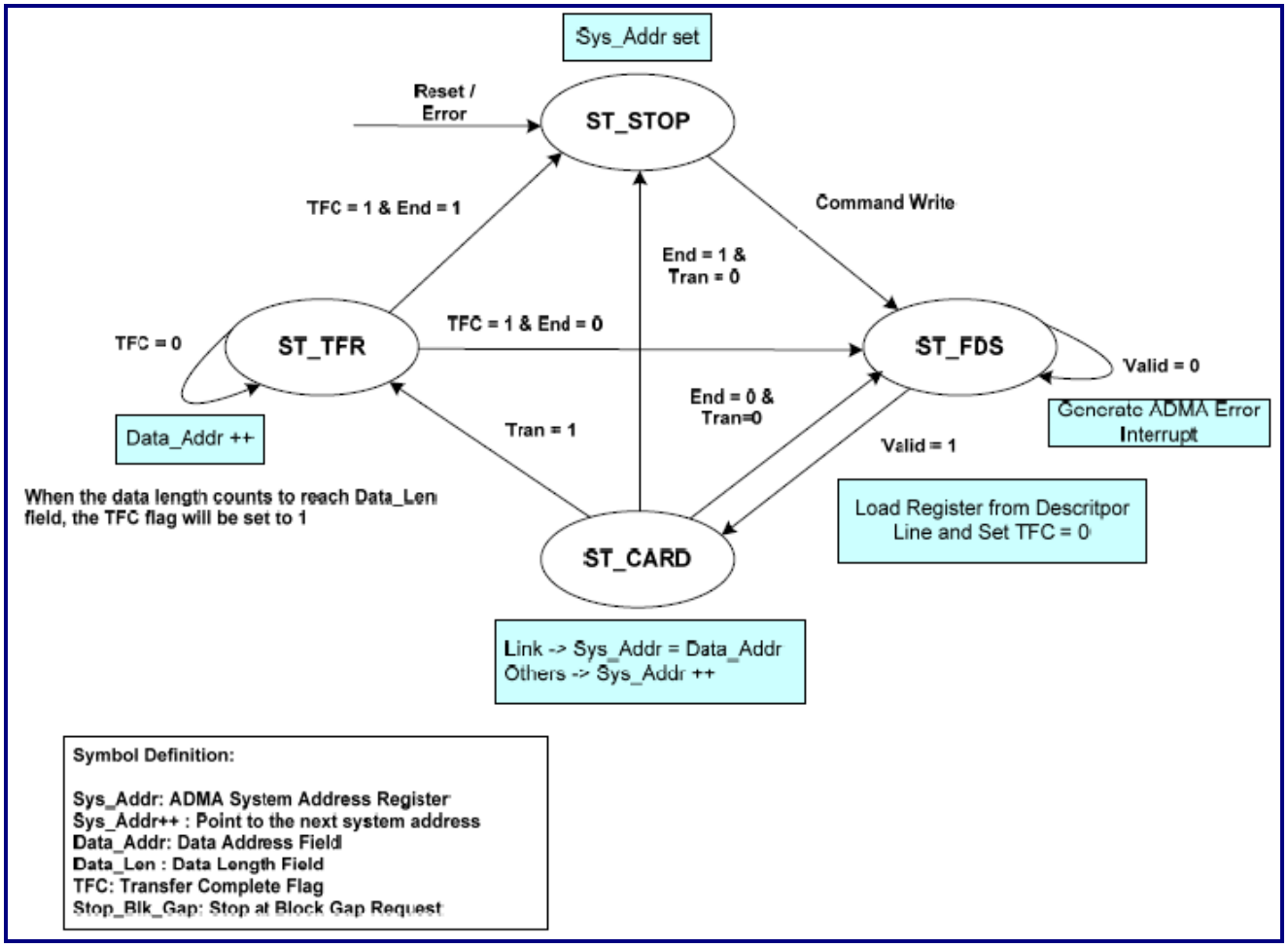


Figure 12-6. State Diagram of ADMA

Table 12-99. ADMA States

State	Description
ST_STOP (Stop DMA)	ADMA will stay at this state under the following conditions: (1) When Reset or Error occurs. (2) When all descriptor line data transfers are completed. If the command register is written, the state will go to the ST_FDS state.
ST_FDS (Fetch Descriptor)	In this state, a descriptor line will be fetched from the system memory to the host controller. After fetching a descriptor line, go to the ST_CADR state.
ST_CADR (Change Address)	In this state, the line system of the next descriptor address will be acquired. In the link operation, the next system address will be loaded by the data address of current descriptor line and the state will change to the ST_FDS state to fetch the next descriptor line. In the other operations, the current system address will be incremented to point to the next descriptor line and the state will change to the ST_TFR state.
ST_TFR (Transfer Data)	The data transfer of one descriptor line will be executed between the system memory and the SD card. If the data transfer is not completed (End = '0'), it will go to the ST_FDS state to fetch the next descriptor line. If the data transfer is completed, it will go to the ST_STOP state.

12.3.4.3 SD Command and Data Input/Output Timing

For the SD interface timing, users can adjust the input/output timing to meet the timing margin.

12.3.4.3.1 Command and Data Output

For the SD interface output timing, the timing adjustment is designed in SD controller. Table 12-100 lists different cases of the timing adjustment.

Table 12-100. Timing Adjustment of SD Interface Output

Clock Divided is Zero	Clock Divided is One	DDR Mode Enable	Int. Edge Enable	Output Timing Case
No	No	No	No	Case 1
No	No	No	Yes	Case 2
No	No	Yes	No	Case 2
No	No	Yes	Yes	Case 2
No	Yes	No	No	Case 1
No	Yes	No	Yes	Case 2
No	Yes	Yes	No	Case 3
No	Yes	Yes	Yes	Case 3
Yes	No	No	No	Case 3
Yes	No	No	Yes	Case 2
Others				Inhibit

Where "clock divided is zero" means that the value of the 10bit clock divider is zero; "clock divided is one" means that the value of the 10bit clock divider is one; "DDR mode enable" means that UHS_mode_sel is set to the DDR50 mode (Host Control 2 register, 0x3E, bits[2:0]); and "Int. Edge Enable" means that bit [16] of the vendor register 0 (0x100) is set to '1'.

The output timing adjustment has three cases. Please refer to Figure 12-7 for details.

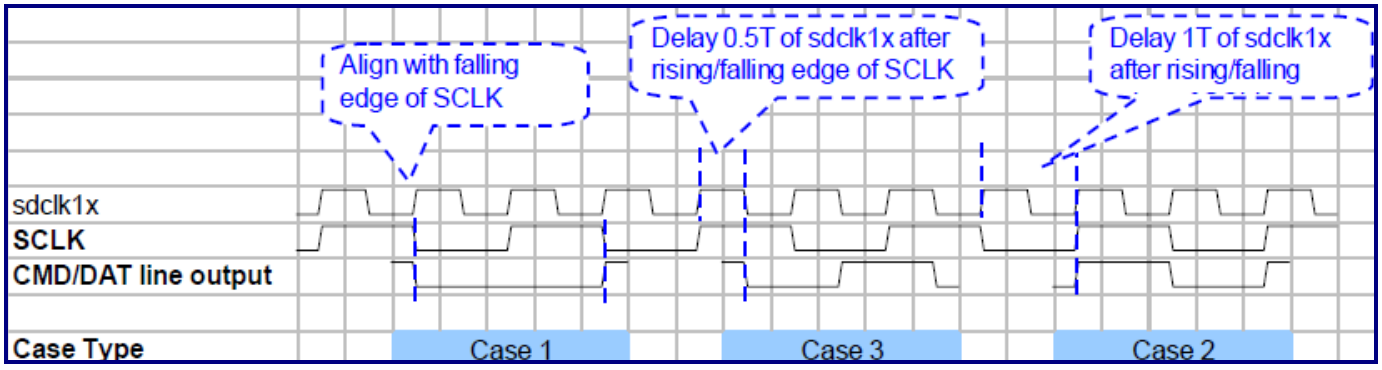


Figure 12-7. SD Output Timing

12.3.4.3.2 Command and Data Input Latching

For the SD input timing, users need to select the latching mechanism to avoid incorrect latching points. The SD controller provides pulse data latching mechanisms.

- Pulse latching mechanism: This mechanism does not require extra multiphase DLL in the system. In general, the system interface has the round-trip effect between the chip and device. To eliminate the round-trip latency, the SD controller designs one adjustable latching pointer to avoid the issue. Users can set bits[13:8] of the vendor register 0 (0x100) to select the latching pointer. The default value is zero and the latching pointer is at the rising edge of the SD clock (SCLK). The latching pointer delays one sdclk1x period once the value increases by one.

12.3.5 Initialization

This section contains the following sections:

- Program Sequence

12.3.5.1 Program Sequence

This section defines the basic program sequence flow chart.

12.3.5.1.1 Card Detection

Figure 12-8 shows the flow of detecting a card. Each step is executed as follows:

(1) Set the vendor defined register 5 to decide the de-bouncing time for card detection.

(2) To enable the normal interrupt status enable register and normal interrupt signal register for card detection, write '1' to the following bits:

card_insert_st_en and **card_remove_st_en** in the normal interrupt status enable register

card_insert_sig_en and card_remove_sig_en in the normal interrupt signal enable register

(3) Write '1' to clear the interrupt status when the host driver detects the card detection interrupt.

If the **card_insert_r** interrupt is generated, write '1' to clear this status register.

If the **card_remove_r** interrupt is generated, write '1' to clear this status register.

(4) Check the present state register to check whether the card is inserted or removed. When **sys_card_inst** is set to '1', the host driver can continue supplying the power and clock to the card.

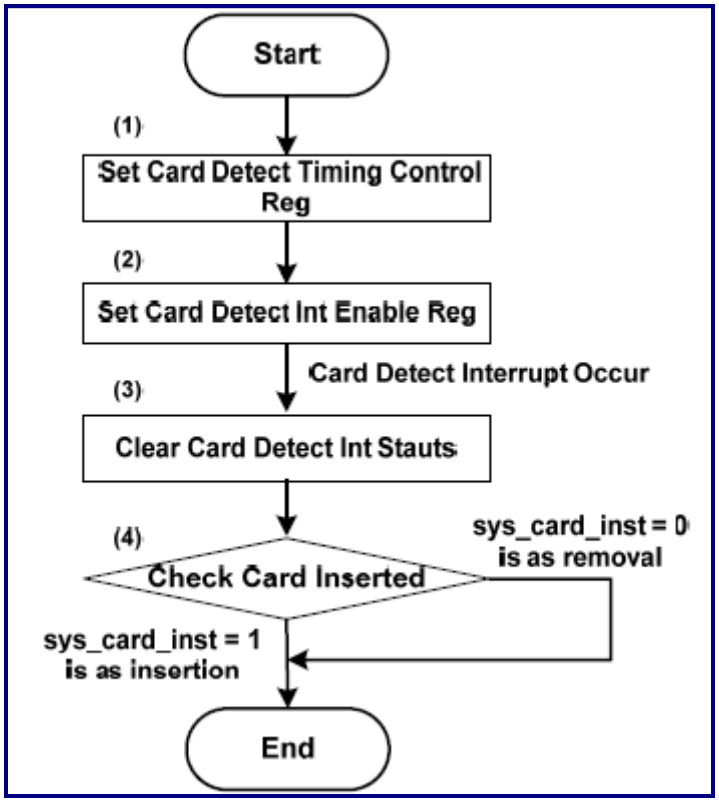


Figure 12-8. Card Detect Sequence

12.3.5.1.2 SD Clock Control

Figure 12-9 shows the flow of detecting an SD card. Each step is executed as follows:

(1) Set **inter_clk_en** and **sd_clk_sel** in the clock control register to generate clock.

(2) Check **clk_stable** in the clock control register to determine whether the clock is stable or not.

(3) Set **sd_clk_en** in the clock control register to output the clock to the **io_sd_clk** pin. If the host driver has to stop the SD clock, **sd_clk_en** should be set to '0'. The host controller then stops the SD clock. When the host driver needs to change the SD clock, **sd_clk_en** should also be set to '0'.

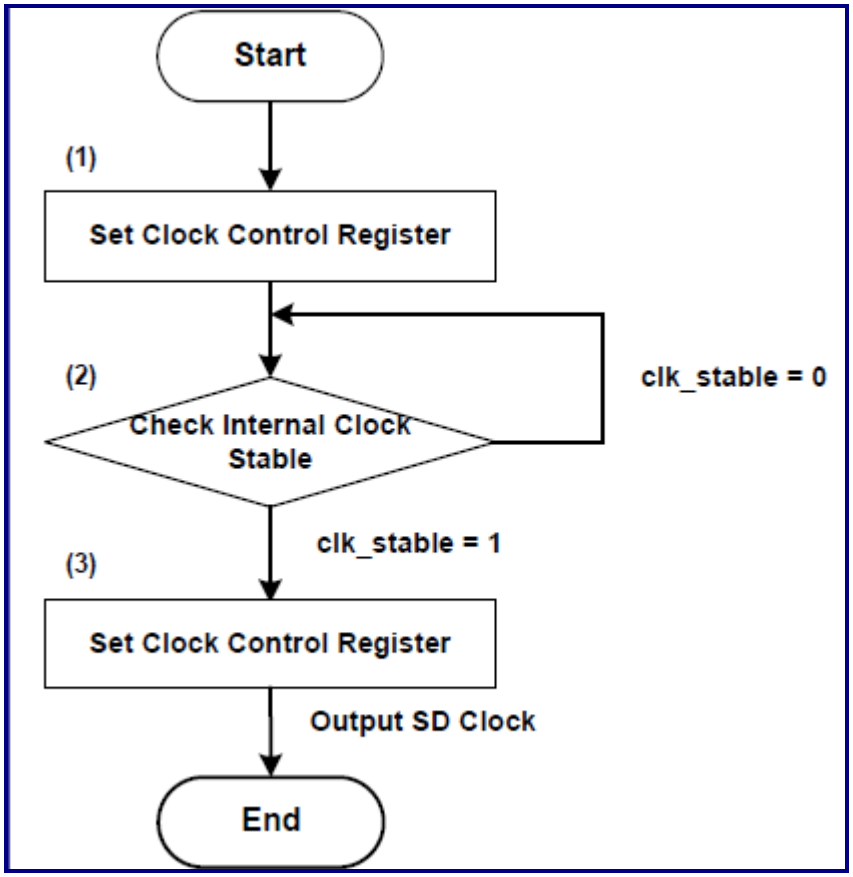


Figure 12-9. SD Clock Control Sequence

12.3.5.1.3 Card Initialization and Identification

This section introduces the initialization and identification of the SD, SDIO, and MMC cards. Figure 12-10 shows the flows of the card initialization and card identification.

(1) The SD bus mode is selected by resetting CMD0.

(2) Issue CMD8 to check the high-capacity SD memory card.

(3) The legacy cards (Not the SD card) does not respond to CMD8. Set the F8 flag to '0' (Used in Step (11)), go to Step (5). Only cards with version 2.0 or higher can respond to CMD8. The host controller needs to check the validity of CRC of the response, verify VHS, check the pattern in an argument that equals to VCA, and check the pattern in the response. If the card response is okay, then set the F8 flag to '1' and go to Step (5). If the response check fails, go to Step (4).

(4) If the initialization fails, the host driver should retry again.

(5) Issue CMD5 to obtain SDIO OCR by setting the voltage window to '0' in the argument.

(6) If no response is detected, the card will not have the SDIO function. Set the SDIO flag to '0' and go to Step (11). If the response is okay, go to Step (7). If the response has an error, set the SDIO flag to '0', go to Step (10). The SDIO flag verifies the initialization of the SDIO functions.

(7) Issue CMD5 to start the initialization by setting the voltage window. If the supplied voltage does not match with the voltage windows of the card, the card will enter an inactive state and will not return to the response.

(8) If no response or error response is received, set the SDIO flag to '0', go to Step (10). If the response is okay, go to Step (9).

(9) Check the busy status bit in the response. If the busy status is released, set the SDIO flag to '1', and go to Step (10). If the busy status is not released, repeat Step (7) until the busy status is released. When the detection timeout of 1 second exits from a loop, set the SDIO flag to '0', go to Step (10).

(10) When all responses at Step (6) and Step (8) are valid, the **MP** (Memory Present) flag in the response can be checked. If the response is okay and **MP** = '0', go to step (28). Otherwise, go to Step (11).

(11) Check the F8 flag set in Step (3). If F8 = '1', go to Step (21); otherwise, go to Step (12).

(12) Receiving OCR can issue an ACMD41 command with a voltage window (Bit 23 to Bit 0) if the argument is set to '0'. CMD55 should be issued before the ACMD41 command.

(13) If the ACMD41 response is not received, the card will not be a SD card, and go to Step (14). If the card responds to ACMD41, go to Step (17); otherwise, go to Step (29).

(14) In this step, the card may be the MMC card, and the host driver can issue the CMD1 by setting the supply voltage to the voltage window.

(15) If the CMD1 response indicates that the card is busy or the host omits the voltage range, the host driver will repeat to issue CMD1. If the OCR response is a non-compatible voltage range, go to step (29).

(16) The host recognizes that the card is a Multi Media Card (MMC) and quits the card initialization.

(17) The memory portion starts the initialization by issuing ACMD41 by setting the supply voltage to the voltage window. If the supplied voltage does not match the voltage window of card, the card will enter the inactive state and will not return the response.

(18) If no response or no error response is received, go to Step (29). If a response is received, go to Step (19).

(19) Check the busy status in a response. If busy is released, go to Step (20).

(20) If the host recognizes that the card is Version 1.xx standard capacity SD memory card, go to Step (32).

(21) OCR is available by issuing ACMD41 and setting the voltage window (Bit 23 to bit 0) in the argument, and it is set to '0'. CMD55 should be issued before ACMD41.

(22) If the card responds to CMD55, it may also respond to CMD41. If the response of ACMD41 is OK, go to Step (23). Otherwise, go to Step (29).

(23) The memory portion starts the initialization by issuing ACMD41 and setting the supply voltage to the voltage window. If the supplied voltage does not match with the voltage window of card, the card will enter the inactive state and does not return the response. HCS in the argument is set to '1', which indicates that the high capacity SD memory card is supported.

(24) If no response or error response is received, go to Step (29). If a good response is received, go to Step (25).

(25) Check the busy status in the response. If busy is released, go to Step (26). While the busy is indicated, repeat to issue ACMD41.

(26) CCS in the response is valid after busy is released. If CCS = '0', it indicates the standard capacity SD memory card and go to Step (27). If CCS = '1', it indicates the High Capacity SD Memory Card and goes to Step (28)

(27) If the host recognizes that the card is the standard capacity SD memory card, go to Step (32).

(28) If the host recognizes that the card is the high capacity SD memory card, go to Step (32).

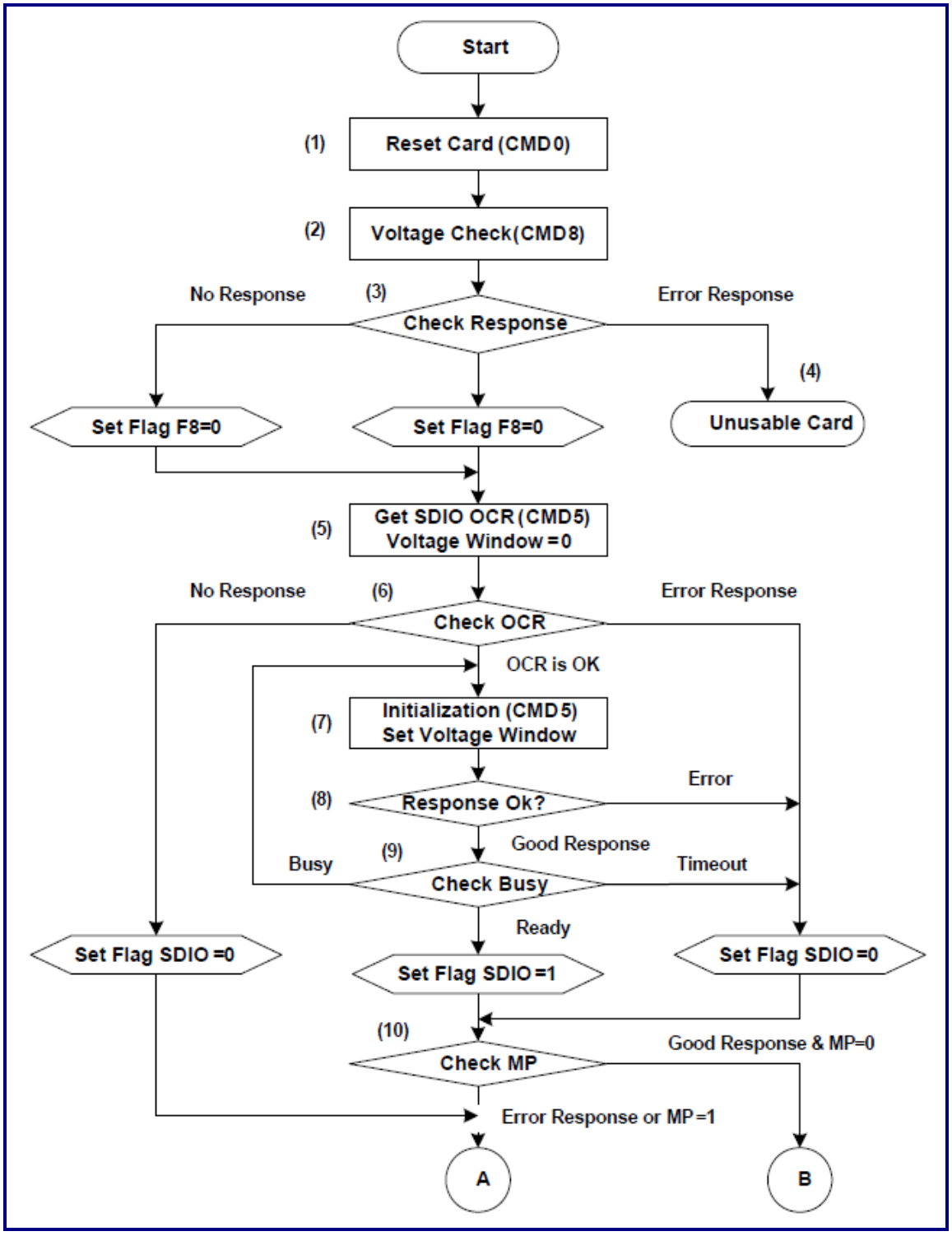
(29) Check the SDIO flag. If SDIO = '1', go to Step (30).

(30) The host recognizes that the card is the SDIO card and go to Step (33).

(31) The host recognizes that the card is unusable.

(32) In case of memory card, CMD2 is issued to get CID, and go to Step (33).

(33) CMD3 is issued to get RCA. If the RCA number is '0', the host should issue CMD3 again.


Start
(1)
Reset Card (CMD0)
(2)
Voltage Check(CMD8)
No Response
(3)
Check Response
Error Response
(4)
Set Flag F8=0
Set Flag F8=0
Unusable Card
(5)
Get SDIO OCR (CMD5)
Voltage Window = 0
No Response
(6)
Check OCR
Error Response
OCR is OK
(7)
Initialization (CMD5)
Set Voltage Window
(8)
Response Ok?
Error
Busy
(9)
Check Busy
Good Response
Timeout
Ready
Set Flag SDIO =0
Set Flag SDIO =1
Set Flag SDIO =0
(10)
Check MP
Good Response & MP=0
Error Response or MP =1
A
B

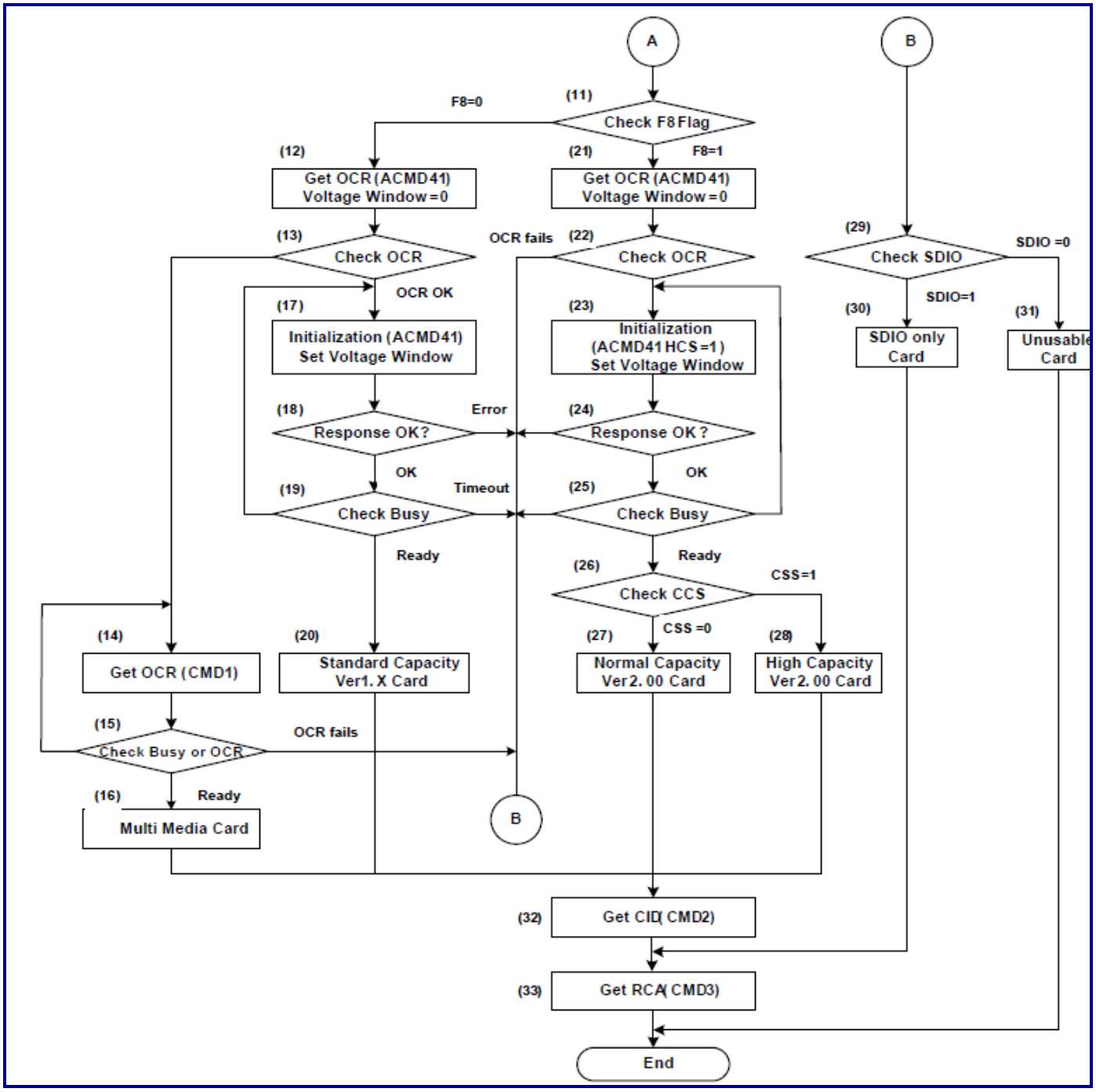


Figure 12-10. Card Initialization and Identification

12.3.5.1.4 Changing Bus Width

Figure 12-11 shows the flow of changing the bus width. Each step is executed as follows:

(1) Set **card_int_st_en** at the normal interrupt status to enable the register to stop detecting the card interrupt when changing the bus width.

(2) Detect the SDIO memory card. If the card is an SDIO memory card, go to Step 3. If the card type is other than the SDIO memory card, go to Step 4.

(3) Set the "**IENM**" bit of CCCR in the SDIO card by using CMD52.

(4) For the SD memory card, the bus width can be changed by using **ACMD6**. For the SDIO memory card, the bus width can be changed by using **CMD52** to set the "**Bus Width**" bit of the Bus Interface Control Register in CCCR. For the MMC memory card, the bus width can be changed by using **CMD6**.

(5) The host controller for the MMC memory card should set the **bus_width_8** bit in the extended bus control register to '1' to activate the 8bit mode. The host controller should set the **data_width** bit in the host control register to '1' to activate the 4bit mode. The host controller should set the **data_width** bit to '0' to activate the 1bit mode.

(6) For the SDIO memory card, go to Step 7. For other cards, go to "End".

(7) Set the "**IENM**" bit of CCCR in the SDIO card to '1' by using **CMD52**.

(8) Set the **card_int_st_en** bit in the normal interrupt status enable register to '1'.

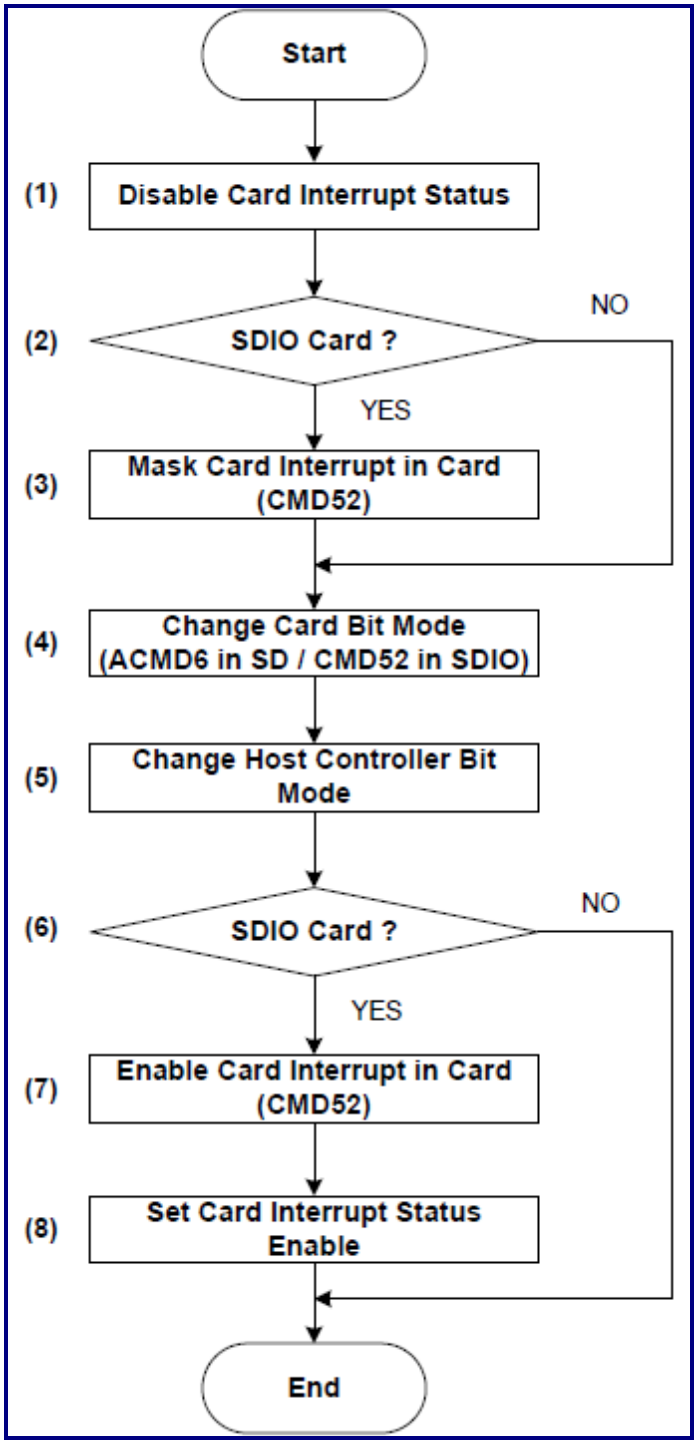


Figure 12-11. Change Bus Width Sequence

12.3.5.1.5 Command Transfer without Data Transfer

In this section, the sequence of a command transfer is explained. Figure 12-12 shows the sequence of a command transfer. Each step is executed as follows:

(1) Check **cmd_inhibit_c** in the present state register. Repeat this step until **cmd_inhibit_c** becomes '0'. When **cmd_inhibit_c** is set to '1', the SD command bus will be used and the host driver cannot issue a SD command.

(2) If the previous SD command uses the SD data bus with a busy signal, go to Step 3. If the previous SD command uses the SD data bus without a busy signal, go to Step 5.

(3) If the host driver issues an abort command, go to Step 5. If the host driver does not issue an about command, go to Step 4.

(4) Check **cmd_inhibit_d** in the present state register. Repeat this step until **cmd_inhibit_d** becomes '0'.

(5) Set the argument to the argument register

(6) Set the command register

(7) Wait for the **cmd_complete_r** interrupt. If the **cmd_complete_r** interrupt occurs, go to Step 8.

(8) Write '1' to clear **cmd_complete_r** in the normal interrupt status register

(9) Check the response register to obtain information

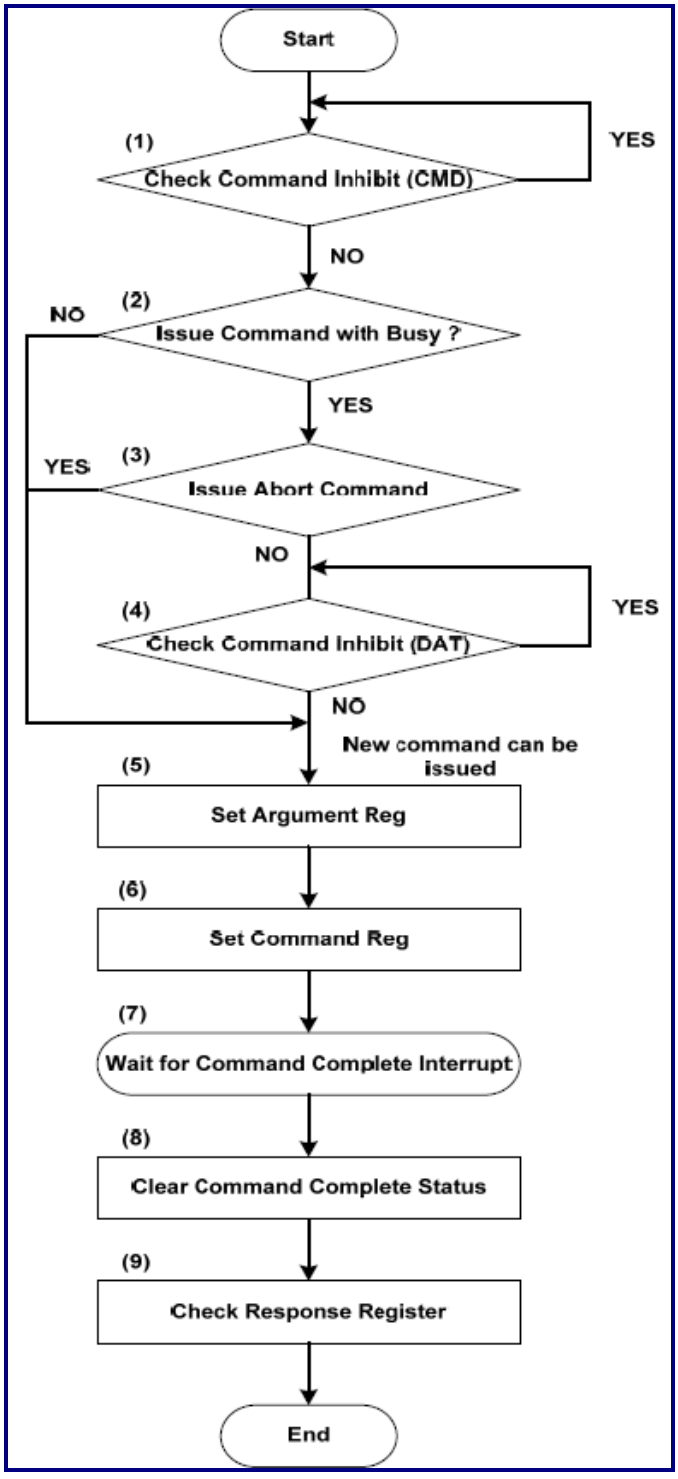


Figure 12-12. Sequence of Command Transfers

12.3.5.1.6 Data Transfer without DMA

This section describes the sequence of a data transfer without DMA. Figure 12-13 shows the flow of a data transfer. Each step is executed as follows:

(1) Set the length to **blk_size_r** in the block size register

(2) Set the count to **blk_cnt_r** in the block count register

(3) Set the argument to the argument register

(4) Set the transfer mode register

(5) Set the command register

(6) Wait for the **cmd_complete_r** interrupt[Note]

(7) When an interrupt occurs, the host driver writes '1' to clear the **cmd_complete_r** status in the normal interrupt status register.

(8) Check the response register to obtain information

(9) In case of a write transfer, go to Step 10. In case of a read transfer, go to Step 14.

(10) Wait for the **buf_w_rdy_r** interrupt

(11) When an interrupt occurs, the host driver will write '1' to clear the **buf_w_rdy_r** status in the normal interrupt status register.

(12) Write the block data to the TX buffer for the write transfer

(13) When all blocks are sent, go to Step 18. If the TX buffer is full (**Buf_wen_r** = '0') or the block transfer has not completed, go to Step 10 to repeat the operation.

(14) Wait for the **buf_r_rdy_r** interrupt

(15) When an interrupt occurs, the host driver will write '1' to clear the **buf_r_rdy_r** status in the normal interrupt status register.

(16) Read the block data from the RX buffer for the read transfer

(17) When all blocks are received, go to Step 18. If the RX buffer is empty (**Buf_ren_r** = '0') or the transfer has not completed, go to Step 10 to repeat the operation.

(18) In the case of a single-block or multiple-block transfer, go to Step 19. In the case of an infinite-block transfer, go to Step 21.

(19) Wait for the **tran_complete_r** interrupt

(20) When an interrupt occurs, the host driver will write '1' to clear the **tran_complete_r** status in the normal interrupt status register.

(21) For an infinite-block transfer, the abort command is used to stop the transfer according to Section 12.3.6.1.9.

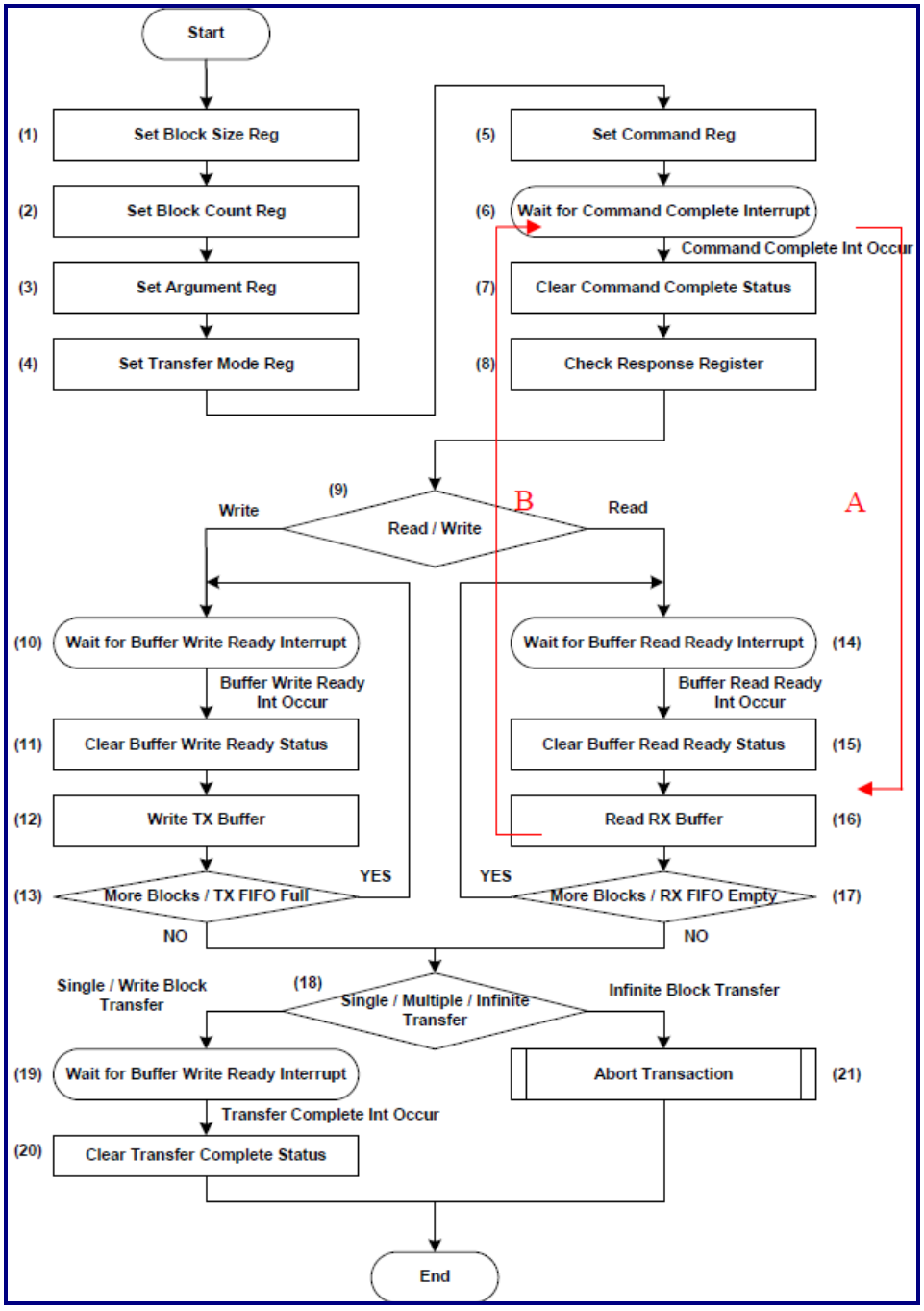
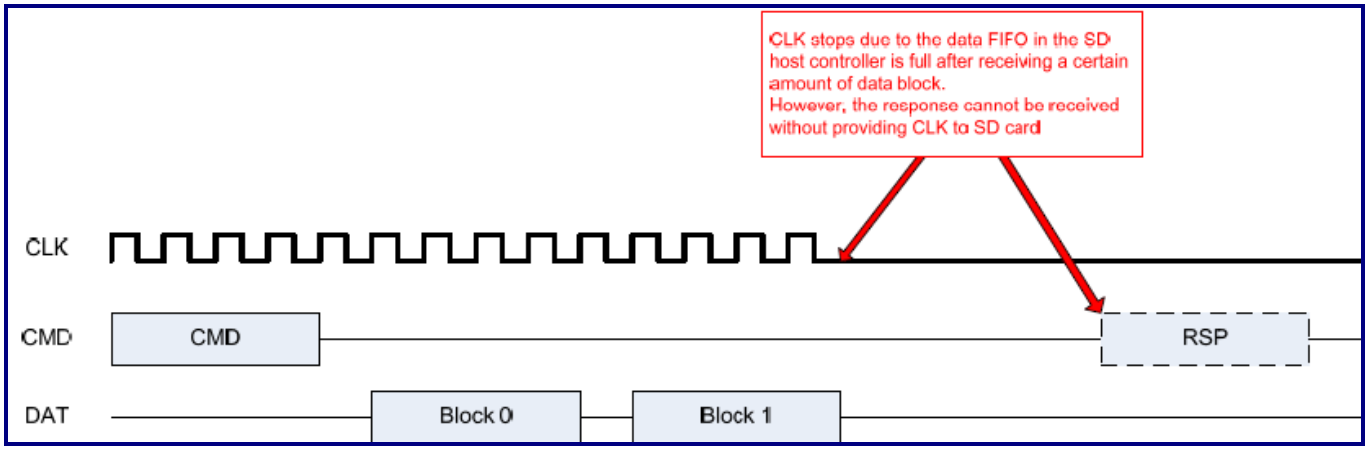


Figure 12-13. Data Transfer without DMA

Note: In some cases, the "Buffer Read Ready" interrupt may be asserted before the "Command Complete" interrupt. If the read data is not read through the data port in time, the data FIFO will be full and SDCLK will stop. This will lead to a command complete timeout because the response cannot be received if SDCLK stops.



The solution is to modify the "Not using DMA" sequence:

The buffer read ready interrupt should be serviced when it asserts even before the command complete asserts

For the read operation:

A. In Step 6, if the buffer read ready interrupt asserts and command complete interrupt does not assert, go to Step 15.

B. After finishing Step 16, if command complete is not received, go back to Step 6.

12.3.5.1.7 Data Transfer with SDMA

Figure 12-14 shows the flow of the data transfer with SDMA. Each step is executed as follows:

(1) Set the data address of the system memory to the SDMA system address register

(2) Set the length to **blk_size_r** in the block size register

(3) Set the count to **blk_cnt_r** in the block count register

(4) Set the argument to the argument register

(5) Set the transfer mode register

(6) Set the command register

(7) Wait for the **cmd_complete_r** interrupt

(8) When an interrupt occurs, the host driver will write '1' to clear the **cmd_complete_r** status in the normal interrupt status register.

(9) Check the response register to obtain information

(10) Wait for the **dma_int_r** interrupt when reaching the SDMA transfer boundary or wait for the **tran_complete_r** interrupt when the data transfer has been finished.

(11) If the **tran_complete_r** interrupt occurs, go to Step 14. If the **dma_int_r** interrupt occurs, go to Step 12. The **tran_complete_r** interrupt has higher priority than the **dma_int_r** interrupt.

(12) When the **dma_int_r** interrupt occurs, write '1' to clear **dma_int_r** in the normal interrupt status register.

(13) Set the next system address to the SDMA system address register, and go to Step 10.

(14) Write '1' to clear **tran_complete_r** and **dma_int_r** in the normal interrupt status register.

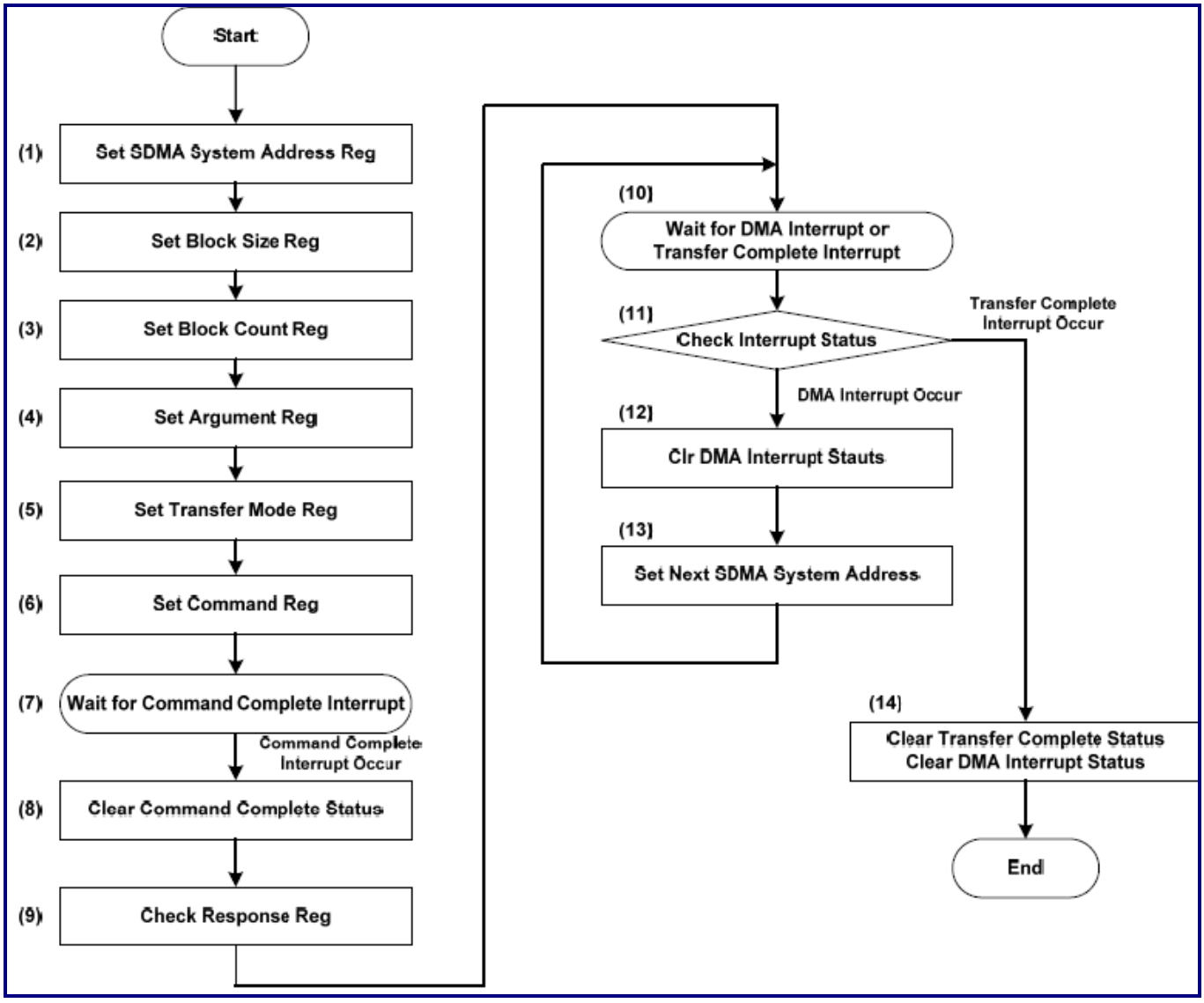


Figure 12-14. Data Transfer with SDMA

12.3.5.1.8 Data Transfer with ADMA

Figure 12-15 shows the flow of the data transfer with ADMA. Each step is executed as follows:

(1) Create a table for the ADMA descriptor in the system memory

(2) Set the address of the descriptor table to the ADMA system address register

(3) Set the length to **blk_size_r** in the block size register

(4) Set the count to **blk_cnt_r** in the block count register

(5) Set the argument to the argument register

(6) Set the transfer mode register

(7) Set the command register

(8) Wait for the **cmd_complete_r** interrupt

(9) When an interrupt occurs, the host driver will write '1' to clear the **cmd_complete_r** status in the normal interrupt status register.

(10) Check the response register to obtain information

(11) Wait for the **dma_err_r** interrupt in the Error interrupt status register when the ADMA error interrupt occurs or wait for the **tran_complete_r** interrupt when the data transfer is finished.

(12) If the **tran_complete_r** interrupt occurs, go to Step 13. If the **dma_err_r** interrupt occurs, go to Step 14.

(13)Write '1' to clear **tran_complete_r** in the normal interrupt status register

(14)Write '1' to clear **dma_err_r** in the error interrupt status register

(15) Abort the ADMA operation. The SD card should issue an abort command to stop the transfer. The host driver should check the ADMA error status register to understand how the ADMA error is generated.

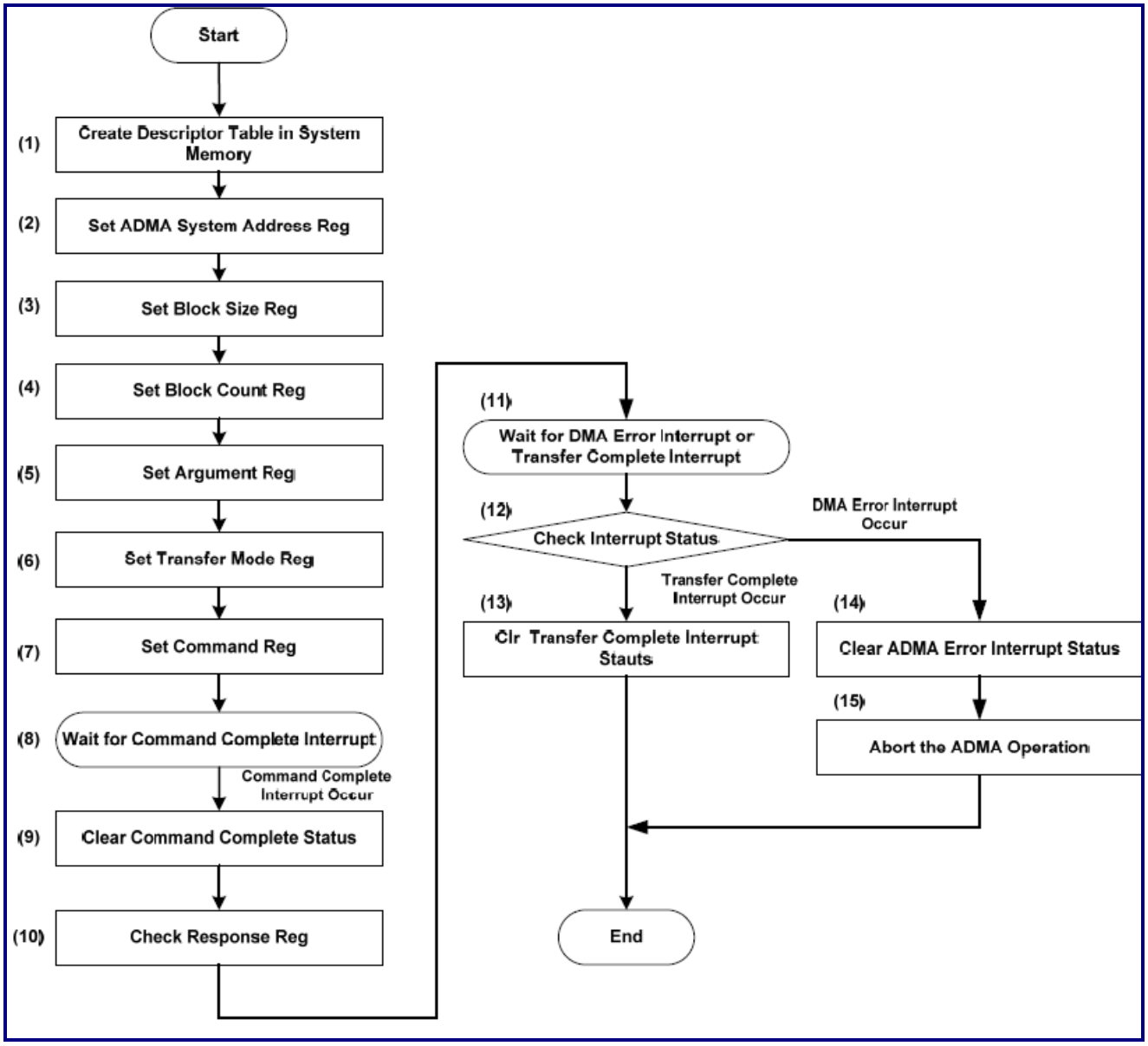


Figure 12-15. Data Transfer with ADMA

12.3.5.1.9 Abort Transaction

Users can operate an abort transaction by issuing the **CMD12** command for the SD card and the **CMD52** command for the SDIO card. The host driver can operate in two transaction stop types. The first type is to stop an infinite-block transfer, while the second type is to stop a multi-block transfer. These two abort transactions are described in the following subsections.

12.3.5.1.9.1 Asynchronous Abort

In the asynchronous abort, the host driver will issue an abort command at any time except when cmd_inhibit_c is set to '1'. The sequence of the asynchronous abort is shown in Figure 12-16. Each step is executed as follows:

(1) Issue an abort command according to Table 12-53

(2) Set **soft_rst_dat** and **soft_rst_cmd** to '1' in the software reset register for resetting the software

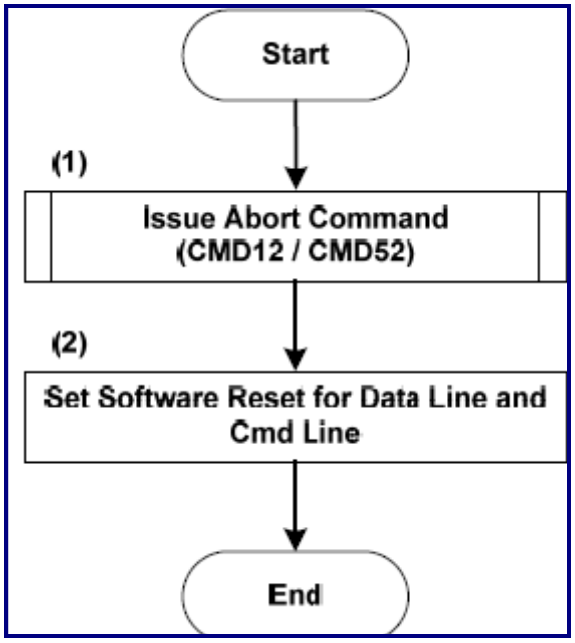


Figure 12-16. Asynchronous Abort Sequence

12.3.5.1.9.2 Synchronous Abort

In a synchronous abort, the host driver will issue an abort command after the data transfer, which is stopped by using **sp_blk_gap_req** in the block gap control register. The sequence of the synchronous abort is shown in Figure 12-17. Each step is executed as follows:

(1) Set **sp_blk_gap_req** to '1' to stop the SD transaction

(2) Wait the **tran_complete_r** interrupt

(3) When an interrupt occurs, write '1' to clear **tran_complete_r** in the normal interrupt status register

(4) Issue an abort command according to Table 12-53

(5) Set **soft_rst_dat** and **soft_rst_cmd** to '1' in the software reset register for resetting the software

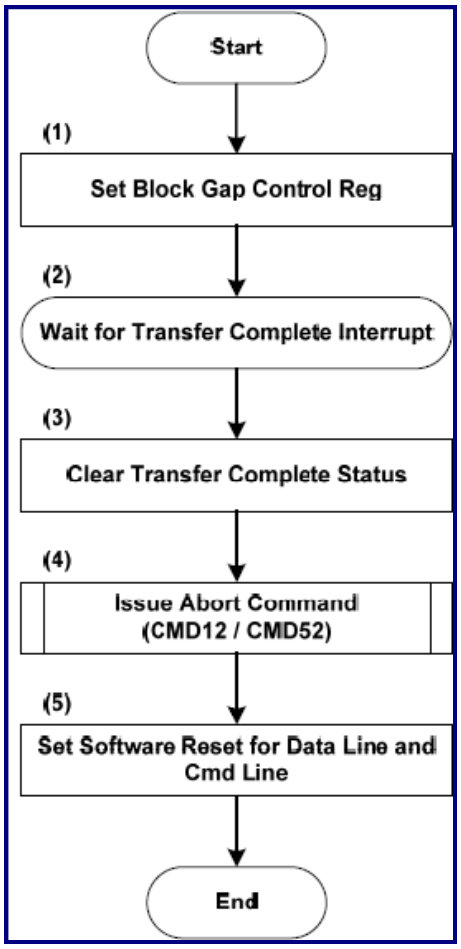


Figure 12-17. Synchronous Abort Sequence

12.3.5.1.10 Suspend/Resume Operation

If the host driver wants to perform the suspend/resume operation, the read wait function in the SDIO card must be activated. ADMA cannot support the suspend/resume function.

12.3.5.1.10.1 Suspend Operation

Figure 12-18 depicts the flow of the suspend operation. Each step is executed as follows:

(1) Set **sp_blk_gap_req** to '1' in the block gap control register to stop the SD transaction

(2) Wait for the interrupt. If **blk_gap_evt_r** is set to '0' and **tran_complete_r** is set to '1' in the normal interrupt status register, the transfer will be completed. Then, go to Step 8. If the **blk_gap_evt_r** interrupt occurs, go to Step 3.

(3) Write '1' to clear **blk_gap_evt_r** in the normal interrupt status register

(4) Wait for the **tran_complete_r** interrupt

(5) Issue the suspend command according to Table 12-53

(6) Check the **BS** response data. If **BS** is set to '0', the SD card will free the bus for the next operation. Then, go to Step 7.

(7) Save the registers (0x00 ~ 0x0D) for the recovery operation

(8) Write '1' to clear **tran_complete_r** in the normal interrupt status register

(9) Set **sp_blk_gap_req** to '0' in the block gap control register for cancelling the "stop" operation at the block gap request

(10) Check **BR** for the response data. If **BR** is set to '1', go to Step 11. If **BR** is set to '0', go the Step 13.

(11) Issue the **CMD52** command to cancel the previous "suspend" command according to Section 12.3.6.1.5.

(12) Check **BS** for the response data. If **BS** is set to '0', go to Step 7. If **BS** is set to '1', go to Step 13.

(13) Write '1' to clear **tran_complete_r** in the normal interrupt status register.

(14) Set **cont_req** to '1' in the block gap control register to restart the previous transaction. At the same time, set **sp_blk_gap_req** to '0'.

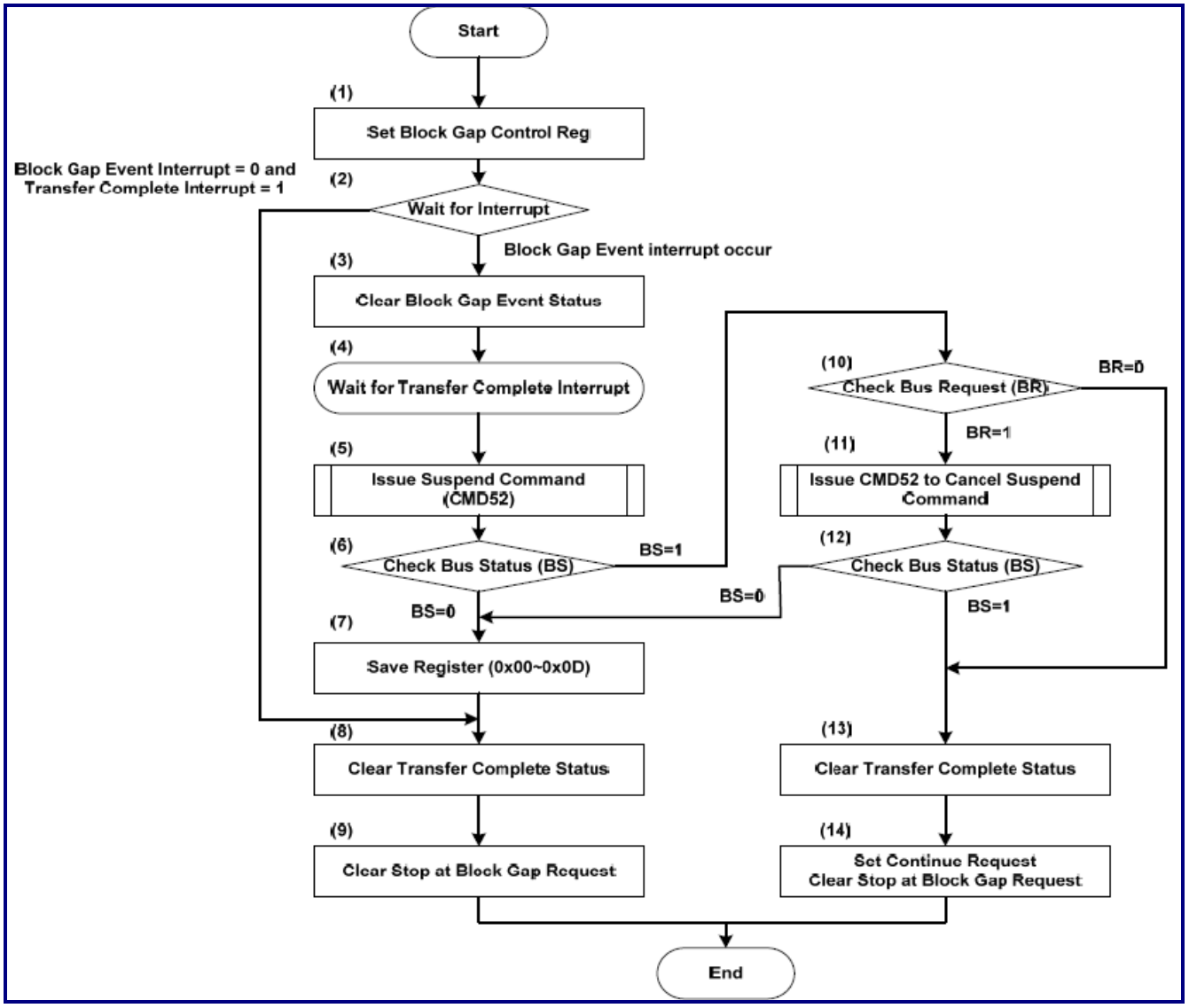


Figure 12-18. Suspend Operation

12.3.5.1.10.2 Resume Sequence

The resume sequence is shown in Figure 12-19. Each step is executed as follows:

(1) Restore registers (0x00 ~ 0x0D)

(2) Issue the resume command according to Table 12-53

(3) Check the **DF** for the response data. If **DF** is '0', no additional data will be transferred. If **DF** is '1', more data should be transferred after resume.

(4) Set **soft_rst_dat** to '1' in the software reset register to reset the data line.

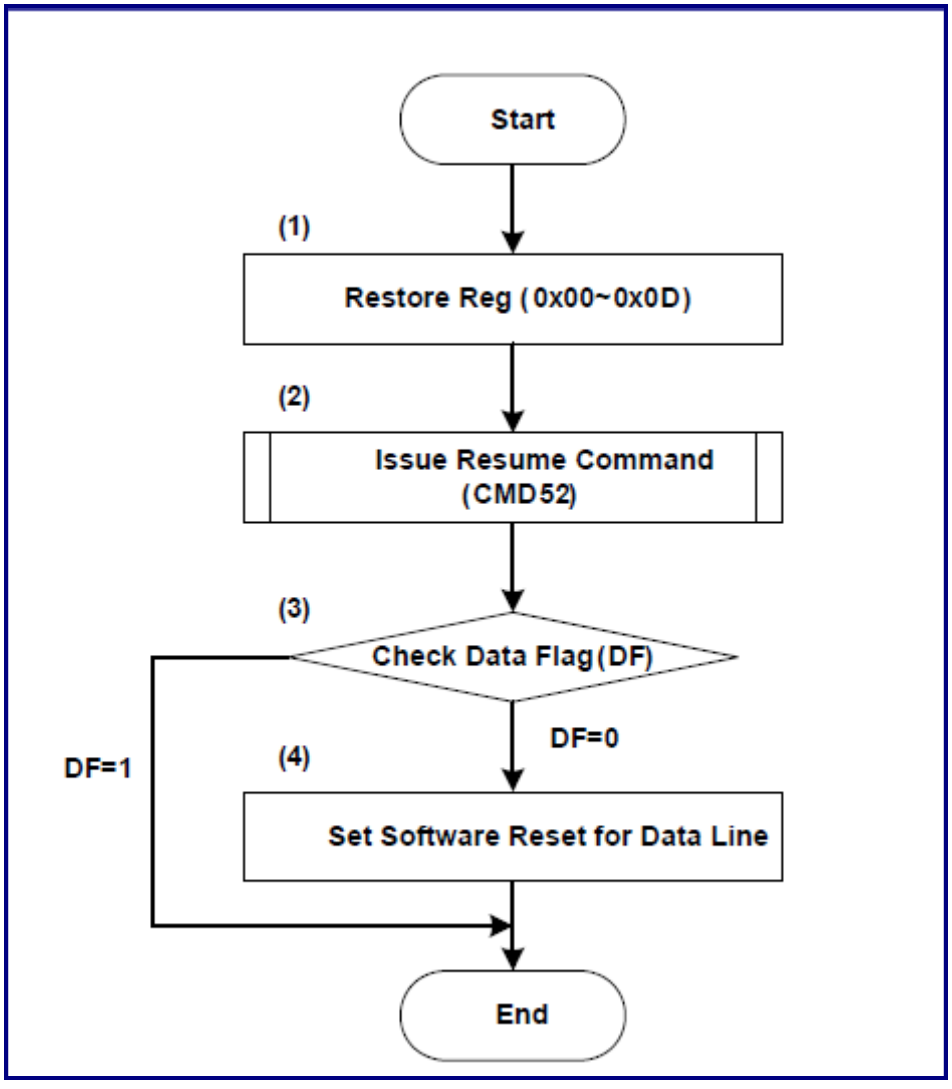


Figure 12-19. Resume Sequence

12.4 Universal Serial Bus (USB)

12.4.1 General Description

The USB OTG 2.0 controller can play a dual-role that can act as a host controller or as a peripheral controller. When it acts as a host controller, it contains a USB host controller that supports transactions at all speeds. Without the software intervention, the host controller can deal with the transaction-based data structure to offload CPU and automatically transmit and receive data on the USB bus. When it acts as a peripheral controller, each endpoint, except for the endpoint 0, will accept the programmable HS/FS transfer types to provide a flexibility to suit all applications. In addition, complying with the OTG standards means that both Session Request Protocol (SRP) and Host Negotiation Protocol (HNP) are supported. The USB 1.1 host controller has the same memory map and register definition as the USB OTG 2.0 controller.

12.4.2 Features

- Compliant with USB Specification, Revision 2.0
- Compliant with On-The-Go supplement to USB 2.0 Specification, Revision 1.0
- Supports UTMI+ Level 3 compliant transceiver
- Compatible with EHCI 1.0
- Built-in PVCI-compatible system bus and optional AMBA AHB bus interface
- Supports OTG SRP and HNP protocols
- Supports point-to-point communications with one high-speed, full-speed, or low-speed device
- Supports high-speed or full-speed hub
- Hardware configurable endpoints as HS/FS device
- Both host and device support isochronous/interrupt/control/bulk transfers
- Compatible with EHCI data structures (FSTN and SITD back points are not supported.)
- Supports embedded DMA access to FIFO
- Supports SRAM interface for FIFO
- Supports suspend, remote wake-up, and resume functions

12.4.3 Memory Map/Register Definition

The notations listed below are used to describe the access types of the registers.

Register Type	Description
RO	**Read-Only:** Register bits are read-only and may not be altered by software. Writing to this register will have no effect.
RW	**Read-Write:** Register bits are read-write and may be either set or cleared by software. Please note that some individual bits in the read/write registers may be Read-Only.
RW1C	**Write-1-to-Clear status:** Register bits indicate status when read, a set bit indicating a status event may be cleared by writing a 1. Writing a 0 to RW1C bits has no effect.
RW1S	**Write-1-to-Set status:** Register bits indicate status when read, a clear bit may be set by writing a '1'. Writing a '0' to RW1S bits has no effect.
ROS	**Sticky-Read-Only:** Register bits are read-only and may not be altered by software. Sticky registers are not initialized or modified by IP Hardware Reset.
RWS	**Sticky-Read-Write:** Register bits are read-write and may be either set or cleared by software to the desired state. Sticky registers are not initialized or modified by IP Hardware Reset.
RW1CS	**Sticky-Write-1-to-Clear status:** Register bits indicate status when read, a set bit indicating s status event may be cleared by writing a '1'. Writing a '0' to RW1CS bits has no effect. Sticky registers are not initialized or modified by IP Hardware Reset.
Rsvd	**Reserved:** Reserved for future RO implementations. Register bits shall be treated as read-only by system software. Rsvd register bits return '0' when read. Software shall ignore the value read from these bits.
RsvdO	**Reserved and Opaque:** Reserved for exclusive use by the controller. Register values may be modified by controller at any time. Software manipulation of this space may cause undetermined results. Software shall not write to this space unless explicitly allowed by instruction.
RsvdP	**Reserved and Preserved:** Reserved for future RW implementations. Software shall preserve the value read for writes to these bits.
RsvdZ	**Reserved and Zero:** Reserved for future RW1C implementations. Software shall use '0' for writes to these bits.
RC	**Read Clears all:** A read of this register will clear the current field value. Writing has no effect.

Register Type	Description
HwInit	**Hardware Initialized:** Register bits are initialized by firmware or hardware mechanisms. Bits are read-only after initialization and may only be reset with Hardware Reset

For the device controller, there are three reset conditions of the default values.

Default reset	Power-On Reset or Hardware Reset Each register will be reset to its default value.
(U)	Bus Reset USB bus reset. Not all registers can use bus reset.
(S)	Soft Reset By setting bit 4 of the Main Control Register (0x100) of the device, software can reset the SOF Frame Number Register (0x10C) and SOF Mask Timer Register (0x110).

12.4.3.1 Register Summary

Table 12-101. Register Summary

Name	Offset Range	Size (Byte)	Description
HC registers	0x000 ~ 0x07F	128	Host controller register space
OTG registers	0x080 ~ 0x0BF	64	OTG controller register space
Global registers	0x0C0 ~ 0x0FF	64	Global register space
Device registers	0x100 ~ 0x1FF	256	Device controller register space

12.4.3.2 General Registers and Functions

Table 12-102. OTG 0(OTG 2.0) Memory-mapped Registers

Offset (Hex)	Name	Size (Byte)	Type	Description	Default Value
000	HCCAP	4	RO, Rsvd	HC Capability Register	0x0100_0010
004	HCSPARAMS	4	RO, Rsvd	HC Structural Parameters	0x0000_0001
008	HCCPARAMS	4	RO, Rsvd	HC Capability Parameters	0x0000_0006
010	USBCMD	4	RW, Rsvd	HC USB Command Register	0
014	USBSTS	4	RO, RW1C, Rsvd	HC USB Status Register	0x0008_0B00
018	USBINTR	4	RW, Rsvd	HC USB Interrupt Enable Register	0x0000_1000
01C	FRINDEX	4	RW, Rsvd	HC Frame Index Register	0
024	PERIODICLISTBASE	4	RW, Rsvd	HC Periodic Frame List Base Address Register	Undefined
028	ASYNCLISTADDR	4	RW, Rsvd	HC Current Asynchronous List Address Register	Undefined
030	PORTSC	4	RO, RW, RW1C, Rsvd	HC Port Status and Control Register	0
040	HCMISC	4	RW, Rsvd	HC Miscellaneous Register	0x0000_1000
044	FS_EOF	4	RW, Rsvd	HC Full-Speed (FS) EOF1 Timing Point Register	0
048	HS_EOF	4	RW, Rsvd	HC High-Speed (HS) EOF1 Timing Point Register	0
080	OTG_CSR	4	RO, RW, Rsvd	OTG Control Status Register	0x0006_0020
084	OTG_ISR	4	RW1C, Rsvd	OTG Interrupt Status Register	0x0000_0040
088	OTG_IER	4	RW, Rsvd	OTG Interrupt Enable Register	0
0C0	GLB_ISR	4	RW1C, Rsvd	Global HC/OTG/DEV Interrupt Status Register	0
0C4	GLB_INT	4	RW, Rsvd	Global Mask of HC/OTG/DEV Interrupt Register	0
100	DEV_CTL	4	RO, RW, Rsvd	Device Main Control Register	0x0000_0420
104	DEV_ADR	4	RW, Rsvd	Device Address Register	0
108	DEV_TST	4	RW, RW1C, Rsvd	Device Test Register	0

Offset (Hex)	Name	Size (Byte)	Type	Description	Default Value
10C	DEV_SFN	4	RO, Rsvd	Device SOF Frame Number Register	0
110	DEV_SMT	4	RW, Rsvd	Device SOF Mask Timer Register	0x0000_044C
114	PHY_TST	4	RW, Rsvd	PHY Test Mode Selector Register	0
118	DEV_VCTL	4	RW, Rsvd	Device Vendor-Specific I/O Control Register	0
11C	DEV_CXCFG	4	RO, Rsvd	Device CX Configuration Status Register	HwInit
120	DEV_CXCFE	4	RO, RW, Rsvd	Device CX Configuration and FIFO Empty Status Register	0x0000_0F20
124	DEV_ICR	4	RW, Rsvd	Device Idle Counter Register	0
130	DEV_MIGR	4	RW, Rsvd	Device Mask of Interrupt Group Register	0
134	DEV_MISG0	4	RW, Rsvd	Device Mask of Interrupt Source Group 0 Register	0
138	DEV_MISG1	4	RW, Rsvd	Device Mask of Interrupt Source Group 1 Register	0x000F_00FF
13C	DEV_MISG2	4	RW, Rsvd	Device Mask of Interrupt Source Group 2 Register	0
140	DEV_IGR	4	RO, Rsvd	Device Interrupt Group Register	0x0000_0004
144	DEV_ISG0	4	RO, RW1C, Rsvd	Device Interrupt Source Group 0 Register	0
148	DEV_ISG1	4	RO, Rsvd	Device Interrupt Source Group 1 Register	0
14C	DEV_ISG2	4	RO, RW1C, Rsvd	Device Interrupt Source Group 2 Register	0x0000_0600
150	DEV_RXZ	4	RW, Rsvd	Device Receive Zero-Length Data Packet Register	0
154	DEV_TXZ	4	RW, Rsvd	Device Transfer Zero-length Data Packet Register	0
158	DEV_ISE	4	RW, Rsvd	Device Isochronous Sequential Error/Abort Register	0
160+ (n-1)*010	DEV_INMPS	4	RW, Rsvd	Device IN Endpoint n MaxPacketSize Register (n = 1 ~ 8)	0x0000_0200

Offset (Hex)	Name	Size (Byte)	Type	Description	Default Value
180+ (n-1)*010	DEV_OUTMPS	4	RW, Rsvd	Device OUT Endpoint n MaxPacketSize Register (n = 1 ~ 8)	0x0000_0200
1A0	DEV_EPMAP0	4	RW, Rsvd	Device Endpoint 1 ~ 4 Map Register	0xFFFF_FFFF
1A4	DEV_EPMAP1	4	RW, Rsvd	Device Endpoint 5 ~ 8 Map Register	0xFFFF_FFFF
1A8	DEV_FMAP	4	RW, Rsvd	Device FIFO Map Register	0x0F0F_0F0F
1AC	DEV_FCFG	4	RW, Rsvd	Device FIFO Configuration Register	0
1B0+ n*010	DEV_FIBC	4	RO, RW, Rsvd	Device FIFO n Instruction and Byte Count Register (n = 0 ~ 3)	0
1C0	DMA_TFN	4	RW, Rsvd	Device DMA Target FIFO Number Register	0
1C4	DMA_CPS0	4	RW, Rsvd	Device DMA Controller Parameter Setting 0 Register	HwCfg
1C8	DMA_CPS1	4	RW, Rsvd	Device DMA Controller Parameter Setting 1 Register	HwCfg
1CC	DMA_CPS2	4	RW	Device DMA Controller Parameter Setting 2 Register	0
1D0	DMA_CPS3	4	RO	Device DMA Controller Parameter Setting 3 Register	0

12.4.3.2.1 Host Controller Registers (Offset = 0x000 ~ 0x07F)

12.4.3.2.2 HC Capability Register (HCCAP, Offset = 0x000)

Table 12-103. HC Capability Register (HCCAP, Offset = 0x000)

Bit	Name	Type	Default Value	Description
[31:16]	HCIVERSION	RO	0x0100	**Host Controller Interface Version Number** This is a 2-byte register containing a BCD encoding of the EHCI revision number supported by the host controller.
[15:8]	-	Rsvd	-	Reserved
[7:0]	CAPLENGTH	RO	0x10	**Capability Register Length** This register is used as an offset to add to the register base to find out the beginning of the operational register space.

12.4.3.2.3 HCSPARAMS - HC Structural Parameters (Offset = 0x004)

Table 12-104. HCSPARAMS - HC Structural Parameters (Offset = 0x004)

Bit	Name	Type	Default Value	Description
[31:4]	-	Rsvd	-	Reserved
[3:0]	N_PORTS	RO	0x1	**Number of Ports** This field specifies the number of the physical downstream ports implemented on the host controller.

12.4.3.2.4 HCCPARAMS - HC Capability Parameters (Offset = 0x008)

Table 12-105. HCCPARAMS - HC Capability Parameters (Offset = 0x008)

Bit	Name	Type	Default Value	Description
[31:3]	-	Rsvd	-	Reserved
2	ASYN_SCH_PARK_CAP	RO	1'b1	**Asynchronous Schedule Park Capability** The host controller supports the park feature for the high-speed queue heads in the asynchronous schedule. This feature can be disabled or enabled, and set to a specific level by using the Asynchronous Schedule Park Mode Enable and Asynchronous Schedule Park Mode Count fields in the USBCMD register.
1	PROG_FR_LIST_FLAG	RO	1'b1	**Programmable Frame List Flag** When this bit is set to 1b1, the system software will specify and use a smaller frame list to configure the host controller via the Frame List Size field of the USBCMD register. This requirement ensures that the frame list is always physically contiguous.
0	-	Rsvd	-	Reserved

12.4.3.2.5 USBCMD - HC USB Command Register (Offset = 0x010)

Table 12-106. USBCMD - HC USB Command Register (Offset = 0x010)

Bit	Name	Type	Default Value	Description
[31:24]	-	Rsvd	-	Reserved
[23:16]	INT_THRC	R/W	8'h08	**Interrupt Threshold Control** This field will be used by the system software to select the maximum rate when the host controller issues an interrupt. The valid values are defined as below:

Bit	Name	Type	Default Value	Description
				Value: Maximum High-Speed interrupt interval 00h: Reserved 01h: 1 micro frame 02h: 2 micro frames 04h: 4 micro frames 08h: 8 micro frames (Default, equals to 1ms) 10h: 16 micro frames (2ms) 20h: 32 micro frames (4ms) 40h: 64 micro frames (8ms) Note: In the Full-Speed mode, this field is reserved.
[15:12]	-	Rsvd	-	Reserved
11	ASYN_PK_EN	R/W	1'b1	**Asynchronous Schedule Park Mode Enable** The software uses this bit to enable or disable the Park mode. When this bit is set to '1', the park mode will be enabled.
10	-	Rsvd	-	Reserved
[9:8]	ASYN_PK_CNT	RW	2'b11	**Asynchronous Schedule Park Mode Count** This field contains the count of number of successive transactions allowed by the host controller to execute from a high-speed queue head on the asynchronous schedule.
7	-	Rsvd	-	Reserved
6	INT_OAAD	RW	1'b0	**Interrupt on Asynchronous Advance Doorbell** This bit will be used as a doorbell by the software to inform the host controller to issue an interrupt after an asynchronous schedule is advanced.
5	ASCH_EN	RW	1'b0	**Asynchronous Schedule Enable** This bit controls the host controller to skip processing the asynchronous schedule. 0: Do not process the asynchronous schedule 1: Use the ASYNCLISTADDR register to access the asynchronous schedule
4	PSCH_EN	R/W	1'b0	**Periodic Schedule Enable** This bit controls the host controller to skip processing the periodic schedule. 0: Do not process the periodic schedule 1: Use the PERIODICKISTBASE register to access the periodic schedule

Bit	Name	Type	Default Value	Description
[3:2]	FRL_SIZE	RW	2'b00	**Frame List Size** This field specifies the size of the frame list. 00: 1024 elements (4096bytes, default value) 01: 512 elements (2048bytes) 10: 256 elements (1024bytes) 11: Reserved
1	HC_RESET	RW	1'b0	**Host Controller Reset** This control bit is used by the software to reset the host controller.
0	RS	RW	1'b0	**Run/Stop** When this bit is set to '1b', the host controller will proceed with the execution of schedule. 0: Stop 1: Run

12.4.3.2.6 USBSTS - HC USB Status Register (Offset = 0x014)

Table 12-107. USBSTS - HC USB Status Register (Offset = 0x014)

Bit	Name	Type	Default Value	Description
[31:16]	-	Rsvd	-	Reserved
15	ASCH_STS	RO	1'b0	**Asynchronous Schedule Status** This bit reports the actual status of the asynchronous schedule.
14	PSCH_STS	RO	1'b0	**Periodic Schedule Status** This bit reports the actual status of the periodic schedule.
13	Reclamation	RO	1'b0	**Reclamation** This is a read-only status bit used to detect the empty asynchronous schedule.
12	HCHalted	RO	1'b1	**Host Controller Halted** This bit will be zero when the Run/Stop bit is set to '1'. The host controller will set this bit to 1b after the execution is stopped, as a result of setting the Run/Stop bit to 0b.
[11:6]	-	Rsvd	-	Reserved
5	INT_OAA	RW1C	1'b0	**Interrupt on Async. Advance** This bit indicates the interrupt assertion on Async. Advance Doorbell.
4	H_SYSERR	RW1C	1'b0	**Host System Error** The host controller will set this bit to '1' when errors occur during a host system access involving the host controller module.
3	FRL_ROL	RW1C	1'b0	**Frame List Rollover** The host controller sets this bit to '1' when the Frame List Index is changed from the maximum value to zero.
2	PO_CHG_DET	RW1C	1'b0	**Port Change Detect** The host controller sets this bit to '1' when a port has a change bit transiting from '0' to '1'. This bit is loaded with OR of all PORTSC change bits.
1	USBERR_INT	RW1C	1'b0	**USB Error Interrupt** The host controller sets this bit to '1' upon the completion of a USB transaction, resulting in an error condition.
0	USB_INT	RW1C	1'b0	**USB Interrupt** The host controller sets this bit to '1' upon the completion of a USB transaction.

12.4.3.2.7 USBINTR - HC USB Interrupt Enable Register (Offset = 0x018)

Table 12-108. USBINTR - HC USB Interrupt Enable Register (Offset = 0x018)

Bit	Name	Type	Default Value	Description
[31:6]	-	Rsvd	-	Reserved
5	INT_OAA_EN	RW	1'b0	**Interrupt on Async. Advance Enable** When this bit is '1' and the interrupt on Async. Advance bit in the USBSTS register is '1', the host controller will issue an interrupt at the next interrupt threshold.
4	H_SYSERR_EN	RW	1'b0	**Host System Error Enable** When this bit is '1' and the Host System Error Status bit in the USBSTS register is '1', the host controller will issue an interrupt.
3	FRL_ROL_EN	RW	1'b0	**Frame List Rollover Enable** When this bit is '1' and the Frame List Rollover bit in the USBSTS register is '1', the host controller will issue an interrupt.
2	PO_CHG_INT_EN	RW	1'b0	**Port Change Interrupt Enable** When this bit is '1' and the Port Change Detect bit in the USBSTS register is '1', the host controller will issue an interrupt.
1	USBERR_INT_EN	RW	1'b0	**USB Error Interrupt Enable** When this bit is'1' and the USBERRINT bit in the USBSTS register is '1', the host controller will issue an interrupt at the next interrupt threshold.
0	USB_INT_EN	RW	1'b0	**USB Interrupt Enable** When this bit is '1' and the USBINT bit in the USBSTS register is '1', the host controller will issue an interrupt at the next interrupt threshold.

12.4.3.2.8 FRINDEX - HC Frame Index Register (Offset = 0x01C)

Table 12-109. FRINDEX - HC Frame Index Register (Offset = 0x01C)

Bit	Name	Type	Default Value	Description
[31:14]	-	Rsvd	-	Reserved
[13:0]	FRINDEX	RW	14'b0	**Frame Index** This register is used by the host controller to index the frame in the periodic frame list. It updates the list every 125 microseconds. The HC Frame Index register cannot be written unless the host controller is at the Halted state.

12.4.3.2.9 PERIODICLISTBASE - HC Periodic Frame List Base Address Register (Offset = 0x024)

Table 12-110. PERIODICLISTBASE - HC Periodic Frame List Base Address Register (Offset = 0x024)

Bit	Name	Type	Default Value	Description
[31:12]	PERI_BASADR	RW	Undefined	**Periodic Frame List Base Address** This field contains the beginning address of the periodic frame list in the system memory. This field corresponds to the memory address signals of [31:12].
[11:0]	-	Rsvd	-	Reserved

12.4.3.2.10 ASYNCLISTADDR - HC Current Asynchronous List Address Register (Offset = 0x028)

Table 12-111. ASYNCLISTADDR - HC Current Asynchronous List Address Register (Offset = 0x028)

Bit	Name	Type	Default Value	Description
[31:5]	Async_ladr	RW	Undefined	**Current Asynchronous List Address** This field contains the address of the next asynchronous queue head to be executed. This field corresponds to the memory address signals of [31:5].
[4:0]	-	Rsvd	-	Reserved

12.4.3.2.11 PORTSC - HC Port Status and Control Register (Offset = 0x030)

Table 12-112. PORTSC - HC Port Status and Control Register (Offset = 0x030)

Bit	Name	Type	Default Value	Description
[31:21]	-	RO	11'b0	Reserved
20	HC_TST_PKDONE	RW	1'b0	**Data Transfer is Done for Test Packet in Host Port Test Control** Firmware has sent all test patterns to FIFO in the PHY test packet mode by writing '1' to this bit.

<table>
<tr><th>Bit</th><th>Name</th><th>Type</th><th>Default Value</th><th>Description</th></tr>
<tr><td>[19:16]</td><td>PORT_TEST</td><td>RW</td><td>4’b0000</td><td>Port Test Control
When this field is zero, the port will not operate in the test mode. A non-zero value indicates that the port is operating in the test mode and the specific test mode is indicated by a specific value. The encoding of the test mode bits are (0110b ~ 1111b are reserved):
<table><tr><th>Bit</th><th>Test Mode</th></tr><tr><td>0000b</td><td>Test mode is not enabled.</td></tr><tr><td>0001b</td><td>Test J_STATE</td></tr><tr><td>0010b</td><td>Test K-STATE</td></tr><tr><td>0011b</td><td>Test SE0_NAK</td></tr><tr><td>0100b</td><td>Test Packet</td></tr><tr><td>0101b</td><td>Test FORCE_ENABLE</td></tr></table>Note: When this signal is set to “0100b” (Test packet), the test packet must be filled into FIFO by DMA first, and then set HC_TST_PKDONE to ‘1’.</td></tr>
<tr><td>[15:12]</td><td>-</td><td>Rsvd</td><td>-</td><td>Reserved</td></tr>
<tr><td>[11:10]</td><td>LINE_STS</td><td>RO</td><td>2’bxx</td><td>Line State
These bits reflect the current logical levels of the D+ and D- signal lines.
<table><tr><th>Bits[11:0]</th><th>USB State</th></tr><tr><td>00b</td><td>SE0</td></tr><tr><td>10b</td><td>J state</td></tr><tr><td>01b</td><td>K state</td></tr><tr><td>11b</td><td>Undefined</td></tr></table></td></tr>
<tr><td>9</td><td>-</td><td>Rsvd</td><td>-</td><td>Reserved</td></tr>
<tr><td>8</td><td>PO_RESET</td><td>RW</td><td>1’b0</td><td>Port Reset
1: Port is reset.
0: Port is not reset.
When software writes ‘1’ to this bit, the bus reset sequence defined in the USB specification will start. Software writes ‘0’ to this bit to terminate the bus reset sequence. Software must keep this bit to ‘1’ long enough to ensure that the reset sequence completes.
Note: Before setting this bit, the RUN/STOP bit should be set to ‘0’.</td></tr>
</table>

Bit	Name	Type	Default Value	Description
7	PO_SUSP	RW	1'b0	**Port Suspend** 1: Port is at the suspend state. 0: Port is not at the suspend state. The PP_EN bit and the PO_SUSP bit of this register define the port states as follows: **Bits[Port Enable, Suspend]** — **Port State** 0X — Disable 10 — Enable 11 — Suspend When at the suspend state, the downstream data propagation will be blocked on this port; except for the port reset. At the suspend state, the port will be sensitive to resume the detection. The host controller will ignore a zero written to this bit. The host controller will unconditionally set this bit to '0' under the following conditions: • Software sets the Force Port Resume bit to '0' (From '1'). • Software sets the Port Reset bit to '1' (From '0'). Note: Before setting this bit, the RUN/STOP bit should be set to '0'.
6	F_PO_RESM	RW	1'b0	**Force Port Resume** 1: Resume detected/driven on port 0: No resume detected/driven on port Software will set this bit to '1' to resume signaling. The host controller will set this bit to '1' if a J-to-K transition is detected when the port is at the suspend state. When this bit transits to '1' because a J-to-K transition is detected, the Port Change Detect bit in the USBSTS register will also be set to '1'.
[5:4]	-	Rsvd	-	Reserved
3	PO_EN_CHG	RW1C	1'b0	**Port Enable/Disable Change** 0: No change 1: Port enabled/disabled status is changed.
2	PO_EN	RW	1'b0	**Port Enable/Disable** 0: Disable 1: Enable Ports can only be enabled by the host controller as part of the reset and enable. Software cannot enable a port by writing '1' to this bit.

Bit	Name	Type	Default Value	Description
1	CONN_CHG	RW1C	1’b0	**Connect Status Change** 0: Do not change the current connect status 1: Change the current connect status This bit indicates that the current connect status is changed on the port.
0	CONN_STS	RO	1’b0	**Current Connect Status** 0: No device is presented. 1: Device is presented on the port. This bit reflects the current state of the port and may not directly correspond to the event that the connect status change bit is set. When TST_FORCEEN is set to ‘1’, this bit will be the output of u_hdiscon.

12.4.3.2.12 HC Miscellaneous Register (Offset = 0x040)

Table 12-113. HC Miscellaneous Register (Offset = 0x040)

Bit	Name	Type	Default Value	Description
[31:7]	-	Rsvd	-	Reserved
6	HostPhy_Suspend	RW	1’b0	**Host Transceiver Suspend Mode** Active high This bit places the transceiver in the suspend mode that draws the minimal power from the power supplies. This bit is only used in the host mode and is in the system clock domain instead of the UCLK clock domain.

Bit	Name	Type	Default Value	Description
[5:4]	EOF2_Time	RW	2'b0	**EOF 2 Timing Points** This field controls the EOF2 timing point before the next SOF.
				High-Speed EOF2 time
				00b 2 clocks (30MHz) = 66ns
				01b 4 clocks (30MHz) = 133ns
				10b 8 clocks (30MHz) = 266ns
				11b 16 clocks (30MHz) = 533ns
				Full-Speed EOF2 time
				00b 20 clocks (30MHz) = 666ns
				01b 40 clocks (30MHz) = 1.33µs
				10b 80 clocks (30MHz) = 2.66µs
				11b 160 clocks (30MHz) = 5.33µs
				Low-Speed EOF2 time
				00b 40 clocks (30MHz) = 1.33µs
				01b 80 clocks (30MHz) = 2.66µs
				10b 160 clocks (30MHz) = 5.33µs
				11b 320 clocks (30MHz) = 10.66µs
[3:2]	EOF1_Time	RW	2'b0	**EOF 1 Timing Points** This field controls the EOF1 timing point before the next SOF. This field should be adjusted according to the maximum packet size.
				High-Speed EOF1 time
				00b 540 clocks (30MHz) = 18µs
				01b 360 clocks (30MHz) = 12µs
				10b 180 clocks (30MHz) = 6µs
				11b 720 clocks (30MHz) = 24µs
				Full-Speed EOF1 time
				00b 1600 clocks (30MHz) = 53.3µs
				01b 1400 clocks (30MHz) = 46.6µs
				10b 1200 clocks (30MHz) = 40µs
				11b 21000 clocks (30MHz) = 700µs
				Low-Speed EOF1 time
				00b 3750 clocks (30MHz) = 125µs
				01b 3500 clocks (30MHz) = 116µs
				10b 3250 clocks (30MHz) = 108µs
				11b 4000 clocks (30MHz) = 133µs

Bit	Name	Type	Default Value	Description
[1:0]	ASYN_SCH_SLPT	RW	2'b01	**Asynchronous Schedule Sleep Timer** This field controls the sleep timer of the asynchronous schedule. 00b 5µs 01b 10µs 10b 15µs 11b 20µs

12.4.3.2.13 HC Full-Speed (FS) EOF1 Timing Point Register (Offset = 0x044)

This register is used to configure the Full-Speed (FS) EOF1 timing point before the next SOF, instead of using the register 0x040 bits[3:2]. When the bits in this register are non-zero, the EOF1 timing point will be set according to this register. Otherwise, this register will be disabled and the EOF1 timing point will be set according to the register 0x040 bits[3:2].

The formula of the Full-Speed (FS) EOF1 timing is showed as below:
FS EOF1 timing = (1ms - (125µs * Reg044[14:12] + (Clock period of UCLK) * Reg044[11:0]))

The bits should be adjusted according to the maximum packet size. Any improper configuration may cause the controller to hang.

Table 12-114. HC Full-Speed (FS) EOF1 Timing Point Register (Offset = 0x044)

Bit	Name	Type	Default Value	Description
[31:15]	-	Rsvd	-	Reserved
[14:12]	FS_EOF1_Time_125us	RW	3'b0	**Full-Speed EOF 1 Timing Points (Unit: 125µs)** The unit of these bits is 125microsecond.
[11:0]	FS_EOF1_Time	RW	12'b0	**Full-Speed EOF 1 Timing Points** The unit of these bits is one UCLK clock period. This value should not be greater than $\frac{125\mu s}{\text{UCLK period}}$.

12.4.3.2.14 HC High-Speed (HS) EOF1 Timing Point Register (Offset = 0x048)

The formula of the High-Speed (HS) EOF1 timing is showed as below:
HS EOF1 timing = (125μs - (Clock period of UCLK) * Reg044[11:0]))

The bits should be adjusted according to the maximum packet size. Any improper configuration may cause the controller to hang.

Table 12-115. HC High-Speed (HS) EOF1 Timing Point Register (Offset = 0x048)

Bit	Name	Type	Default Value	Description
[31:12]	-	Rsvd	-	Reserved
[11:0]	HS_EOF1_Time	RW	12'b0	**High-Speed EOF 1 Timing Points** This register is used to configure the HS EOF1 timing point before the next SOF, instead of using the register 0x040 bits[3:2]. If the bits in this register are non-zero, the EOF1 timing point will be set according to this register. Otherwise, this register will be disabled and the EOF1 timing point will be set according to the register 0x040 bits[3:2]. The unit of these bits is one UCLK clock period. This value should not be greater than $\frac{125\mu s}{\text{UCLK period}}$.

12.4.3.3 OTG Controller Registers (Offset = 0x080 ~ 0x0BF)

12.4.3.3.1 OTG Control Status Register (OTG_CSR, Offset = 0x080)

Table 12-116. OTG Control Status Register (OTG_CSR, Offset = 0x080)

Bit	Name	Type	Default Value	Description
[31:24]	-	Rsvd	-	Reserved and read as zeroes
[23:22]	HOST_SPD_TYP	RO	2'b00	**Host Speed Type** This field will indicate the speed type when the OTG device functions as a host. 2'b10: HS 2'b00: FS 2'b01: LS 2'b11: Reserved

Bit	Name	Type	Default Value	Description
21	ID	RO	1	**Current ID** This bit records the current ID of USB 2.0 OTG controller. 0: A-device 1: B-device
20	CROLE	RO	1	**Current Role** This bit records the current role of USB 2.0 OTG controller. 0: Host 1: Device
19	VBUS_VLD	RO	0	**A-device V_{bus} Valid** This bit indicates whether or not the voltage on V_{bus} is above the A-device V_{bus} valid threshold.
18	A_SESS_VLD	RO	0	**A-device Session Valid** This bit indicates whether or not the voltage on V_{bus} is above the A-device session valid threshold.
17	B_SESS_VLD	RO	0	**B-Device Session Valid** This bit indicates whether or not the voltage on V_{bus} is above the B-device session valid threshold.
16	B_SESS_END	RO	1	**B-Device Session End** This bit indicates whether or not the voltage on V_{bus} is below the B-device session end threshold.
[15:12]	-	RsvdP	4'b0	Reserved for internal testing only These bits should remain '0'.
11	HDISCON_FLT_SEL	R/W	0	This bit selects a timer to filter out the HDISCON noise from UTMI+. 0: Approximate to 135µs 1: Approximate to 270µs
10	VBUS_FLT_SEL	RW	0	This bit selects a timer to filter out the VBUS_VLD noise from UTMI+. 0: Approximate to 135µs 1: Approximate to 472µs
9	ID_FLT_SEL	RW	1'b0	This bit selects a timer to filter out the ID noise from UTMI+. 0: Approximate to 3ms 1: Approximate to 4ms
8	A_SRP_RESP_TYP	RW	0	**SRP Response Type** This bit determines the SRP type that the A-device should respond. 0: A-device responds to the V_{bus} pulsing. 1: A-device responds to the data-line pulsing.

Bit	Name	Type	Default Value	Description
7	A_SRP_DET_EN	RW	0	**A-device SRP Detection Enable** This bit controls whether or not the A-device should detect SRP from the B-device. 0: SRP detection is disabled. 1: SRP detection is enabled.
6	A_SET_B_HNP_EN	RW	0	This bit indicates the current role is A-device and the HNP function of B-device has been enabled. This bit should be set and cleared by the software. 0: No effect 1: HNP feature of B-device has been enabled. This bit will be valid when the current role is A-device. This bit will be cleared to '0' only after the A-device issues a USB reset.
5	A_BUS_DROP	RW	1	**A-device Bus Drop** This bit determines if the A-device wants to power-down V_{bus}. Writing this bit to '1' will clear BUS_REQ of A-device.
4	A_BUS_REQ	RW	0	**A-device Bus Request** This bit determines if A-device should control the bus. 0: A-device stops driving V_{bus} and bus traffic. 1: A-device drives V_{bus} and generates bus traffic.
3	-	Rsvd	-	Reserved and read as zeroes
2	B_DSCHRG_VBUS	RW	0	**B-device Discharge V_{bus}** This bit is used to determine if discharging V_{bus} is required after the V_{bus} pulsing during SRP. 0: V_{bus} will not be discharged after the V_{bus} pulsing. 1: V_{bus} will be discharged for 50ms after the V_{bus} pulsing.
1	B_HNP_EN	RW	0	This bit indicates that the B-device has been enabled to perform HNP. This bit will be cleared only after the B-device is reset by the host. 0: Disable HNP 1: Enable HNP
0	B_BUS_REQ	RW	0	**B-device Bus Request** This bit determines if the B-device should control the bus. After the SRP pulsing is finished, this bit will be automatically cleared by the hardware. 0: B-device stops driving V_{bus} and generating the bus traffic. 1: B-device requests to take control of bus.

12.4.3.3.2 OTG Interrupt Status Register (OTG_ISR, Offset = 0x084)

This register defines the interrupt status of USB 2.0 OTG controller. The interrupt status is not masked by interrupt enable. That is, if an event happens, the corresponding status bit will be set to '1' even if the corresponding interrupt enable bit is set to '0'.

Table 12-117. OTG Interrupt Status Register (OTG_ISR, Offset = 0x084)

Bit	Name	Type	Default Value	Description
[31:13]	-	Rsvd	-	Reserved and read as zeroes
12	APLGRMV	RW1C	0	**Mini-A Plug Remove** This bit will be set to '1' when the mini-A plug is removed. Writing '1' clears this bit while writing '0' has no effect.
11	BPLGRMV	RW1C	0	**Mini-B Plug Remove** This bit will be set to '1' when the mini-B plug is removed. Writing '1' clears this bit while writing '0' has no effect.
10	OVC	RW1C	0	**Over Current Detection** This bit will be set to '1' when V_{bus} is not reaching VBUS_VLD within the expected time. Writing '1' clears this bit while writing '0' has no effect. This bit will be valid only when the current role is A-device.
9	IDCHG	RW1C	0	**ID Change** This bit will be set to '1' when the current ID of USB 2.0 OTG controller changes from A-device to B-device or from B-device to A-device. Writing '1' clears this bit while writing '0' has no effect.
8	RLCHG	RW1C	0	**Role Change** This bit will be set to '1' when the current role of USB 2.0 OTG controller changes from host to peripheral or from peripheral to host. Writing '1' clears this bit while writing '0' has no effect.
7	-	Rsvd	-	Reserved
6	B_SESS_END	RW1C	1	**B_Sess_End** This bit will be set to '1' when B_SESS_END is high. Writing '1' clears this register while writing '0' has no effect.
5	A_VBUS_ERR	RW1C	0	**A-device V_{bus} Error** This bit will be set to '1' when the OTG state machine moves to the "VBUS_ERROR" state. Writing '1' clears this bit while writing '0' has no effect.

Bit	Name	Type	Default Value	Description
4	A_SRP_DET	RW1C	0	**A-device Detects SRP from B-device** This bit will be set to '1' when A-device detects SRP from B-device. Writing '1' clears this bit while writing '0' has no effect.
[3:1]	-	Rsvd	-	Reserved
0	B_SRP_DN	RW1C	0	**B-device SRP Done** This bit will be set to '1' after the B-device has completed the SRP signaling. Writing '1' clears this bit while writing '0' has no effect.

12.4.3.3.3 OTG Interrupt Enable Register (OTG_IER, Offset = 0x088)

This register defines the interrupt enable of an interrupt event. This register will not mask the interrupt status. It will only mask the interrupt generation.

Table 12-118. OTG Interrupt Enable Register (OTG_IER, Offset = 0x088)

Bit	Name	Type	Default Value	Description
[31:13]	-	Rsvd	-	Reserved
12	APLGRMV_EN	RW	0	APLGRMV interrupt enable
11	A_WAIT_CON_EN (HOV)	RW	0	A_WAIT_CON interrupt enable
10	OVC_EN (HOV)	RW	0	OVC interrupt enable
9	IDCHG_EN	RW	0	IDCHG interrupt enable
8	RLCHG_EN	RW	0	RLCHG interrupt enable
7	-	Rsvd	-	Reserved
6	B_SESS_END_EN (POV)	RW	0	B_SESS_END interrupt enable
5	A_VBUS_ERR_EN (HOV)	RW	0	A_VBUS_ERR interrupt enable
4	A_SRP_DET_EN	RW	0	A_SRP_DET interrupt enable
[3:1]	-	Rsvd	-	Reserved
0	B_SRP_DN_EN	RW	0	B_SRP_DN interrupt enable

12.4.3.4 Global Controller Registers (Offset = 0x0C0 ~ 0x0FF)

12.4.3.4.0 Global HC/OTG/DEV Interrupt Status Register (GLB_ISR, Offset = 0x0C0)

Table 12-119. Global HC/OTG/DEV Interrupt Status Register (GLB_ISR, Offset = 0x0C0)

Bit	Name	Type	Default Value	Description
[31:3]	-	Rsvd	-	Reserved
2	HC_INT	RW1C	1'b0	**HC Interrupt** This bit will be set to '1' when an interrupt is issued from the host controller block.
1	OTG_INT	RW1C	1'b0	**OTG Interrupt** This bit will be set to '1' when an interrupt is issued from the OTG controller block.
0	DEV_INT	RW1C	1'b0	**Device Interrupt** This bit will be set to '1' when an interrupt is issued from the device controller block.

12.4.3.4.1 Global Mask of HC/OTG/DEV Interrupt Register (GLB_INT, Offset = 0x0C4)

Table 12-120. Global Mask of HC/OTG/DEV Interrupt Register (GLB_INT, Offset = 0x0C4)

Bit	Name	Type	Default Value	Description
[31:4]	-	Rsvd	-	Reserved
3	INT_POLARITY	RW	1'b0	This bit controls the polarity of the system interrupt signal sys_int_n, the default is active low. 0: Active low (Default) 1: Active high
2	MHC_INT	RW	1'b0	This bit masks the interrupt bits of the HC Interrupt. 0: Enable the corresponding interrupt 1: Disable the corresponding interrupt
1	MOTG_INT	RW	1'b0	This bit masks the interrupt bits of the OTG Interrupt. 0: Enable the corresponding interrupt 1: Disable he corresponding interrupt
0	MDEV_INT	RW	1'b0	This bit masks the interrupt bits of the Device Interrupt. 0: Enable the corresponding interrupt 1: Disable the corresponding interrupt

12.4.3.5 Device Controller Registers (Offset = 0x100 ~ 0x1FF)

12.4.3.5.1 Device Main Control Register (DEV_CTL, Offset = 0x100)

Table 12-121. Device Main Control Register (DEV_CTL, Offset = 0x100)

Bit	Name	Type	Default Value	Description
[31:12]	-	Rsvd	-	Reserved
[11:10]	IDLE_DEGLITCH (HOV)	RW	2'b01	**Idle Line State Deglitch** The tolerance for glitches at the Idle line state can be configured by this register. During the Idle state, controller monitors the line state and determines the end-of-idle depends on different settings. 0b00: Treat a non-idle cycle as end-of-idle 0b01: Treat two continuous non-idle cycles as end-of-idle 0b10: Treat three continuous non-idle cycles as end-of-idle 0b11: Treat four continuous non-idle cycles as end-of-idle Please note that these bits should be carefully set according to the electrical instability due to cable.
9	FORCE_FS	RW	1'b0	**Force Device to Full-Speed** If this bit is set to 0b1, the high-speed negotiation will be disabled when the reset from USB host starts, which forces controller to be at full speed in the device mode. If this bit is 0b0, the controller will remain at high speed in the device mode. Please note that users can only modify this bit before the unplug bit is set; otherwise, it will cause unexpected results. The default value of this bit is set to '1' if the USB 1.1 wrapper configuration is enabled.
[8]	-	Rsvd	-	Reserved
7	SYSBUS_WIDTH (HOV)	RW	1'b0	**System Bus Width** 1: System bus width is 8 bits. 0: System bus width is 32 bits. Please note that setting this bit will directly affect the access time when AP needs to obtain the SETUP 8 bytes from SETUP_CMD_RPORT (0x1D0). If this bit is set to '1', AP shall access the register 0x1D0 eight times to get the whole SETUP 8 bytes. Otherwise, the command can be obtained by reading the register twice.
6	HS_EN (HOV)	RO	1'b0	**High-Speed Status** 1: Device is in the High-Speed mode. 0: Device is in the Full-Speed mode.

Bit	Name	Type	Default Value	Description
5	CHIP_EN (HOV)	RW	1'b1	**Chip Enable** Writing '1' will enable the write cycle of the FIFO controller. Please do not write '0' in the normal operation mode.
4	SFRST (HOV)	RW	1'b0	**Device Software Reset** Writing '1' will set the software-initiated reset to the controller device. This bit cannot be set when controller is in the suspend mode because u_clk is stopped. Setting this bit will de-assert the pw_save output if it is asserted. By setting this bit, the Chirp sequence will be terminated, the command FIFO will be cleared, and the Frame Number Register and the SOF Timer Mask Register will be cleared. Please note that the data FIFO status will not be cleared (Software reset will be self-cleared).
3	GOSUSP	RW	1'b0	**Go Suspend** Writing '1' will activate the suspend mode of PHY.
2	GLINT_EN (HOV)	RW	1'b0	**Global Interrupt Enable** Writing '1' will enable all interrupts. Individual interrupt can be masked by setting the corresponding bits in the interrupt mask register (Index 0x144 ~ 0x14C).
1	HALF_SPEED (HOV)	RW	1'b0	**Half Speed Enable** 1: The FIFO controller asserts ACK to DMA for every two clock cycles. 0: The FIFO controller asserts ACK to DMA continuously. This bit is set to '1' when SCLK is slower than 30MHz.
0	CAP_RMWAKUP	RW	1'b0	**Capability of Remote Wakeup** Writing '1' indicates that controller has the capability of being wakened up by the "wakeup" signal.

12.4.3.5.2 Device Address Register (DEV_ADR, Offset = 0x104)

Table 12-122. Device Address Register (DEV_ADR, Offset = 0x104)

Bit	Name	Type	Default Value	Description
[31:8]	-	Rsvd	-	Reserved
7	AFT_CONF	RW	1'b0 1'b0(U)	**After Set Configuration** Writing '1' indicates that the device has successfully executed the SET_CONFIGURATION command. The controller device will not respond to any non-control transfer before this bit is set.
[6:0]	DEVADR	RW	7'b0 7'b0(U)	**Device Address** This bit records the latest USB device address for each SET_ADDRESS.

12.4.3.5.3 Device Test Register (DEV_TST, Offset = 0x108)

Table 12-123. Device Test Register (DEV_TST, Offset = 0x108)

Bit	Name	Type	Default Value	Description
[31:7]	-	Rsvd	-	Reserved
6	DISGENSOF	RW	1'b0	**Disable the generation of SOF** Please always use the SOF issued by the host instead of the self-generation SOF.
5	TST_MOD (HOV)	RW	1'b0	**Test Mode** Writing '1' will turn this bit on. When this bit is set to '1', controller will enter the test mode. In the normal mode, the controller uses a counter for 10 ms to detect a USB reset. The count is a large number. In the test mode, the controller will use a smaller counter for the USB reset detection to save the test cycles on test machine.
4	TST_DISTOG (HOV)	RW	1'b0	**Disable Toggle Sequence** Writing '1' will disable the toggle sequence.
3	TST_DISCRC (HOV)	RW	1'b0	**Disable CRC** When setting this bit to '1', the controller will not append CRC for the upstream packets.
2	TST_CLREA	RW1C	1'b0	**Clear External Side Address** Writing '1' then a '0' will clear the external side address for the loop-back test (This bit is self-clear).
1	TST_LPCX	RW	1'b0	**Loop-back Test for Control Endpoint** Writing '1' indicates the loop-back test for control transfers.
0	TST_CLRFF (HOV)	RW	1'b0	**Clear FIFO** Writing '1' will clear all the FIFO counters and location counters of the PAM (This bit is self-clear).

12.4.3.5.4 Device SOF Frame Number Register (DEV_SFN, Offset = 0x10C)

Table 12-124. Device SOF Frame Number Register (DEV_SFN, Offset = 0x10C)

Bit	Name	Type	Default Value	Description
[31:14]	-	Rsvd	-	Reserved
[13:11]	USOFN	RO	3'b0	SOF Micro Frame Number Bits[2:0]
			3'b0 (S)	This field records the micro frame number of the high-speed mode.
[10:0]	SOFN	RO	11'b0	SOF Frame Number Bits[10:0]
			11'b0 (S)	This field records the frame number of the high-speed and full-speed modes.

12.4.3.5.5 Device SOF Mask Timer Register (DEV_SFN, Offset = 0x110)

Table 12-125. Device SOF Mask Timer Register (DEV_SFN, Offset = 0x110)

Bit	Name	Type	Default Value	Description
[31:16]	-	Rsvd	-	Reserved
[15:0]	SOFTM	R/W	16'h44C	**SOF Mask Timer**
			16'h44C (S)	Time count since the last SOF in the 30MHz clock bit.

12.4.3.5.6 PHY Test Mode Selector Register (PHY_TST, Offset = 0x114)

Table 12-126. PHY Test Mode Selector Register (PHY_TST, Offset = 0x114)

Bit	Name	Type	Default Value	Description
[31:5]	-	Rsvd	-	Reserved
4	TST_PKT	RW	1'b0	**Test Mode for Packet** Upon writing '1' to this bit, controller will repetitively send the packet defined in the UTMI specification to the transceiver. After the set_feature command shows the test mode and the index, Test_Packet, that has been decoded, this bit will be asserted.
3	TST_SE0NAK	RW	1'b0	Upon writing '1', the D+/D- lines will be set to HS, the quiescent state. The device will only respond to a valid HS IN token and will always respond to the IN token with NAK.
2	TST_KSTA	RW	1'b0	Upon writing '1', the D+/D- lines will be set to the high-speed K state.
1	TST_JSTA	RW	1'b0	Upon writing '1', the D+/D- lines will be set to the high-speed J state.

Bit	Name	Type	Default Value	Description
0	UNPLUG	RW	1’b1	When UNPLUG is set to ‘1’, the device controller will set PHY in the Non-Driving mode to emulate the detachment of a device even if it is really plugged. The USB host will not detect the plugging of a device. Such an event is called “soft-detachment”. After a hardware reset, the UNPLUG will be ‘1’ and the device will therefore be soft-detached. When the USB host detects the attachment of a device, the PHY must drive D+ and D- in the manner defined in the USB specification. In order to let the PHY drive D+ and D-, AP should clear UNPLUG after the hardware reset. If the AP does not clear the UNPLUG bit, the device will always be soft-detached and the USB host will never detect the attachment of the device.

12.4.3.5.7 Device Vendor-Specific I/O Control Register (DEV_VCTL, Offset = 0x118)

Table 12-127. Device Vendor-Specific I/O Control Register (DEV_VCTL, Offset = 0x118)

Bit	Name	Type	Default Value	Description
[31:6]	-	Rsvd	-	Reserved
5	VCTLOAD_N	RW	1’b1	**Vendor-Specific Test Mode Control Load** This bit controls the active-low output, u_vctload_n, to PHY. Writing ‘1’ to this bit sets the u_vctload_n output to ‘1’. When this bit is cleared, the u_vctload_n outputs will be set to ‘0’.
[4:0]	VCTL	RW	5’b0	**Vendor-Specific Test Mode Control** The programmed value is delivered to PHY via the output, “u_vctl”.

12.4.3.5.8 Device CX Configuration Status Register (DEV_CXCFG, Offset = 0x11C)

Table 12-128. Device CX Configuration Status Register (DEV_CXCFG, Offset = 0x11C)

Bit	Name	Type	Default Value	Description
[31:8]	-	Rsvd	-	Reserved
[7:0]	VSTA (HOV)	RO	HwInit	**Vendor-Specific Test Mode Status**

12.4.3.5.9 Device CX Configuration and FIFO Empty Status Register (DEV_CXCFE, Offset = 0x120)

Table 12-129. Device CX Configuration and FIFO Empty Status Register (DEV_CXCFE, Offset = 0x120)

Bit	Name	Type	Default Value	Description
31	-	Rsvd	-	Reserved
[30:24]	CX_FNT	RO	7b0	**CX FIFO Byte Count**
[23:12]	-	Rsvd	-	Reserved
[11:8]	F_EMP	RO	4b1111	**FIFO EMPTY** These bits are for FW to check if FIFO is fully empty.
[7:6]	-	Rsvd	-	Reserved
5	CX_EMP	RO	1'b1	**CX FIFO Empty** '1' indicates that the endpoint 0 FIFO is empty.
4	CX_FUL	RO	1'b0	**CX FIFO Full** '1' indicates that the endpoint 0 FIFO is full.
3	CX_CLR	RW	1'b0	**Clear CX FIFO Data** Writing '1' will clear the data in the endpoint 0 FIFO. Please note that for endpoint 0, all data in FIFO will be cleared regardless of whether the previous SETUP or IN or OUT transaction has been completed or not (This bit is self-clear).
2	CX_STL	RW	1'b0 1'b0(U)	**Stall CX** Writing '1' to this bit can stall endpoint 0. The stall status will be cleared with the next setup transaction. This bit will be cleared automatically when the transaction of endpoint 0 is finished. Upon detecting the bus reset, the firmware should clear this bit. Note: When setting this bit, CX_DONE must be set in the same write operation.
1	TST_PKDONE	RW	1'b0	**Data Transfer is Done for Test Packet** The firmware has completely sent the whole test patterns to the endpoint 0 FIFO for a PHY test by writing '1' to this bit. This bit is cleared by a hardware reset.
0	CX_DONE	RW	1'b0 1'b0(U)	**Data Transfer Done for CX** The firmware has finished the whole packet transaction for endpoint 0 by writing '1' to this bit. This bit is cleared by a hardware reset or by a p_endcx or p_comfail internal signal.

12.4.3.5.10 Device Idle Counter Register (DEV_ICR, Offset = 0x124)

Table 12-130. Device Idle Counter Register (DEV_ICR, Offset = 0x124)

Bit	Name	Type	Default Value	Description
[31:3]	-	Rsvd	-	Reserved
[2:0]	IDLE_CNT	RW	3'b0	These bits control the timing delay from the time indicated in the GOSUSP bit of the main control register to the time when the device enters the suspend mode. The timing delay is denoted as t_{susp_delay} below. 0b000: t_{susp_delay} = 0 ms 0b001: t_{susp_delay} = 1 ms 0b010: t_{susp_delay} = 2 ms 0b011: t_{susp_delay} = 3 ms 0b100: t_{susp_delay} = 4 ms 0b101: t_{susp_delay} = 5 ms 0b110: t_{susp_delay} = 6 ms 0b111: t_{susp_delay} = 7 ms Note: USB 2.0 specifications define T_{SUSP} to mandate the device that should enter the suspend mode no later than 10 ms after D+/D-is continuously at the idle state. The firmware programmer should be cautious in programming the value of t_{susp_delay}.

12.4.3.5.11 Device Mask of Interrupt Group Register (DEV_MIGR, Offset = 0x130)

Table 12-131. Device Mask of Interrupt Group Register (DEV_MIGR, Offset = 0x130)

Bit	Name	Type	Default Value	Description
[31:3]	-	Rsvd	-	Reserved
2	MINT_G2	RW	1'b0	Mask of interrupt of source group 2 0: Enable the corresponding interrupt 1: Disable the corresponding interrupt
1	MINT_G1	RW	1'b0	Mask of interrupt of source group 1 0: Enable the corresponding interrupt 1: Disable the corresponding interrupt
0	MINT_G0	RW	1'b0	Mask of interrupt of source group 0 0: Enable the corresponding interrupt 1: Disable the corresponding interrupt

12.4.3.5.12 Device Mask of Interrupt Source Group 0 Register (DEV_MISG0, Offset = 0x134)

Table 12-132. Device Mask of Interrupt Source Group 0 Register (DEV_MISG0, Offset = 0x134)

Bit	Name	Type	Default Value	Description
[31:6]	-	Rsvd	-	Reserved
5	MCX_COMABORT_INT	RW	1'b0	Mask interrupt of the control transfer command abort 0: Enable the corresponding interrupt 1: Disable the corresponding interrupt
4	MCX_COMFAIL_INT	RW	1'b0	Mask interrupt of the host emits extra IN or OUT data interrupt bit 0: Enable the corresponding interrupt 1: Disable the corresponding interrupt
3	MCX_COMEND	RW	1'b0	Mask the host end of a command (Entering the status stage) interrupt bit 0: Enable the corresponding interrupt 1: Disable the corresponding interrupt
2	MCX_OUT_INT	RW	1'b0	Mask the interrupt bits of endpoint 0 for OUT 0: Enable the corresponding interrupt 1: Disable the corresponding interrupt
1	MCX_IN_INT	RW	1'b0	Mask the interrupt bits of endpoint 0 for IN 0: Enable the corresponding interrupt 1: Disable the corresponding interrupt
0	MCX_SETUP_INT	RW	1'b0	Mask endpoint 0 setup data received interrupt bit 0: Enable the corresponding interrupt 1: Disable the corresponding interrupt

12.4.3.5.13 Device Mask of Interrupt Source Group 1 Register (DEV_MISG1, Offset = 0x138)

Table 12-133. Device Mask of Interrupt Source Group 1 Register (DEV_MISG1, Offset = 0x138)

Bit	Name	Type	Default Value	Description
[31:20]	-	Rsvd	-	Reserved
19	MF3_IN_INT	RW	1'b1	**Mask the IN interrupt bits of FIFO 3** 0: Enable the corresponding interrupt 1: Disable the corresponding interrupt

Bit	Name	Type	Default Value	Description
18	MF2_IN_INT	RW	1'b1	**Mask the IN interrupt bits of FIFO 2** 0: Enable the corresponding interrupt 1: Disable the corresponding interrupt
17	MF1_IN_INT	RW	1'b1	**Mask the IN interrupt bits of FIFO 1** 0: Enable the corresponding interrupt 1: Disable the corresponding interrupt
16	MF0_IN_INT	RW	1'b1	**Mask the IN interrupt bits of FIFO 0** 0: Enable the corresponding interrupt 1: Disable the corresponding interrupt
[15:8]	-	Rsvd	-	Reserved
7	MF3_SPK_INT	RW	1'b1	**Mask the Short Packet interrupt of FIFO 3** 0: Enable the corresponding interrupt 1: Disable the corresponding interrupt
6	MF3_OUT_INT	RW	1'b1	**Mask the OUT interrupt of FIFO 3** 0: Enable the corresponding interrupt 1: Disable the corresponding interrupt
5	MF2_SPK_INT	RW	1'b1	**Mask the Short Packet interrupt of FIFO 2** 0: Enable the corresponding interrupt 1: Disable the corresponding interrupt
4	MF2_OUT_INT	RW	1'b1	**Mask the OUT interrupt of FIFO 2** 0: Enable the corresponding interrupt 1: Disable the corresponding interrupt
3	MF1_SPK_INT	RW	1'b1	**Mask the Short Packet interrupt of FIFO 1** 0: Enable the corresponding interrupt 1: Disable the corresponding interrupt
2	MF1_OUT_INT	RW	1'b1	**Mask the OUT interrupt of FIFO 1** 0: Enable the corresponding interrupt 1: Disable the corresponding interrupt
1	MF0_SPK_INT	RW	1'b1	**Mask the Short Packet interrupt of FIFO 0** 0: Enable the corresponding interrupt 1: Disable the corresponding interrupt
0	MF0_OUT_INT	RW	1'b1	**Mask the OUT interrupt of FIFO 0** 0: Enable the corresponding interrupt 1: Disable the corresponding interrupt

12.4.3.5.14 Device Mask of Interrupt Source Group 2 Register (DEV_MISG2, Offset = 0x13C)

Table 12-134. Device Mask of Interrupt Source Group 2 Register (DEV_MISG2, Offset = 0x13C)

Bit	Name	Type	Default Value	Description
[31:11]	-	Rsvd	-	Reserved
10	MDev_Wakeup_byVBUS	RW	1'b0	**Mask Dev_Wakeup_byVBUS interrupt** Mask active Dev_Wakeup_byVBUS interrupt bit 0: Enable the corresponding interrupt 1: Disable the corresponding interrupt
9	MDev_Idle (HOV)	RW	1'b0	**Mask Dev_Idle interrupt** Mask active the Dev_Idle interrupt bit 0: Enable the corresponding interrupt 1: Disable the corresponding interrupt
8	MDMA_ERROR (HOV)	RW	1'b0	**Mask DMA Error Interrupt** Mask active the DMA Error interrupt bit 0: Enable the corresponding interrupt 1: Disable the corresponding interrupt
7	MDMA_CMPLT (HOV)	RW	1'b0	**Mask DMA Completion Interrupt** Mask active the DMA Completion interrupt bit 0: Enable the corresponding interrupt 1: Disable the corresponding interrupt
6	MRX0BYTE_INT	RW	1'b0	**Mask Received Zero-length Data Packet Interrupt** Mask active Received Zero-length Data Packet interrupt bit 0: Enable the corresponding interrupt 1: Disable the corresponding interrupt
5	MTX0BYTE_INT	RW	1'b0	**Mask Transferred Zero-length Data Packet Interrupt** Mask active Transferred Zero-length Data Packet interrupt bit 0: Enable the corresponding interrupt 1: Disable the corresponding interrupt
4	MSEQ_ABORT_INT	RW	1'b0	**Mask ISO Sequential Abort Interrupt** Mask active ISO Sequential Abort interrupt bit 0: Enable the corresponding interrupt 1: Disable the corresponding interrupt

Bit	Name	Type	Default Value	Description
3	MSEQ_ERR_INT	RW	1'b0	**Mask ISO Sequential Error Interrupt** Mask active Received ISO Sequential Error interrupt bit 0: Enable the corresponding interrupt 1: Disable the corresponding interrupt
2	MRESM_INT	RW	1'b0	**Mask Resume Interrupt** Mask active Resume State Change Interrupt bit 0: Enable the corresponding interrupt 1: Disable the corresponding interrupt
1	MSUSP_INT	RW	1'b0	**Mask Suspend Interrupt** Mask active Suspend State Change Interrupt bit 0: Enable the corresponding interrupt 1: Disable the corresponding interrupt
0	MUSBRST_INT	RW	1'b0	**Mask USB Reset Interrupt** Mask the Bus Reset Interrupt bit 0: Enable the corresponding interrupt 1: Disable the corresponding interrupt

12.4.3.5.15 Device Interrupt Group Register (DEV_IGR, Offset = 0x140)

Table 12-135. Device Interrupt Group Register (DEV_IGR, Offset = 0x140)

Bit	Name	Type	Default Value	Description
[31:3]	-	Rsvd	-	Reserved
2	INT_G2	RO	1'b0	This bit indicates interrupts occurred in Group 2.
1	INT_G1	RO	1'b0	This bit indicates interrupts occurred in Group 1.
0	INT_G0	RO	1'b0	This bit indicates interrupts occurred in Group 0.

12.4.3.5.16 Device Interrupt Source Group 0 Register (DEV_ISG0, Offset = 0x144)

Table 12-136. Device Interrupt Source Group 0 Register (DEV_ISG0, Offset = 0x144)

Bit	Name	Type	Default Value	Description
[31:6]	-	Rsvd	-	Reserved
5	CX_COMABT_INT	RW1C	1'b0	This bit indicates that a command abort event has happened. For the interrupts recorded in this source register, the command abort interrupt has the highest priority. For a command abort interrupt, the AP should only clear the CX_COMABT_INT bit. All other operations will be unnecessary and should be avoided. In general, the command abort interrupt will be accompanied by CX_SETUP_INT. AP should first serve the command abort interrupt to clear CX_COMABT_INT because CXF FIFO is frozen for AP to access when CX_COMABT_INT remains as '1'. In order to get 8-byte for SETUP for the command abort, AP should clear CX_COMABT_INT first.
4	CX_COMFAIL_INT	RO	1'b0	This bit indicates that the control transfer has been abnormally terminated. This bit will be asserted when the device receives extra IN/OUT token at the data stage of the control transfer. Once this bit is asserted, it will be kept at '1' before AP sets the CX_STL bit of the CX_Config_Status register. After setting the CX_STL bit of the CX_Config_Status register, the AP should set the CX_DONE bit of the CX_Config_Status register.
3	CX_COMEND_INT	RO	1'b0	This bit indicates that the control transfer has entered the status stage. This bit will remain asserted before the firmware sets the CX_DONE bit of the CX Configuration and Status Register (Address 0x120, bit 0).
2	CX_OUT_INT	RO	1'b0	This bit indicates that the control transfer contains valid data for the control-write transfers. This bit will remain asserted until the firmware starts to read data from control transfer FIFO (CXF) of device.

Bit	Name	Type	Default Value	Description
1	CX_IN_INT	RO	1'b0	This bit indicates that the firmware should write data for the control-read transfer to control transfer FIFO. For control transfer reads with length less than or equal to 64 bytes, this bit will never be asserted. The firmware will decode the 8-byte data sent at the SETUP stage of control transfer. The firmware should write the first payload of data into the control transfer FIFO if the 8-byte indicates the control-read transfer without asserting this bit. This bit will be asserted only when the length of the control-read transfer is greater than 64 bytes and the USB host has successfully received the data of previous packet. For example, for a 65-byte control-read transfer, the firmware should automatically write the first 64 bytes after it decodes eight byte of the SETUP data. The firmware will be interrupted to write the 65th byte when the USB host ACKs to the first 64 bytes. This bit will remain asserted until FW starts to write data into the control transfer FIFO of device.
0	CX_SETUP_INT	RO	1'b0	This bit will remain asserted until the firmware starts to read data from the control transfer FIFO of device.

12.4.3.5.17 Device Interrupt Source Group 1 Register (DEV_ISG1, Offset = 0x148)

Table 12-137. Device Interrupt Source Group 1 Register (DEV_ISG1, Offset = 0x148)

Bit	Name	Type	Default Value	Description
[31:20]	-	Rsvd	-	Reserved
19	F3_IN_INT	RO	1'b0	This bit becomes '1' to indicate that FIFO 3 is ready to be written. This bit will be cleared under the following two conditions: • A packet with the maximum size is received in FIFO 3. • End of the DMA transaction
18	F2_IN_INT	RO	1'b0	This bit becomes '1' to indicate that FIFO 2 is ready to be written. This bit will be cleared under the following two conditions: • A packet with the maximum size is received in FIFO 2. • End of the DMA transaction
17	F1_IN_INT	RO	1'b0	This bit becomes '1' to indicate that FIFO 1 is ready to be written. This bit is cleared under the following two conditions: • A packet with the maximum size is received in FIFO 1. • End of the DMA transaction

Bit	Name	Type	Default Value	Description
16	F0_IN_INT	RO	1'b0	This bit becomes '1' to indicate that FIFO 0 is ready to be written. This bit will be cleared under the following two conditions: • A packet with the maximum size is received in FIFO 0. • End of the DMA transaction
[15:8]	-	Rsvd	-	Reserved
7	F3_SPK_INT	RO	1'b0	This bit becomes '1' when short-packet data are received in FIFO 3. This bit will be cleared once the DMA master reads FIFO 3.
6	F3_OUT_INT	RO	1'b0	This bit becomes '1' when FIFO 3 is ready to be read. This bit will be cleared when all data in FIFO 3 are read out.
5	F2_SPK_INT	RO	1'b0	This bit becomes '1' when short-packet data are received in FIFO 2. This bit will be cleared when the DMA master reads FIFO 2.
4	F2_OUT_INT	RO	1'b0	This bit becomes '1' when FIFO 2 is ready to be read. This bit will be cleared when all data in FIFO 2 are read out.
3	F1_SPK_INT	RO	1'b0	This bit becomes '1' when short-packet data are received in FIFO 1. This bit will be cleared when the DMA master reads FIFO 1.
2	F1_OUT_INT	RO	1'b0	This bit becomes '1' when FIFO 1 is ready to be read. This bit will be cleared when all data in FIFO 1 are read out.
1	F0_SPK_INT	RO	1'b0	This bit becomes '1' when short-packet data are received in FIFO 0. This bit will be cleared when the DMA master reads FIFO 0.
0	F0_OUT_INT	RO	1'b0	This bit becomes '1' when FIFO 0 is ready to be read. This bit will be cleared when all data in FIFO 0 are read out.

12.4.3.5.18 Device Interrupt Source Group 2 Register (DEV_ISG2, Offset = 0x14C)

Table 12-138. Device Interrupt Source Group 2 Register (DEV_ISG2, Offset = 0x14C)

Bit	Name	Type	Default Value	Description
[31:11]	-	Rsvd	-	Reserved
10	Dev_Wakeup_byVBUS	RO	1'b0	**Device Wakeup by V_{bus} Interrupt** When the device is at the idle state and the V_{bus} signal is high, this bit will be set to 0b1.
9	Dev_Idle (HOV)	RO	1'b0	**Device Idle Interrupt** This bit can be activated under the following two conditions: ● Both A-device and B-device state machines are at the idle state. ● The SessEnd signal is high.
8	DMA_ERROR (HOV)	RW1C	1'b0	**DMA Error Interrupt** The DMA operation cannot be finished normally and an error signal has been received. When CPU initiates DMA to fill up or read out FIFO of the device, and the DMA controller receives an error response from the system bus, this bit will be set. This bit can only be cleared by the firmware. This bit is not affected by a USB bus reset.
7	DMA_CMPLT (HOV)	RW1C	1'b0	**DMA Completion Interrupt** The DMA operation has finished normally. When CPU initiates the DMA to fill up or read out FIFO of the device, this bit will be set after the mission is complete. This bit can only be cleared by the firmware. This bit is not affected by a USB bus reset.
6	RX0BYTE_INT	RW1C	1'b0	**Received Zero-length Data Packet Interrupt** The device receives a zero-length data packet from the USB host. This bit will be set when the device receives a zero-length data packet from the USB host. The firmware may further check the register 0x150 to determine which endpoint should receive the zero-length data packet from the USB host. When the interrupt occurs, the device will NAK the next OUT transaction to the same endpoint until the corresponding bit (In 0x150) is cleared by the firmware. This bit is not affected by a USB bus reset.

Bit	Name	Type	Default Value	Description
5	TX0BYTE_INT	RW1C	1'b0	**Transferred Zero-length Data Packet Interrupt** The device returns a zero-length data packet to the USB host. This bit will be set under the following two conditions: ● When the USB host issues an IN transaction to an isochronous endpoint and when the device is not ready to return the data, the device will transfer a zero-length data packet to the USB host. In such case, this bit will be set. ● When the TX0BYTE_IEPx bit is set, controller will return a zero-length data packet to the next IN transaction of the same endpoint after the endpoint data in FIFO are transferred. The firmware may further check the register 0x154 to determine which endpoint will return a zero-length data packet to the USB host. After AP served the interrupt request, this bit must be cleared by the firmware. This bit is not affected by a USB bus reset.
4	ISO_SEQ_ABORT_INT	RW1C	1'b0	**ISO Sequential Abort Interrupt** High bandwidth isochronous sequential abort When the device detects an incomplete DATA PID sequence during a micro-frame, this bit will be set. For example, the condition that a device detects an MDATA followed by an SOF will be regarded as "sequential abort". The firmware should further check the register 0x154 to determine which endpoint should receive the isochronous sequential abort. After AP serviced the interrupt request, this bit must be cleared by the firmware. This bit is not affected by a USB bus reset.
3	ISO_SEQ_ERR_INT	RW1C	1'b0	**Isochronous Sequential Error Interrupt** High bandwidth isochronous sequential error When a device detects a DATA PID sequence error of an isochronous transaction in the high bandwidth, this bit will be set. Any sequence that is out of order will be taken as a "sequence error". The firmware should further check the register 0x154 to determine which endpoint should receive the isochronous sequential error. After AP served the interrupt request, this bit must be cleared by the firmware. This bit is not affected by a USB bus reset.
2	RESM_INT	RW1C	1'b0	**Resume Interrupt** Resume-state-change interrupt bit When the device detects the resume event from the host, this bit will be set. After AP served the interrupt request, this bit must be cleared by the firmware. When a USB bus reset occurs, this bit will also be cleared.

Bit	Name	Type	Default Value	Description
1	SUSP_INT	RW1C	1'b0 1'b0 (U)	**Suspend Interrupt** Suspend-state-change interrupt bit When the USB bus remains in an idle state for over 3 ms, this bit will be set. This bit must be cleared before the firmware sets the "GOSUSP" bit of the 0x100 register. This bit will also be cleared when the USB bus reset or resume occurs.
0	USBRST_INT	RW1C	1'b0	**USB Reset Interrupt** Bus reset interrupt bit. When the device detects the USB bus reset from the host, this bit will be set. When AP serves the interrupt request, this bit must be cleared by the firmware.

12.4.3.5.19 Device Receive Zero-Length Data Packet Register (DEV_RXZ, Offset = 0x150)

All bits in this register are status bits that respond to the zero-length data packet received by an endpoint.

Table 12-139. Device Receive Zero-Length Data Packet Register (DEV_RXZ, Offset = 0x150)

Bit	Name	Type	Default Value	Description
[31:8]	-	Rsvd	-	Reserved
7	RX0BYTE_EP8	R/W	1'b0	Endpoint 8 receives a zero-length data packet.
6	RX0BYTE_EP7	R/W	1'b0	Endpoint 7 receives a zero-length data packet.
5	RX0BYTE_EP6	R/W	1'b0	Endpoint 6 receives a zero-length data packet.
4	RX0BYTE_EP5	R/W	1'b0	Endpoint 5 receives a zero-length data packet.
3	RX0BYTE_EP4	R/W	1'b0	Endpoint 4 receives a zero-length data packet.
2	RX0BYTE_EP3	R/W	1'b0	Endpoint 3 receives a zero-length data packet.
1	RX0BYTE_EP2	R/W	1'b0	Endpoint 2 receives a zero-length data packet.
0	RX0BYTE_EP1	R/W	1'b0	Endpoint 1 receives a zero-length data packet.

12.4.3.5.20 Device Transfer Zero-length Data Packet Register (DEV_TXZ, Offset = 0x154)

All bits in this register are status bits that respond to the zero-length data packet sent by an endpoint.

Table 12-140. Device Transfer Zero-Length Data Packet Register (DEV_TXZ, Offset = 0x154)

Bit	Name	Type	Default Value	Description
[31:8]	-	Rsvd	-	Reserved
7	TX0BYTE_EP8	R/W	1'b0	Endpoint 8 transfers zero-length data packet.
6	TX0BYTE_EP7	R/W	1'b0	Endpoint 7 transfers zero-length data packet.
5	TX0BYTE_EP6	R/W	1'b0	Endpoint 6 transfers zero-length data packet.
4	TX0BYTE_EP5	R/W	1'b0	Endpoint 5 transfers zero-length data packet.
3	TX0BYTE_EP4	R/W	1'b0	Endpoint 4 transfers zero-length data packet.
2	TX0BYTE_EP3	R/W	1'b0	Endpoint 3 transfers zero-length data packet.
1	TX0BYTE_EP2	R/W	1'b0	Endpoint 2 transfers zero-length data packet.
0	TX0BYTE_EP1	R/W	1'b0	Endpoint 1 transfers zero-length data packet.

12.4.3.5.21 Device Isochronous Sequential Error/Abort Register (DEV_ISE, Offset = 0x158)

All bits in this register are status bits that respond to the sequential error/abort occurred in an ISO endpoint.

Table 12-141. Device Isochronous Sequential Error/Abort Register (DEV_ISE, Offset = 0x158)

Bit	Name	Type	Default Value	Description
[31:24]	-	Rsvd	-	Reserved
23	ISO_SEQ_ERR_EP8	R/W	1'b0	Endpoint 8 encounters an isochronous sequential error.
22	ISO_SEQ_ERR_EP7	R/W	1'b0	Endpoint 7 encounters an isochronous sequential error.
21	ISO_SEQ_ERR_EP6	R/W	1'b0	Endpoint 6 encounters an isochronous sequential error.
20	ISO_SEQ_ERR_EP5	R/W	1'b0	Endpoint 5 encounters an isochronous sequential error.
19	ISO_SEQ_ERR_EP4	R/W	1'b0	Endpoint 4 encounters an isochronous sequential error.
18	ISO_SEQ_ERR_EP3	R/W	1'b0	Endpoint 3 encounters an isochronous sequential error.
17	ISO_SEQ_ERR_EP2	R/W	1'b0	Endpoint 2 encounters an isochronous sequential error.
16	ISO_SEQ_ERR_EP1	R/W	1'b0	Endpoint 1 encounters an isochronous sequential error.
[15:8]	-	Rsvd	-	Reserved
7	ISO_ABT_ERR_EP8	R/W	1'b0	Endpoint 8 encounters an isochronous sequential abort.
6	ISO_ABT_ERR_EP7	R/W	1'b0	Endpoint 7 encounters an isochronous sequential abort.
5	ISO_ABT_ERR_EP6	R/W	1'b0	Endpoint 6 encounters an isochronous sequential abort.

Bit	Name	Type	Default Value	Description
4	ISO_ABT_ERR_EP5	R/W	1'b0	Endpoint 5 encounters an isochronous sequential abort.
3	ISO_ABT_ERR_EP4	R/W	1'b0	Endpoint 4 encounters an isochronous sequential abort.
2	ISO_ABT_ERR_EP3	R/W	1'b0	Endpoint 3 encounters an isochronous sequential abort.
1	ISO_ABT_ERR_EP2	R/W	1'b0	Endpoint 2 encounters an isochronous sequential abort.
0	ISO_ABT_ERR_EP1	R/W	1'b0	Endpoint 1 encounters an isochronous sequential abort.

12.4.3.5.22 Device IN Endpoint x MaxPacketSize Register (One per Endpoint, *x* = 1 ~ 8) (DEV_INMPS, Offset = 0x160 + 4(*x* - 1))

Table 12-142. Device IN Endpoint x MaxPacketSize Register (One per Endpoint, *x* = 1 ~ 8) (DEV_INMPS, Offset = 0x160 + 4(*x* - 1))

Bit	Name	Type	Default Value	Description
[31:16]	-	Rsvd	-	Reserved
15	TX0BYTE_IEPx	RW	1'b0	**Transfer a Zero-length Data Packet from Endpoint x to USB Host** This bit should be set after the last packet of a transaction is sent to FIFO. After the endpoint data in FIFO are transferred, the device will return a zero-length data packet to the next IN transaction to the same endpoint. The AP should not send the next packet to the same endpoint until TX0BYTE_INT for the endpoint occurs. This bit will be automatically cleared by the hardware when TX0BYTE_INT occurs.
[14:13]	TX_NUM_HBW_IEPx	RW	2'b0	**Transaction Number of High Bandwidth Endpoint x** TX_NUM_HBW[1:0] (Only valid for the isochronous transfer) 00, 01: Indicate that the endpoint x is in non-high-bandwidth 10: Two transactions for each micro-frame 11: Three transactions for each micro-frame
12	RSTG_IEPx	RW	1'b0	**Reset Toggle Sequence for IN Endpoint x** The firmware resets the toggle bit of indexed endpoint x by writing '1' to this bit. This bit should also be cleared by the firmware.
11	STL_IEPx	RW	1'b0 1'b0(U)	**Stall IN Endpoint x** The indexed endpoint *x* can be stalled by writing '1' to this bit. The stalled status of the indexed endpoint *x* can be cleared by writing '0' to this bit. Before setting this bit, users should check the FIFO empty register (0x120) and make sure that the related FIFO is empty.

Bit	Name	Type	Default Value	Description
[10:0]	MAXPS_IEPx	RW	0x200	**Max Packet Size of IN Endpoint x** The maximum packet size of endpoint x capable of sending or receiving data that is smaller than or equal to this size. Note: This size must not exceed the FIFO size.

12.4.3.5.23 Device OUT Endpoint x MaxPacketSize Register (One per Endpoint, *x* = 1 ~ 8) (DEV_OUTMPS, Offset = 0x180 + 4(*x* - 1))

Table 12-143. Device OUT Endpoint x MaxPacketSize Register (One per Endpoint, *x* = 1 ~ 8) (DEV_OUTMPS, Offset = 0x180 + 4(*x* - 1))

Bit	Name	Type	Default Value	Description
[31:13]	-	Rsvd	-	Reserved
12	RSTG_OEPx	RW	1'b0	**Reset Toggle Sequence for OUT Endpoint x** The firmware resets the toggle bit of the indexed endpoint *x* by writing '1' to this bit. This bit should also be cleared by the firmware.
11	STL_OEPx	RW	1'b0	**Stall OUT Endpoint x** Writing '1' to STL_OEP will stall endpoint *x*. The stalled status of the indexed endpoint *x* can be cleared by writing '0' to this bit.
[10:0]	MAXPS_OEPx	RW	0x200	**Max Packet Size of OUT Endpoint x** The maximum packet size of endpoint *x* capable of sending or receiving data that is smaller than or equal to this size.

12.4.3.5.24 Device Endpoint 1 ~ 4 Map Register (DEV_EPMAP0, Offset = 0x1A0)

Table 12-144. Device Endpoint 1 ~ 4 Map Register (DEV_EPMAP0, Offset = 0x1A0)

Bit	Name	Type	Default Value	Description
[31:30]	-	Rsvd	-	Reserved
[29:28]	FNO_OEP4	RW	2'b11	**FIFO Number of OUT Endpoint 4** This register records the physical FIFO number of logical OUT endpoint 4.
[27:26]	-	Rsvd	-	Reserved
[25:24]	FNO_IEP4	RW	2'b11	**FIFO Number of IN Endpoint 4** This register records the physical FIFO number of logical IN endpoint 4.
[23:22]	-	Rsvd	-	Reserved

Bit	Name	Type	Default Value	Description
[21:20]	FNO_OEP3	RW	2'b11	**FIFO Number of OUT Endpoint 3** This register records the physical FIFO number of logical OUT endpoint 3.
[19:18]	-	Rsvd	-	Reserved
[17:16]	FNO_IEP3	RW	2'b11	**FIFO Number of IN Endpoint 3** This register records the physical FIFO number of logical IN endpoint 3.
[15:14]	-	Rsvd	-	Reserved
[13:12]	FNO_OEP2	RW	2'b11	**FIFO Number of OUT Endpoint 2** This register records the physical FIFO number of logical OUT endpoint 2.
[11:10]	-	Rsvd	-	Reserved
[9:8]	FNO_IEP2	RW	2'b11	**FIFO Number of IN Endpoint 2** This register records the physical FIFO number of logical IN endpoint 2.
[7:6]	-	Rsvd	-	Reserved
[5:4]	FNO_OEP1	RW	2'b11	**FIFO Number of OUT Endpoint 1** This register records the physical FIFO number of logical OUT endpoint 1.
[3:2]	-	Rsvd	-	Reserved
[1:0]	FNO_IEP1	RW	2'b11	**FIFO Number of IN Endpoint 1** This register records the physical FIFO number of logical IN endpoint 1.

12.4.3.5.25 Device Endpoint 5 ~ 8 Map Register (DEV_EPMAP1, Offset = 0x1A4)

Table 12-145. Device Endpoint 5 ~ 8 Map Register (DEV_EPMAP1, Offset = 0x1A4)

Bit	Name	Type	Default Value	Description
[31:30]	-	Rsvd	-	Reserved
[29:28]	FNO_OEP8	RW	2'b11	**FIFO Number of OUT Endpoint 8** This register records the physical FIFO number of logical OUT endpoint 8.
[27:26]	-	Rsvd	-	Reserved
[25:24]	FNO_IEP8	RW	2'b11	**FIFO Number of IN Endpoint 8** This register records the physical FIFO number of logical IN endpoint 8.

Bit	Name	Type	Default Value	Description
[23:22]	-	Rsvd	-	Reserved
[21:20]	FNO_OEP7	RW	2'b11	**FIFO Number of OUT Endpoint 7** This register records the physical FIFO number of logical OUT endpoint 7.
[19:18]	-	Rsvd	-	Reserved
[17:16]	FNO_IEP7	RW	2'b11	**FIFO Number of IN Endpoint 7** This register records the physical FIFO number of logical IN endpoint 7.
[15:14]	-	Rsvd	-	Reserved
[13:12]	FNO_OEP6	RW	2'b11	**FIFO Number of OUT Endpoint 6** This register records the physical FIFO number of logical OUT endpoint 6.
[11:10]	-	Rsvd	-	Reserved
[9:8]	FNO_IEP6	RW	2'b11	**FIFO Number of IN Endpoint 6** This register records the physical FIFO number of logical IN endpoint 6.
[7:6]	-	Rsvd	-	Reserved
[5:4]	FNO_OEP5	RW	2'b11	**FIFO Number of OUT Endpoint 5** This register records the physical FIFO number of logical OUT endpoint 5.
[3:2]	-	Rsvd	-	Reserved
[1:0]	FNO_IEP5	RW	2'b11	**FIFO Number of IN Endpoint 5** This register records the physical FIFO number of logical IN endpoint 5.

12.4.3.5.26 Device FIFO Map Register (DEV_FMAP, Offset = 0x1A8)

Table 12-146. Device FIFO Map Register (DEV_FMAP, Offset = 0x1A8)

Bit	Name	Type	Default Value	Description
[31:30]	-	Rsvd	-	Reserved
[29:28]	Dir_FIFO3	RW	2'b0	**FIFO 3 Direction** Data transfer direction 2'b00: Out 2'b01: In 2'b10: Bidirectional 2'b11: Not allowed

Bit	Name	Type	Default Value	Description
[27:24]	EPNO_FIFO3	RW	4'b1111	**Endpoint number of FIFO 3** This field records the physical endpoint number of physical FIFO 3.
[23:22]	-	Rsvd	-	Reserved
[21:20]	Dir_FIFO2	RW	2'b0	**FIFO 2 Direction** Data transfer direction 2'b00: Out 2'b01: In 2'b10: Bidirectional 2'b11: Not allowed
[19:16]	EPNO_FIFO2	RW	4'b1111	**Endpoint number of FIFO 2** This field records the physical endpoint number of physical FIFO 2.
[15:14]	-	Rsvd	-	Reserved
[13:12]	Dir_FIFO1	RW	2'b0	**FIFO 1 Direction** Data transfer direction 2'b00: Out 2'b01: In 2'b10: Bidirectional 2'b11: Not allowed
[11:8]	EPNO_FIFO1	RW	4'b1111	**Endpoint number of FIFO 1** This field records the physical endpoint number of physical FIFO 1.
[7:6]	-	Rsvd	-	Reserved
[5:4]	Dir_FIFO0	RW	2'b0	**FIFO 0 Direction** Data transfer direction 2'b00: Out 2'b01: In 2'b10: Bidirectional 2'b11: Not allowed
[3:0]	EPNO_FIFO0	RW	4'b1111	**Endpoint number of FIFO 0** This field records the physical endpoint number of physical FIFO 0.

12.4.3.5.27 Device FIFO Configuration Register (DEV_FCFG, Offset = 0x1AC)

Table 12-147. Device FIFO Configuration Register (DEV_FCFG, Offset = 0x1AC)

Bit	Name	Type	Default Value	Description
[31:30]	-	Rsvd	-	Reserved
29	EN_F3	RW	1'b0	**Enable FIFO 3** Writing '1' to this bit indicates that FIFO is enabled.
28	BLKSZ_F3	RW	1'b0	**Block Size of FIFO 3** BLKSIZE_F3 = 0: The maximum packet size of transferred packets is smaller than or equal to 512bytes. BLKSIZE_F3 = 1: The maximum packet size of transferred packets is smaller than or equal to 1024bytes and greater than 512bytes.
[27:26]	BLKNO_F3	RW	2'b0	**Block Number of FIFO 3** BLKNUM_F3 = 00: Single block BLKNUM_F3 = 01: Double blocks BLKNUM_F3 = 10: Triple blocks BLKNUM_F3 = 11: Reserved
[25:24]	BLK_TYP_F3	RW	2'b0	**Transfer Type of FIFO 3** This field indicates the transfer type used for the FIFOx transfer. TYP_F3 = 00: Reserved TYP_F3 = 01: Isochronous type TYP_F3 = 10: Bulk type TYP_F3 = 11: Interrupt type
[23:22]	-	Rsvd	-	Reserved
21	EN_F2	RW	1'b0	**Enable FIFO 2** A '1' indicates that the FIFO is enabled.
20	BLKSZ_F2	RW	1'b0	**Block Size of FIFO 2** BLKSIZE_F2 = 0: The maximum packet size of transferred packets is smaller than or equal to 512bytes. BLKSIZE_F2 = 1: The maximum packet size of transferred packets is smaller than or equal to 1024bytes and greater than 512bytes.
[19:18]	BLKNO_F2	RW	2'b0	**Block Number of FIFO 2** BLKNUM_F2 = 00: Single block BLKNUM_F2 = 01: Double blocks BLKNUM_F2 = 10: Triple blocks BLKNUM_F2 = 11: Reserved

Bit	Name	Type	Default Value	Description
[17:16]	BLK_TYP_F2	RW	2'b0	**Transfer Type of FIFO 2** This field indicates the transfer type used for the FIFO 2 transfer. TYP_F2 = 00: Reserved TYP_F2 = 01: Isochronous type TYP_F2 = 10: Bulk type TYP_F2 = 11: Interrupt type
[15:14]	-	Rsvd	-	Reserved
13	EN_F1	RW	1'b0	**Enable FIFO 1** '1' indicates that FIFO is enabled.
12	BLKSZ_F1	RW	1'b0	**Block Size of FIFO 1** BLKSIZE_F1: 0: The maximum packet size of transferred packets is smaller than or equal to 512bytes. BLKSIZE_F1: 1: The maximum packet size of transferred packets is smaller than or equal to 1024bytes and greater than 512bytes.
[11:10]	BLKNO_F1	RW	2'b0	**Block Number of FIFO 1** BLKNUM_F1: 00: Single block BLKNUM_F1: 01: Double blocks BLKNUM_F1: 10: Triple blocks BLKNUM_F1: 11: Reserved
[9:8]	BLK_TYP_F1	RW	2'b0	**Transfer Type of FIFO 1** This field indicates the transfer type used for the FIFO 1 transfer. TYP_F1: 00: Reserved TYP_F1: 01: Isochronous type TYP_F1: 10: Bulk type TYP_F1: 11: Interrupt type
[7:6]	-	Rsvd	-	Reserved
5	EN_F0	RW	1'b0	**Enable FIFO 0** '1' indicates that FIFO is enabled.
4	BLKSZ_F0	RW	1'b0	**Block Size of FIFO 0** BLKSIZE_F0: 0: The maximum packet size of transferred packets is smaller than or equal to 512 bytes. BLKSIZE_F0: 1: The maximum packet size of transferred packets is smaller than or equal to 1024 bytes and greater than 512 bytes.

Bit	Name	Type	Default Value	Description
[3:2]	BLKNO_F0	RW	2'b0	**Block Number of FIFO 0** BLKNUM_F0: 00: Single block BLKNUM_F0: 01: Double blocks BLKNUM_F0: 10: Triple blocks BLKNUM_F0: 11: Reserved
[1:0]	BLK_TYP_F0	RW	2'b0	**Transfer Type of FIFO 0** This field indicates the transfer type used for FIFO 0 transfer. TYP_F0: 00: Reserved TYP_F0: 01: Isochronous type TYP_F0: 10: Bulk type TYP_F0: 11: Interrupt type

12.4.3.5.28 Device FIFO x Instruction and Byte Count Register (One per FIFO, n = 0 ~ 3) (DEV_FIBC, Offset = 0x1B0 + 4n)

Table 12-148. Device FIFO x Instruction and Byte Count Register (One per FIFO, n = 0 ~ 3) (DEV_FIBC, Offset = 0x1B0 + 4n)

Bit	Name	Type	Default Value	Description
[31:13]	-	Rsvd	-	Reserved
12	FFRST	RW	1'b0	**FIFO x reset** When this bit is set, FIFO will be reset by firmware. This bit will be automatically cleared.
11	-	Rsvd	-	Reserved
[10:0]	BC_Fx	RO	11'b0	**OUT FIFO x Byte Count** BC_Fx[10:0] indicates the byte number of data stored in FIFO for OUT EPX.

12.4.3.5.29 Device DMA Target FIFO Number Register (DMA_TFN, Offset = 0x1C0)

Note: Only one ACC_X bit can be set at the same time.

Table 12-149. Device DMA Target FIFO Number Register (DMA_TFN, Offset = 0x1C0)

Bit	Name	Type	Default Value	Description
[31:5]	-	Rsvd	-	Reserved

Bit	Name	Type	Default Value	Description
4	ACC_CXF	RW	1'b0	**Accessing Control Transfer FIFO** When this bit is set to '1', the DMA target FIFO will be control transfer FIFO.
3	ACC_F3	RW	1'b0	**Accessing FIFO 3** When this bit is set to '1', the DMA target FIFO will be FIFO 3.
2	ACC_F2	RW	1'b0	**Accessing FIFO 2** When this bit is set to '1', the DMA target FIFO will be FIFO 2.
1	ACC_F1	RW	1'b0	**Accessing FIFO 1** When this bit is set to '1', the DMA target FIFO will be FIFO 1.
0	ACC_F0	RW	1'b0	**Accessing FIFO 0** When this bit is set to '1', the DMA target FIFO will be FIFO 0.

12.4.3.5.30 Device DMA Controller Parameter Setting 1 Register (DMA_CPS1, Offset = 0x1C8)

Table 12-150. Device DMA Controller Parameter Setting 1 Register (DMA_CPS1, Offset = 0x1C8)

Bit	Name	Type	Default Value	Description
31	DevPhy_ Suspend (HOV)	RW	1'b0	**Device Transceiver Suspend Mode** Active high Place the transceiver in the suspend mode that draws the minimal power from the power supplies. This bit is only used in the device mode. This bit is in the system clock domain instead of the UCLK clock domain.
[30]	-	Rsvd	-	Reserved
[29]	UNDEF_LEN _BURST (HOV)	RW	1'b1	**AHB Undefined Length Burst** This bit is used for configuring the burst behavior of AHB Initiator Interface. 1'b0: AHB Master initiates at most 4-beat burst transfers 1'b1: AHB Master initiates undefined length burst transfers
[28:25]	R_HPROT (HOV)	RW	4'b0011	**User Definition of AHB HPROT**
[24:8]	DMA_LEN (HOV)	RW	0x0_0000	**DMA Length** These bytes indicate the total bytes moved by the DMA controller. The unit is byte. The maximum length will be 128KB - 1 and must not be configured to '0'. During handling the control transfer, the maximum length must not exceed 64.

Bit	Name	Type	Default Value	Description
[7:5]	-	Rsvd	-	Reserved
4	CLRFIFO_ DMAABORT (HOV)	RW	1'b0	**Clear FIFO when DMA_ABORT** This bit will be set to '1' when combined with the DMA_ABORT bit to clear the content in FIFO after completing the DMA abort. If users need to abort DMA and clear the contents in FIFO, this bit must be set to '1' together with DMA_ABORT. The contents in FIFO will not be cleared if this bit is cleared to '0'.
3	DMA_ABORT (HOV)	RW	1'b0	**DMA Abort** This bit forces the DMA abort during the DMA active. This bit is set to 0b1 to stop the DMA data movement and will be cleared when DMA is stopped. Please note this bit can only be valid in the device mode. Please note that DMA_START and DMA_ABORT cannot be set simultaneously. Doing so will result in unexpected outcomes. Setting this bit to 0b1 when the DMA_START is 0b0 will have no effect.
2	DMA_IO (HOV)	RW	1'b0	**DMA I/O to I/O** This bit forces the DMA controller not to toggle an address. This bit will be set when DMA targets the I/O device instead of the system memory. If this register is set to '1', DMA_LEN must be an integer multiple of DWORD (Four bytes) and DMA_MADDR must be aligned to the boundary of DWORD (Four bytes).
1	DMA_TYPE (HOV)	RW	1'b0	**DMA Type** The transfer type of data moving 0: FIFO to Memory 1: Memory to FIFO
0	DMA_START (HOV)	RW	1'b0	**DMA Start** This register informs the DMA controller to initiate a DMA transfer. This bit is set to start the transfer and is cleared when the DMA operation is completed. Please note that this bit cannot be cleared by the software; it can only be cleared by the hardware in the case of either DMA completion or DMA error. Please note that DMA_LEN, DMA_TYPE and DMA_MADDR must be configured before DMA_START is set.

12.4.3.5.31 Device DMA Controller Parameter Setting 2 Register (DMA_CPS2, Offset = 0x1CC)

Table 12-151. Device DMA Controller Parameter Setting 2 Register (DMA_CPS2, Offset = 0x1CC)

Bit	Name	Type	Default Value	Description
[31:0]	DMA_MADDR	RW	32'h0000	**DMA Memory Address** The starting address of a memory to request a DMA transfer

12.4.3.5.32 Device DMA Controller Parameter Setting 3 Register (DMA_CPS3, Offset = 0x1D0)

Table 12-152. Device DMA Controller Parameter Setting 3 Register (DMA_CPS3, Offset = 0x1D0)

Bit	Name	Type	Default Value	Description
[31:0]	SETUP_CMD_RPORT	RO	-	**Control Transfer Setup Command Read Port** CPU reads 8bytes setup command from this port instead of programming DMA to move this data. Users should note that CPU must read one word each time when it reads this port (A half-word or byte read is not allowed). Before reading this port, the DMA target FIFO (1c0h) must be set to control the transfer FIFO even though DMA is not used to move this data. Each time when CPU reads this port, the internal FIFO pointer will be increased. Under this condition that the pointer increases erroneously, users should not set the target FIFO to the control transfer FIFO while using ICE to scan all ports.

Chapter 13

Peripherals

This chapter contains the following sections:

13.1 GPIO Controller

13.1.1 General Description

The general-purpose I/O controller is a user-programmable general-purpose input/output controller. It is used to input or output data from the system and device. Each GPIO can be programmed as an input or output. GPIO can also be an interrupt input that is supported at the rising edge, falling edge, both edge, and the high/low level interrupt sense types. GPIO provides of up to 32 programmable I/O ports; and each port can be independently programmed.

13.1.2 Features

- Up to 32 independent input, output, and output enable buses for bidirectional I/O pins
- Each port can be bypassed
- Each port can individually trigger the GPIO interrupt when the INTR option is set and programmed as the input pin
- Triggers interrupt generation of each port at rising edge, falling edge, both edges, or at high level or low level when the INTR option is set
- Each port can be pulled high or pulled low when "Pull" option is set (Require to program I/O pad)
- Each port can choose pre-scaled or PCLK clock source when INTR option is set
- Separately sets or clears output data bit
- Sets all ports as inputs by hardware reset

13.1.3 Programming Model

13.1.3.1 Summary of General-purpose I/O Registers

Table 13-1shows the offset, type, width, reset value, name, and configuration option of each GPIO programming register.

Table 13-1. GPIO Register

Offset	Type	Width	Reset	Name	Configuration	Description
0x00	R/W	[gn:0]	0x0	GpioDataOut	None	GPIO data output register
0x04	R	[gn:0]	0x0	GpioDataIn	None	GPIO data input register
0x08	R/W	[gn:0]	0x0	PinDir	None	GPIO direction register 0: Input 1: Output
0x0C	R/W	[gn:0]	0x0	PinBypass	None	GPIO bypass register 0: No bypass 1: Bypass
0x10	W	[gn:0]	0x0	GpioDataSet	None	GPIO data bit set register When writing to this register, the corresponding bits in the data register are set to 1, and the other bits remain unchanged.
0x14	W	[gn:0]	0x0	GpioDataClear	None	GPIO data bit clear register When writing to this register, the corresponding bits in the data register are cleared to 0, and the other bits remain unchanged.
0x18	R/W	[gn:0]	0x0	PinPullEnable	PULL	GPIO pull-up register 0: Pin is not pulled. 1: Pin is pulled.
0x1C	R/W	[gn:0]	0x0	PinPullType	PULL	GPIO pull-high/pull-low register 0: Pin is pulled low. 1: Pin is pulled high.
0x20	R/W	[gn:0]	0x0	IntrEnable	INTR	GPIO interrupt enable register 0: Pin interrupt is disabled. 1: Pin interrupt is enabled.

Offset	Type	Width	Reset	Name	Configuration	Description
0x24	R	[gn:0]	0x0	IntrRawState	INTR	GPIO interrupt raw status register 0: Interrupt is not detected. 1: Interrupt is detected.
0x28	R	[gn:0]	0x0	IntrMaskedState	INTR	GPIO interrupt masked status register 0: Interrupt is not detected or masked. 1: Interrupt is detected and not masked.
0x2C	R/W	[gn:0]	0x0	IntrMask	INTR	GPIO interrupt mask register 0: Mask is disabled. 1: Mask is enabled.
0x30	W	[gn:0]	0x0	IntrClear	INTR	GPIO interrupt clear 0: No effect 1: Clear interrupt
0x34	R/W	[gn:0]	0x0	IntrTrigger	INTR	GPIO interrupt trigger method register 0: Edge-triggered 1: Level-triggered
0x38	R/W	[gn:0]	0x0	IntrBoth	INTR	GPIO edge-trigger interrupt by single or both edges 0: Single edge 1: Both edges
0x3C	R/W	[gn:0]	0x0	IntrRiseNeg	INTR	GPIO interrupt triggered at the rising or falling edge 0: Rising edge 1: Falling edge GPIO interrupt triggered by high or low level 0: High level 1: Low level
0x40	R/W	[gn:0]	0x0	BounceEnable	INTR	GPIO pre-scale clock enable When enabled, the PCLK will be divided by the BouncePreScale clocks. This signal is used to extend the clock cycle of detecting interrupt. 0: Disable 1: Enable
0x44	R/W	[23:0]	0x7D0	BouncePreScale	INTR	GPIO Pre-scale, used to adjust different PCLK frequencies The allowable range is from 0x1 to 0xFFFFFF.
0x7C	R	[31:0]	0x—	RevisionNum	None	GPIO revision number

13.1.3.2 Register Descriptions

The following subsections provide the detailed descriptions of the general-purpose I/O registers.

13.1.3.2.1 GpioDataOut

The GpioDataOut register is the GPIO data out register. When PinDir indicates the pin is an output, the GpioDataOut register is connected to io_out. When PinDir indicates that the pin is an input, GpioDataOut can hold the data.

Table 13-2. GPIO Data-Out Register

Bit	Name	Type	Comment
[gn:0]	GpioDataOut	Read/Write	None

Note: GpioDataOut[I] is dedicated for io_out[I]. I is ranging from 0 to gn.

13.1.3.2.2 GpioDataIn

The GpioDataIn register is the GPIO data in register. When the PinDir indicates that the pin is an input, the GpioDataIn will latch gpio_in at the PCLK rising edge. When PinDir indicates that the pin is an output, the GpioDataIn register is a “don’t care” register.

Table 13-3. GPIO Data-In Register

Bit	Name	Type	Comment
[gn:0]	GpioDataIn	Read	None

Note: GpioDataIn[I] is dedicated for io_out[I]. I is ranging from 0 to gn.

13.1.3.2.3 PinDir

The PinDir register controls gpio_en. When PinDir indicates that the pin is an output, the related bits of gpio_en are set to 1. Otherwise, the related bits of gpio_en are set to 0.

Table 13-4. Pin Direction Register

Bit	Name	Type	Comment
[gn:0]	PinDir	Read/Write	None

Note: PinDir[I] is dedicated for pin[I]. I is ranging from 0 to gn.

13.1.3.2.4 PinBypass

The PinBypass register controls the bypass mode. When the PinBypass register indicates that the pin is in the bypass mode, gpio_en is connected to gpio_bps_en, gpio_in is connected to gpio_bps_out, and gpio_bps_in is connected to gpio_out.

Table 13-5. Pin Bypass Mode Register

Bit	Name	Type	Comment
[gn:0]	PinBypass	Read/Write	None

Note: PinBypass[I] is dedicated for pin[I]. I is ranging from 0 to gn.

13.1.3.2.5 GpioDataSet

GpioDataSet is the bit operation logic. When writing to this address, if some bits of GpioDataSet are '1,' the related bits of GpioDataOut will be set. For example, if GpioDataOut[7:0] = 0x23, and pwdata[7:0] from the APB is 0x47, when writing to the GpioDataSet address, the result of GpioDataOut[7:0] will be 0x67.

Table 13-6. GPIO Data Bit Set Register

Bit	Name	Type	Comment
[gn:0]	GpioDataSet	Write	None

13.1.3.2.6 GpioDataClear

GpioDataClear is the bit operation logic. When writing to this address, if some bits of GpioDataClear are 1s, the related bits of GpioDataOut will be cleared. For example, if GpioDataOut[7:0] = 0x23, and pwdata[7:0] from the APB is 0x47, when writing to the GpioDataClear address, the result of GpioDataOut[7:0] will be 0x20.

Table 13-7. GPIO Data Bit Clear Register

Bit	Name	Type	Comment
[gn:0]	GpioDataClear	Write	None

13.1.3.2.7 PinPullEnable

The PinPullEnable register controls gpio_pullup and gpio_down. If PinPullEnable indicates that the pin is disabled to Pull, gpio_pullup and gpio_down are masked to 0.

Table 13-8. Pin Pull Enable Register

Bit	Name	Type	Comment
[gn:0]	PinPullEnable	Read/Write	Require Pull option

Note: PinPullEnable[I] is dedicated for PAD[I]. I is ranging from 0 to gn.

13.1.3.2.8 PinPullType

PinPullType controls gpio_pullup and gpio_down. If PinPullEnable is enabled and PinPullType is set to 0, gpio_pullup will be set to 0, and gpio_down will be set to 1. Please refer Table 13-10 for the pull truth table.

Table 13-9. Pin Pull Type Register

Bit	Name	Type	Comment
[gn:0]	PinPullType	Read/Write	Require Pull option

Notes:

1. PinPullType[I] is dedicated for PAD[I]. I is ranging from 0 to gn.
2. The pull function requires that the I/O pad can be pulled.

Table 13-10. Pull Truth Table

PinPullEnable	PinPullType	gpio_pullup	gpio_down
0	X	0	0
1	0	0	1
1	1	1	0

13.1.3.2.9 IntrEnable

The IntrEnable register controls the enable or disable interrupt detection logic. It is a mask of the interrupt detection logic. When the pin direction is the input and the interrupt detection is enabled, the pin can accept interrupt from pad. The sensed state is stored in the IntrMaskedState register (Masked by IntrEnable). Before turning on IntrEnable, the programmer can clear the masked state by writing a 1 to IntrClear to ensure the initial state.

Table 13-11. Interrupt Enable Register

Bit	Name	Type	Comment
[gn:0]	IntrEnable	Read/Write	Require INTR option

Note: IntrEnable[I] is dedicated for pin[I]. I is ranging from 0 to gn.

13.1.3.2.10 IntrRawState

The IntrRawState register is the raw result of the interrupt detection. When IntrEnable is enabled, the IntrRawState register will reflect the interrupt detection status. The programmer can poll this register to detect an interrupt.

Table 13-12. Interrupt Raw State Register

Bit	Name	Type	Comment
[gn:0]	IntrRawState	Read	Require INTR option

Note: IntrRawState[I] is dedicated for pin[I]. I is ranging from 0 to gn.

13.1.3.2.11 IntrMaskedState

The IntrMaskedState register is the masked result of the interrupt detection. The IntrMaskedState register is controlled by IntrEnable, IntrRawState, and IntrMask registers.

Table 13-13. Interrupt Masked State Register

Bit	Name	Type	Comment
[gn:0]	IntrMaskedState	Read	Require INTR option

Note: IntrMaskedState[I] is dedicated for pin[I]. I is ranging from 0 to gn.

13.1.3.2.12 IntrMask

The IntrMask register is the mask register of the interrupt detection. It masks the IntrRawState register. For example, if IntrEnable[0] = 1, IntrRawState[0] = 1, and IntrMask[0] = 1, then IntrMaskedState[0] will never change to 1.

Table 13-14. Interrupt Mask Register

Bit	Name	Type	Comment
[gn:0]	IntrMask	Read/Write	Require INTR option

Note: IntrMask[I] is dedicated for pin[I]. I is ranging from 0 to gn.

13.1.3.2.13 IntrClear

The IntrClear register is the bit operation logic. If some bits of pwdata are set, when writing to the IntrClear address, the related bits of IntrMaskedState and IntrRawState will be cleared.

Table 13-15. Interrupt Clear Register

Bit	Name	Type	Comment
[gn:0]	IntrClear	Write	Require INTR option

13.1.3.2.14 IntrTrigger

The IntrTrigger register indicates the interrupt trigger method of each pin. If IntrTrigger is 0, the interrupt is triggered at the edge; otherwise, the interrupt is triggered at the level.

Table 13-16. Interrupt Trigger Method Register

Bit	Name	Type	Comment
[gn:0]	IntrTrigger	Read/Write	Require INTR option

Note: IntrTrigger[I] is dedicated for pin[I]. I is ranging from 0 to gn.

13.1.3.2.15 IntrBoth

The IntrBoth register indicates that the edge is triggered at both edges or single edge. If IntrTrigger indicates that the edge trigger and IntrBoth is 0, the interrupt edge trigger is done by the single edge. If IntrTrigger indicates that the edge trigger and IntrBoth is 1, the interrupt edge trigger is done by both edges.

Table 13-17. Interrupt Both Edge Trigger Register

Bit	Name	Type	Comment
[gn:0]	IntrBoth	Read/Write	Require INTR option

Note: IntrBoth[I] is dedicated for pin[I]. I is ranging from 0 to gn.

13.1.3.2.16 IntrRiseNeg

The IntrRiseNeg register indicates whether the edge trigger is at the rising edge or falling edge. If IntrTrigger is the edge trigger, IntrBoth is the single edge, and IntrRiseNeg is '0,' the interrupt edge trigger is done at the rising edge. If IntrTrigger is the edge trigger, IntrBoth is the single edge, and IntrRiseNeg is '1,' the interrupt edge trigger is done at the falling edge.

Table 13-18. Interrupt Rise or Neg Edge Trigger Register

Bit	Name	Type	Comment
[gn:0]	IntrRiseNeg	Read/Write	Require INTR option

Note: IntrRiseNeg[I] is dedicated for pin[I]. I is ranging from 0 to gn.

13.1.3.2.17 BounceEnable

The BounceEnable register controls the bounce function. If BounceEnable is on, the interrupt detection is sampled by the extended clock. The extension number is controlled by the BouncePreScale register.

Table 13-19. Bounce Enable Register

Bit	Name	Type	Comment
[gn:0]	BounceEnable	Read/Write	Require INTR option

Note: BounceEnable[I] is dedicated for pin[I]. I is ranging from 0 to gn.

13.1.3.2.18 BouncePreScale

The BouncePreScale register is an register to indicate the bounce timer. It can extend PCLK to the BouncePreScale cycles. This register can be used to adjust the interrupt sample clock period in different machines. The reset value is 0x7D0, which means that if the APB clock frequency is 66MHz, the de-bounce clock will be divided by (0x7D0+1) to 32.98kHz. The programmer can adjust this register to fit different systems.

Table 13-20. Bounce Clock Pre-scale Register

Bit	Name	Type	Comment
[23:0]	BouncePreScale	Read/Write	Require INTR option

13.2 I²C Controller

13.2.1 General Description

The I²C interface controller allows the host processor to serve as a master or slave in the I²C bus. Data are transmitted to and received from the I²C bus via a buffered interface.

13.2.2 Features

- Supports standard and fast modes by using clock division register
- Supports 7bit, 10bit, and general-call addressing modes
- Supports glitch suppression for de-bounce circuit
- Programmable slave address
- Supports master-transmit, master-receive, slave-transmit, and slave-receive modes
- Supports multi-master mode
- Supports general-call address detection in slave mode
- Not supports START byte procedure
- Supports system manager bus 2.0 in master and slave modes
- Supports all system manager bus 2.0 protocol commands except quick command and host notify protocol

13.2.3 Programming Model

13.2.3.1 Summary of I²C Controller Registers

The following registers are associated with the I²C interface controller. These registers are allocated within the peripheral memory-mapped addresses of the host processor.

Table 13-21. Summary of I²C Controller Registers

Offset	Type	Description	Reset Value
0x00	R/W	I²C Control Register (CR)	0x0000_0000
0x04	R/RC	I²C Status Register (SR)	0x0000_0000
0x08	R/W	I²C Clock Division Register (CDR)	0x0000_0000
0x0C	R/W	I²C Data Register (DR)	0x0000_0000
0x10	R/W	I²C Slave Address Register (SAR)	0x0000_0000
0x14	R/W	I²C Setup/Hold Time and Glitch Suppression Setting Register (TGSR)	0x0000_0401
0x18	R	I²C Bus Monitor Register (BMR)	0x0000_0003
0x1C	R/W	SM Bus Control Register (SMCR)	0x0000_0000
0x20	R/W	Maximum Timeout Register (MAXTR)	0x003f_ffff
0x24	R/W	Minimum Timeout Register (MINTR)	0x003f_ffff
0x28	R/W	Master Extend Time Register (METR)	0x000f_ffff
0x2C	R/W	Slave Extend Time Register (SETR)	0x001f_ffff
0x30	R	I²C Revision Register	-
0x34	R	I²C Feature Register	-

13.2.3.2 Register Descriptions

The following subsections describe the details of the I^{2}C controller registers.

13.2.3.2.1 I^{2}C Control Register (CR, Offset = 0x00)

The host processor uses the I^{2}C Control Register (CR) to control transmitting and receiving data from the I^{2}C bus. Please note that SCL_EN must be set to '1' to enter the master mode unless users want to release SCLout in a special case. Users can set SCL_EN to '0' to enter the slave mode. The START and STOP bits can only be set in the master mode.

Table 13-22 shows the bit assignments of the I^{2}C control register.

Table 13-22. I^{2}C Control Register (CR, Offset = 0x00)

Bit	Name	Type	Description
[31:18]	-	-	Reserved
17	Test_bit	R/W	In special test mode; this bit must be set to '0'.
16	SDA_LOW	R/W	If this bit is set, SDAout should be tied to '0'. For a normal case, it is suggested setting this bit to '0'.
15	SCL_LOW	R/W	If this bit is set, SCLout should be tied to '0'. For a normal case, it is suggested setting this bit to '0'.
14	STARTI_EN	R/W	If this bit is set, the I^{2}C controller will interrupt the host processor when I^{2}C controller detects a start condition on the I^{2}C bus.
13	ALI_EN	R/W	If this bit is set, the I^{2}C controller will interrupt the host processor when the I^{2}C controller loses the arbitration in the master mode.
12	SAMI_EN	R/W	If this bit is set, the I^{2}C controller will interrupt the host processor when the I^{2}C controller detects a slave address that matches the SAR register or a general-call address (When GC_EN is set).
11	STOPI_EN	R/W	If this bit is set, the I^{2}C controller will interrupt the host processor when the I^{2}C controller detects a stop condition on the I^{2}C bus.
10	BERRI_EN	R/W	If this bit is set, the I^{2}C controller will interrupt the host processor when the I^{2}C controller detects non-ACK responses of the slave device after one data byte has been sent in the master mode.
9	DRI_EN	R/W	If this bit is set, the I^{2}C controller will interrupt the host processor when the DR register has received one data byte on the I^{2}C bus.
8	DTI_EN	R/W	If this bit is set, the I^{2}C controller will interrupt the host processor when the DR register has transmitted one data byte on the I^{2}C bus.

Bit	Name	Type	Description
7	TB_EN	R/W	When Transfer Byte Enable (TB_EN) is set, the I^2C controller will be ready to receive or transmit one byte. Otherwise, the I^2C controller will insert the wait state by pulling SCLout low on the I^2C bus. Please note that TB_EN will be automatically cleared when the device receives or transmits one byte or detects the stop/start condition.
6	ACK/NACK	R/W	When the I^2C controller is in the master-receive or slave-receive mode, the controller will send out the acknowledge signal. 0: ACK 1: NACK
5	STOP	R/W	When the I^2C controller is in the master mode, this controller will initiate a stop condition after transferring the next data byte on the I^2C bus.
4	START	R/W	When the I^2C bus is idle, the I^2C controller will initiate a start condition or initiate a repeated start condition after transferring the next data byte on the I^2C bus in the master mode.
3	GC_EN	R/W	By enabling this bit, the I^2C controller will respond to a general-call message as a slave.
2	SCL_EN	R/W	Enable the I^2C controller clock output for the master mode operation
1	I2C_EN	R/W	Enable the I^2C bus interface controller
0	I2C_RST	R/W	Reset the I^2C controller This bit will be automatically cleared after two PCLK clocks.

13.2.3.2.2 I^2C Status Register (SR, Offset = 0x04)

The I^2C and SMBus interrupts are controlled by the ISI2C pin. The ISI2C pin will be set when the interrupt enable bits, (CR) and (SMCR), in the control register and corresponding status register are set. When the interrupt of the I^2C controller is asserted, software will read the SR bits to check the status of the I^2C controller.

The SR bit is also used to clear the following interrupts by reading the register status:

- DR receive data is completed
- DR transmit data is completed
- Slave address is detected
- Bus error is detected
- Start condition is detected
- Stop condition is detected
- Arbitration loss is detected

- Leave suspend mode
- Receive alert response address
- Enter suspend mode
- Alert active detected
- Detect maximum timeout
- Detect minimum timeout
- Detect master extend time
- Detect slave extend time

Table 13-23 shows the bit assignments of the I^2C status register.

Table 13-23. I^2C Status Register (SR, Offset = 0x04)

Bit	Name	Type	Description
[31:22]	-	-	Reserved
21	SAL	RC	When this bit is set, SM bus controller will lose arbitration in the slave mode.
20	DDA	RC	When this bit is set, the slave receive address will be the device default address.
19	ARA	RC	When this bit is set, the slave receive address will be the alert response address.
18	RESUME	RC	When this bit is set, the system will leave the suspend mode.
17	SUSPEND	RC	When this bit is set, the system will enter the suspend mode.
16	ALERT	RC	When this bit is set, the master will read Alert Response Address (ARA).
15	TOUTMax	RC	When this bit is set, the single clock held low will be longer than TTIMEOUT, MAX.
14	TOUTMin	RC	When this bit is set, the single clock held low will be longer than TTIMEOUT, MIN.
13	MEXT	RC	When this bit is set, the cumulative clock low extend time will be longer than TLow_MEXT.
12	SEXT	RC	When this bit is set. the cumulative clock low extend time will be longer than TLow_SEXT.
11	START	RC[1]	When this bit is set, the I^2C controller will detect a start condition on the I^2C bus.
10	AL	RC	When this bit is set, the I^2C controller will lose arbitration in the master mode.
9	GC	RC	When this bit is set, the I^2C controller will receive a slave address that matches the general-call address in the slave mode.
8	SAM	RC	When this bit is set, the I^2C controller will receive a slave address that matches the address of the slave register (SAR) in the slave mode.

[1] RC means "Read and clear".

Bit	Name	Type	Description
7	STOP	RC	When this bit is set, the I^{2}C controller will detect a stop condition on the I^{2}C bus.
6	BERR	RC	When this bit is set, the I^{2}C controller will detect non-ACK response of the slave device after one data byte has been transmitted in the master mode.
5	DR	RC	When this bit is set, the data register (DR) will receive one data byte on the I^{2}C bus.
4	DT	RC	When this bit is set, the data register (DR) will transmit one data byte on the I^{2}C bus.
3	BB	R	When this bit is set, the I^{2}C bus will be busy and the I^{2}C controller will not be involved in the transaction.
2	I^{2}CB	R	When this bit is set, the I^{2}C controller will be busy during the time period between START and STOP.
1	ACK	R	When this bit is set, the I^{2}C controller will receive or send non-acknowledgement.
0	RW	R	When this bit is set, the I^{2}C controller will serve in a master-receive or slave-transmit mode.

13.2.3.2.3 I^{2}C Clock Division Register (CDR, Offset = 0x08)

The I^{2}C clock division register (CDR) defines the divided value used to generate the I^{2}C SCL clock. This register is used with an internal 18bit counter. When the SCL enable bit in the control register is set, this counter will decrement from the programmed value to zero, and then will reload the programmed value and will decrement again. Each time the counter reaches zero, the SCL line transaction will start from high to low, or vice versa, depending on the current state. This register can be configured to select the transfer speed needed on the I^{2}C bus.

Because the controller supports the speed of the I^{2}C bus up to 400kHz, the minimum PCLK must be 6MHz.

Table 13-24. I^{2}C Clock Division Register (CDR, Offset = 0x08)

Bit	Name	Type	Description
[31:18]	-	-	Reserved
[17:0]	COUNT	R/W	Counter value used to generate an I^{2}C clock (SCLout) from the internal bus clock (PCLK). The relationship between PCLK and I^{2}C bus clock (SCLout) is shown in the following equation, where GSR is TGSR[12:10]: SCLout = PCLK/(2 * (COUNT + 2) + GSR)

13.2.3.2.4 I²C Data Register (DR, Offset = 0x0C)

The I²C Data Register (DR) is used by the host processor to transmit and receive data from the I²C bus. The DR is accessed by either the host processor or the I²C controller Shift Register (SHR). Data coming from the I²C bus interface are received by the DR register after a full data byte has been received. Data going out of the I²C bus interface are written into the DR register by the host processor and sent to the serial bus. When the I²C controller is in the transmit mode, the host processor writes data into the DR register over the APB bus. This occurs when a master transition is initiated or when a data transmit interrupt is signaled. Data are moved from the DR register to the SHR register when the transfer byte enable (TB_EN) in the control register is set. The data transmit interrupt is signaled when one byte of data has been transferred on the I²C bus. If DR is not written by the host processor before the next byte package, the I²C controller will insert a wait state until the host processor writes to the DR and sets the Transfer Byte Enable bit. When the I²C controller is in the received mode, the processor reads the DR register data over the APB bus, when the received interrupt of DR is signaled. When the I²C controller has received one new data byte, it will automatically clear transfer byte enable bit (TB_EN) and issue the ACK/NACK on the I²C bus. After issuing ACK/NACK, the I²C controller will insert DR received interrupt to processor. Users must set the Transfer Byte bit again for the next byte transfer on the I²C bus.

Table 13-25. I²C Data Register (DR, Offset = 0x0C)

Bit	Name	Type	Description
[31:8]	-	-	Reserved
[7:0]	DR	R/W	Buffer for the I²C bus data transmission and reception

13.2.3.2.5 I²C Slave Address Register (SAR, Offset = 0x10)

The I²C slave address register defines the I²C controller 1bit or slave address to which the processor responds when the I²C controller operates in the slave mode. The host processor writes this register before enabling I²C operation. The register is fully programmable, so it can be set to a value other than the fixed slave peripheral address preexisted in the system.

Table 13-26. I²C Slave Address Register (SAR, Offset = 0x10)

Bit	Name	Type	Description
31	EN10	R/W	10bit addressing mode enable bit

Bit	Name	Type	Description
[30:10]	-	-	Reserved
[9:7]	SAR	R/W	The most significant 3bit address to which the I^{2}C controller responds when I^{2}C operates in the 10bit addressing slave mode (EN10: 1). When EN10: 0, the I^{2}C controller ignores these three bits.
[6:0]	SAR	R/W	The 7bit address to which the I^{2}C controller responds when the I^{2}C operates in the 7bit addressing slave mode (EN10: 0) or the least significant 7bit address to which the I^{2}C controller responds when the I^{2}C operates in the 10bit addressing slave mode.

13.2.3.2.6 I^{2}C Set/Hold Time and Glitch Suppression Setting Register (TGSR, Offset = 0x14)

The I^{2}C set/hold time and glitch suppression setting register (TGSR[9:0]) defines the PCLK clock cycles. After the SCL bus goes low, the data will be sent to the SDA bus when the I^{2}C controller serves as a transmitter, or an acknowledgement will be sent to the SDA bus when the I^{2}C controller serves as a receiver. The I^{2}C set/hold time and glitch suppression setting register (TGSR[12:10]) defines the values of the PCLK clock period when the I^{2}C bus interface has a built-in glitch suppression logic. Glitches are suppressed according to TGSR[12:10] * PCLK clock period. For example, with a 66MHz (15ns period) PCLK clock and TGSR[12:10] = 3'b100, glitches of 60ns or less are suppressed. With a 40MHz (25ns period) clock and TGSR[12:10] = 2'b010, glitches of 50ns or less are suppressed. This is within the 50ns glitch suppression specification. The only limitation is: CDR > 3 + GSR + TSR.

Table 13-27. I^{2}C Set/Hold Time and Glitch Suppression Setting Register (TGSR, Offset = 0x14)

Bit	Name	Type	Description
[31:13]	-	-	Reserved
[12:10]	GSR	R/W	These bits define the values of PCLK clock period when the I^{2}C Bus Interface has built-in glitch suppression logic. Glitch is suppressed according to "GSR * PCLK" clock period.
[9:0]	TSR	R/W	These bits define the delay values of the PCLK clock cycles that the data or acknowledgement will be driven into the I^{2}C SDA bus after the I^{2}C SCL bus goes low. The actual delay value is GSR+TSR+4. Figure 13-1 shows the relationship. Note: TSR cannot be set to zero.

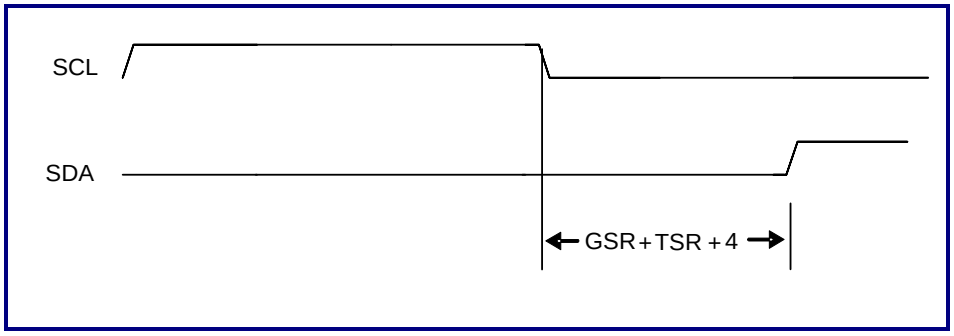


Figure 13-1. Relationship among TSR, SCL, and SDA

13.2.3.2.7 I²C Bus Monitor Register (BMR, Offset = 0x18)

Table 13-28. I²C Bus Monitor Register (BMR, Offset = 0x18)

Bit	Name	Type	Description
[31:2]	-	-	Reserved
1	SCLin	R	This bit continuously reflects the value of the SCLin pin.
0	SDAin	R	This bit continuously reflects the value of the SDAin pin.

13.2.3.2.8 I²C Revision Register (Offset = 0x30)

Table 13-29. I²C Revision Register (Offset = 0x30)

Bit	Name	Type	Description
[31:0]	REVISION	R	These bits represent the revision numbers.

13.2.3.2.9 I²C Feature Register (Offset = 0x34)

Table 13-30. I²C Feature Register r (Offset = 0x34)

Bit	Name	Type	Description
[31:1]	-	-	Reserved
0	SMBUS	R	1: SM bus is configured. 0: No SM bus is configured.

13.2.3.2.10 SM Control Register (SMCR, Offset = 0x1C)

The host processor uses the SM Control Register (SMCR) to control the SMBus suspend and alert outputs. Because suspend is an optional signal, some devices cannot detect if the system is in the suspend mode or not. To prevent these devices from transferring data in the system suspend mode, the slave device should pull down SCL and do not release SCL until SMBUS resumes when the device issues start in the suspend mode. The I^2C bus interface controller can only support this operation in the slave mode.

Table 13-31. SM Control Register (SMCR, Offset = 0x1C)

Bit	Name	Type	Description
[31:12]	-	-	Reserved
11	SUSOUT_EN	R/W	If set, enable the suspend output. It is only used in the master mode.
10	SUS_LOW	R/W	If set, the system will issue suspend after stop.
9	ALERT_LOW	R/W	If set, the system will pull down ALERTout.
8	SAL_EN	R/W	If set, enable the SM bus controller to interrupt the host processor when the slave arbitration loses.
[7:6]	-	-	Reserved
5	RSM_EN	R/W	If set, enable the SM bus controller to interrupt the host processor when the system leaves the suspend mode.
4	SUS_EN	R/W	If set, enable the SM bus controller to interrupt the host processor when the system enters the suspend mode.
3	ALERT_EN	R/W	If set, enable the SM bus controller to interrupt the host processor when the system needs to read the Alert Response Address (ARA).
2	TOUT_EN	R/W	If set, enable the SM bus controller to interrupt the host processor when the clock held low single is longer than TTIMEOUT_MAX or TTIMEOUT_MIN.
1	MEXT_EN	R/W	If set, enable the SM bus controller to interrupt the host processor when the cumulative clock low extend time is longer than TLow_MEXT.
0	SEXT_EN	R/W	If set, enable the SM bus controller to interrupt the host processor when the cumulative clock low extend time is longer than TLow_SEXT.

13.2.3.2.11 SM Maximum Timeout Register (MAXTR, Offset = 0x20)

This register can be configured to set the maximum timeout on the SM bus. Please note that PCLK will be used as the clock of the Timer.

Table 13-32. SM Maximum Timeout Register (MAXTR, Offset = 0x20

Bit	Name	Type	Description
[31:22]	-	-	Reserved
[21:0]	TMAX	R/W	The register defines the count for maximum timeout

13.2.3.2.12 SM Minimum Timeout Register (MINTR, Offset = 0x24)

Table 13-33. SM Minimum Timeout Register (MINTR, Offset = 0x24)

Bit	Name	Type	Description
[31:22]	-	-	Reserved
[21:0]	TMIN	R/W	The register defines the count for minimum timeout

13.2.3.2.13 SM Master Extend Time Register (METR, Offset = 0x28)

Table 13-34. SM Master Extend Time Register (METR, Offset = 0x28)

Bit	Name	Type	Description
[31:20]	-	-	Reserved
[19:0]	TMEXT	R/W	The register defines the count for master access time from “START to ACK,” “ACK to ACK,” and “ACK to STOP.”

13.2.3.2.14 SM Slave Extend Time Register (SETR, Offset = 0x2C)

Table 13-35. SM Slave Extend Time Register (SETR, Offset = 0x2C)

Bit	Name	Type	Description
[31:21]	-	-	Reserved
[20:0]	TSEXT	R/W	The register defines the count for slave access time from “START to STOP.”

13.3 IR Controller

13.3.1 General Description

The IR receiver interface is used for the remote control. The GPIO receive input signals from the IR module and the IRDET module are decoded by software programming. The IRDET module supports most popular protocols, including NEC, Sharp, Sony, and Philips.

13.3.2 Features

- Disables IRDA by software programming
- Provides divider counter to program appropriate frequency
- Provides data arrival interrupt

13.3.3 Programming Model

13.3.3.1 Control Register (Offset = 0x00)

Table 13-36. Control Register (Offset = 0x00)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:5]	-	-	Reserved	-	-
[4]	MSB_F	R/W	MSB first 1: ir_data[31] is the MSB bit. 0: ir_data[0] is the MSB bit.	0x0	HR
[3]	bi_phase	R/W	Bi-phase mode selection	0x0	HR
[2]	iil	R/W	Inverse of the input level	0x0	HR
[1]	NSCD	R/W	This bit indicates that no start code will be detected.	0x0	HR
[0]	enable	R/W	Enable IRDET	0x0	HR

13.3.3.2 Param0 Register (Offset = 0x04)

Table 13-37. Param0 Register (Offset = 0x04)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:16]	timeout_th	R/W	Timeout threshold	0x0	HR
[14:0]	high_th0	R/W	Logic-high threshold	0x0	HR

13.3.3.3 Param1 Register (Offset = 0x08)

Table 13-38. Param1 Register (Offset = 0x08)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:16]	low_th1	R/W	Logic-low threshold 1	0x0	HR
[15:0]	low_th0	R/W	Logic-low threshold 0	0x0	HR

13.3.3.4 Param2 Register (Offset = 0x0C)

Table 13-39. Param2 Register (Offset = 0x0C)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:21]	-	-	Reserved	-	-
[20:16]	channel_sel	R/W	These bits select the GPIO pin to be the IR input. 00000: GPIO[0] 00001: GPIO[1] 00010: GPIO[2] --- 11111: GPIO[31]	0x0	HR
[13:8]	fifo_size	R/W	fifo_size-1	-	-
[7:0]	clk_div	R/W	Clock divider The clock divider source is 12MHz.	0x0	HR

13.3.3.5 Status Register (Offset = 0x10)

Table 13-40. Status Register (Offset = 0x10)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:8]	-	-	Reserved	-	-
[7:2]	data_amount	R/W	These bits indicate the data amount.	-	-
[1]	repeat	R/W	Repeat command	-	-
[0]	ir_int	R/W	Data arrival interrupt Write 1'b1 to clear	0x0	HR

13.3.3.6 IR Data Register (Offset = 0x14)

Table 13-41. IR Data Register (Offset = 0x14)

Bit	Name	Type	Description	Reset Value	Reset Type
[31:0]	IRDATA	R/W	IR receive data	0x0	HW/WR

13.3.3.7 Programming Setting

13.3.3.7.1 NEC Protocol

- Carrier frequency of 28kHz
- Pulse distance modulation
- Bit time of 1.125ms or 2.25ms
- 8bit address and 8bit command length
- Address and command transmitted twice for reliability

The register settings are suggested as follows:

- Set Param0 (0x04) = 32'h0400_0032
- Set Param1 (0x08) = 32'h0070_0190
- Set Param2 (0x0C) = 32'h0000_1f_77
- Set Status (0x10) = 8'h3
- Set Control (0x00) = 8'b00001

13.3.3.7.2 RCA Protocol

- Carrier frequency of 56kHz
- Pulse distance modulation
- Bit time of 1.5ms or 2.5ms
- 4bit address and 8bit command length (12bit protocol)
- Complement of codes sent out after real code for reliability

The register settings are suggested as follows:

- Set Param0 (0x04) = 32'h0400_0032
- Set Param1 (0x08) = 32'h0070_0190
- Set Param2 (0x0C) = 32'h0000_1f_77
- Set Status (0x10) = 8'h3
- Set Control (0x00) = 8'b10001

13.3.3.7.3 Sharp Protocol

- Carrier frequency of 38kHz
- Pulse distance modulation
- Bit time of 1ms or 2ms
- 8bit command and 5bit address length

The register settings are suggested as follows:

- Set Param0 (0x04) = 32'h0fc0_001e
- Set Param1 (0x08) = 32'h0076_0fc0
- Set Param2 (0x0C) = 32'h0008_1e_77
- Set Status (0x10) = 8'h3
- Set Control (0x00) = 8'b00011

13.3.3.7.4 Philips RC-5 Protocol

- Carrier frequency of 36kHz
- Bi-phase coding (Aka Manchester coding)
- 5bit address and 6bit command length (7 command bits for RC5X)
- Constant bit time of 1.778ms (64cycles of 36kHz)

The register settings are suggested as follows:

- Set Param0 (0x04) = 32'h0110_0000
- Set Param1 (0x08) = 32'h0087_0000
- Set Param2 (0x0C) = 32'h0010_1f_77
- Set Status (0x10) = 8'h3
- Set Control (0x00) = 8'b110

13.3.3.7.5 Nokia NRC17 Protocol

- Carrier frequency of 38kHz
- Bi-phase coding
- Constant bit time of 1ms
- 8bit command, 4bit address, and 4bit sub-code length
- Battery empty indication is possible.

The register settings are suggested as follows:

- Set Param0 (0x04) = 32'h0130_0017
- Set Param1 (0x08) = 32'h004b_00f8
- Set Param2 (0x0C) = 32'h001f_3f_77
- Set Status (0x10) = 8'h3
- Set Control (0x00) = 8'b01001

13.4 UART Controller

13.4.1 General Description

The UART controller is a serial communication element that implements the UART operation mode and is backward compatible to 16550 to support the existing communication software.

13.4.2 Features

- High-speed NS 16C550A-compatible UART
- Programmable baud rate up to 115.2Kbps
- Ability to add or delete standard asynchronous communication bits (Start, stop, and parity) of serial data
- Programmable baud rate generator to divide internal clock from 1 to (216 - 1) to generate internal 16X clock
- Fully programmable serial interface:
  - 5bit, 6bit, 7bit, or 8bit characters
  - Even, odd, and no parity detection
  - 1, 1.5, or 2 stops bit generation
- Complete status reporting capability
- Ability to generate and detect line breaks
- Fully prioritized interrupt system controls
- Separates DMA requests for transmit and receive data service
- Break, parity, overrun, and framing error simulation for UART mode
- Provides 16byte transmit FIFO and 16byte receive FIFO by UART

13.4.3 Programming Model

13.4.3.1 Summary of UART Controller Registers

Table 13-42. Summary of UART Controller Registers

Offset	Type	Width	Name	Description	Reset Value
UART Mode					
+0x00	R	8	RBR	Receiver Buffer Register	0x00
	W	8	THR	Transmitter Holding Register	0x00
+0x04	R/W	4	IER	Interrupt Enable Register	0x00
+0x08	R	8	IIR	Interrupt Identification Register	0x01
	W		FCR	FIFO Control Register	0x00
+0x0C	R/W	8	LCR	Line Control Register	0x00
+0x10	R/W	7	MCR	Modem Control Register	0x00
+0x14	R	8	LSR	Line Status Register	0x60
	W		TST	Testing Register	0x00
+0x18	R	8	MSR	Modem Status Register	0x00
+0x1C	R/W	8	SPR	Scratch Pad Register	0x00
Registers accessible when DLAB = 1					
+0x00	R/W	8	DLL	Baud Rate Divisor Latch Least Significant Byte	0x01
+0x04	R/W	8	DLM	Baud Rate Divisor Latch Most Significant Byte	0x00
+0x08	R/W	5	PSR	Prescaler Register	0x01

13.4.3.2 Register Descriptions

13.4.3.2.1 Receiver Buffer Register (RBR, Offset = 0x00 for Read)

Users can receive data by reading this read-only register. It can be the data read port of RX FIFO or 1byte register, depending on enabling or disabling FIFOs.

- If FIFOs are enabled, this location refers to the top of a 16byte FIFO (Next to be read).
- If FIFOs are not enabled, this location is a 1byte register (The bottom word of receive FIFO) that receives the contents of the receiver shift register once a character has been assembled.

Table 13-43. Receiver Buffer Register (RBR, Offset = 0x00 for Read)

Bit	Name	Type	Function
8	RBR	R	Receive data port

13.4.3.2.2 Transmitter Holding Register (THR, Offset = 0x00 for Write)

The transmitter holding register is used to write the transmitter holding register or the transmit FIFO, depending on enabling or disabling FIFO.

- If FIFO is enabled, data will be written to this location and pushed to the transmit FIFO.
- If FIFO is not enabled, data will be written to this location and stored in the transmitter holding register (The bottom entry of the transmit FIFO).

If the width of a transmitted character is less than eight bits, this character must be right-justified. The left bit (i.e. MSB) is "don't care" bit. For example, with a word length of five bits, writing 0xd3 or 0xf3 will result in a transmission of 13h character. Before writing to this register, users must ensure that the UART controller is ready to accept data for transmission; that is, to ensure that the THR Empty flag is set in LSR (Please refer to the description of this register below).

Table 13-44. Transmitter Holding Register (THR, Offset = 0x00 for Write)

Bit	Name	Type	Function
8	THR	W	Transmit data port

13.4.3.2.3 Interrupt Enable Register (IER, Offset = 0x04)

This register individually enables each possible interrupt source. Writing '1' to any bit in this register enables the corresponding interrupt, while writing '0' to any bit of this register disables the operation. For detailed descriptions of the interrupt sources, please refer to the description of the Interrupt Identification Register (IIR) on the next page.

Table 13-45. Interrupt Enable Register (IER, Offset = 0x04)

Bit	Name	Type	Function
[7:4]	Reserved	-	-
3	MODEM Status	R/W	This bit enables the modem status interrupt when set to logic ‘1’.
2	Receiver Line Status	R/W	This bit enables the Receiver Line Status Interrupt when set to logic ‘1’.
1	THR Empty	R/W	This bit enables the Transmitter Holding Register Empty Interrupt when set to logic ‘1’.
0	Receiver Data Available	R/W	This bit enables the Received Data Available Interrupt (And character reception timeout interrupts in the FIFO mode) when set to logic ‘1’.

13.4.3.2.4 Interrupt Identification Register (IIR, Offset = 0x08)

This register identifies the interrupt of the highest priority that is currently pending. The UART controller implements a 4-level priority encoder from the highest priority to the lowest priority, as follows:

1. Receive Line Status (Highest priority)
2. Receive Data Ready and Character Reception Timeout (Second priority)
3. Transmitter Holding Register Empty (Third priority)
4. Modem Status (Lowest priority)

Table 13-47 describes the interrupt conditions and identification codes, together with the reset method.

Table 13-46. Interrupt Identification Register (IIR, Offset = 0x08)

Bit	Name	Type	Function
[7:6]	FIFO mode enable	R	These bits are set when FCR[0] is set to ‘1’.
5	Reserved	R	This bit is always ‘0’.
4	Tx FIFO full	R	This bit is set to ‘1’ when TX FIFO is full.
3	FIFO mode only	R	In the 16450 mode, this bit is ‘0’. In the FIFO mode, this bit is set along with bit 2 when a timeout interrupt is pending.
[2:1]	Interrupt Identification Code	R	These bits identify the highest priority interrupt that is pending. Please note that when an interrupt source is considered as pending, the corresponding bit in the IER must be enabled.
0	Interrupt Pending	R	This bit is used in a prioritized interrupt environment to identify the pending interrupt. 0: When an interrupt is pending, the IIR contents can be used as a pointer to the appropriate interrupt service routine. 1: No interrupt is pending.

Table 13-47. Interrupt Control

Interrupt Identification Register					Interrupt Set and Reset Functions		
Bit 3	**Bit 2**	**Bit 1**	**Bit 0**	**Priority Level**	**Interrupt Type**	**Interrupt Source Description**	**Interrupt Reset Method**
0	0	0	1	-	None	There is no interrupt pending.	None
0	1	1	0	Highest	Receiver Line Status	There is an overrun error, parity error, framing error, or break interrupt indication corresponding to the received data on top of the receive FIFO. Please note that the FIFO error flag in LSR does not influence this interrupt, which is related only to the data on top of the Rx FIFO. This is directly related to the presence of '1' in any of the LSR bits 1 to 4.	Read the Line Status Register (LSR)
0	1	0	0	Second	Received Data Ready	In the non-FIFO mode, there is received data available in the RHR register. In the FIFO mode, the number of characters in the receive FIFO is equal to or greater than the trigger level programmed in FCR. The interrupt signal will stay active while the number of words in FIFO stays higher than that value and will be cleared when the microprocessor reads the necessary words to make the number of words in the FIFO less than the trigger level. Please note that this is not directly related to LSR bit 0, which always indicates that there is at least one word ready.	Read the Receiver Buffer Register (RBR)

Interrupt Identification Register					Interrupt Set and Reset Functions		
Bit 3	**Bit 2**	**Bit 1**	**Bit 0**	**Priority Level**	**Interrupt Type**	**Interrupt Source Description**	**Interrupt Reset Method**
1	1	0	0	Second	Character Reception Timeout	There is at least one character in the receive FIFO and during a time corresponding to four characters at the selected baud rate, no new character has been received. A FIFO timeout interrupt will occur, if the following conditions exist: 1. At least one character is in FIFO. 2. The most recent serial character received was longer than four continuous characters time ago (If 2 stop bits are programmed, the second one is included in this time delay). 3. The most recent CPU read of FIFO was longer than four continuous characters time ago.	Read the Receiver Buffer Register (RBR) Note: In the UART mode, if RX timeout is asserted, reset the FIFO will not clear it, users must read RBR.
0	0	1	0	Third	Transmitter Holding Register Empty	In the non-FIFO mode, the 1byte THR is empty. In the FIFO mode, the complete 16byte transmit FIFO is empty. Consequently, 1 to 16 characters can be written to THR. That is to say, the THR Empty bit in LSR is 1.	Write the Transmitter Holding Register (THR) Alternatively, reading the Interrupt Identification Register (IIR) will also clear the interrupt if this is the interrupt type being currently indicated (this will not clear the flag in the LSR).
0	0	0	0	Fourth	Modem Status	A change has been detected in the Clear To Send (CTS), Data Set Ready (DSR) or Carrier Detect (CD) input lines or a trailing edge in the Ring Indicator (RI) input line. That is to say, at least one of MSR bits 0 to 3 is one.	Read the Modem Status Register (MSR)

13.4.3.2.5 FIFO Control Register (FCR, Offset = 0x08 for Write)

This is a write-only register at the same location as IIR (IIR is a read-only register). This register is used to enable and clear FIFOs and set the RX FIFO trigger level. If users reset FIFO, the status FIFO will not be cleared. Users must clear the status FIFO by reading the corresponding register.

Table 13-48. FIFO Control Register (FCR, Offset = 0x08 for Write)

Bit	Name	Type	Function
[7:6]	RXFIFO_TRGL	W	These bits are used to set the trigger level of the RX FIFO interrupt. Please refer to Table 13-49.
[5:4]	TXFIFO_TRGL	W	These bits are used to set the trigger level of the TX FIFO interrupt. Please refer to Table 13-50.
3	DMA Mode	W	This bit selects the UART DMA mode. The DMA mode affects the behavior of the DMA signaling outputs pins (irda_nrxrdy and irda_ntxrdy).
2	TX FIFO Reset	W	Setting this bit to logic 1 clears all bytes in TX FIFO and resets the counter logic to '0'. The shift register is not cleared so that any reception active will continue. This bit will be automatically returned to zero.
1	RX FIFO Reset	W	Setting this bit to logic 1 clears all bytes in Rx FIFO and resets the counter logic to '0'. The shift register is not cleared so that any reception active will continue. Setting this bit also clears the status FIFO. This bit will be automatically returned to zero.
0	FIFO Enable	W	Set this bit to logic 1 enables both the transmit, receive, and status FIFOs. Changing this bit will automatically reset the transmit and receive FIFOs. In a FIR mode, the device driver should always set this bit as '1'.

Table 13-49. FIFO Trigger Level of Receiver

FCR Code		FIFO Trigger Level of 16-byte Receiver	FIFO Trigger Level of 32-byte Receiver
Bit 7	**Bit 6**		
0	0	1 character	1 character
0	1	4 characters	8 characters
1	0	8 characters	16 characters
1	1	14 characters	28 characters
FCR Code		**FIFO Trigger Level of 64-byte Receiver**	**FIFO Trigger Level of 128-byte Receiver**
Bit 7	**Bit 6**		
0	0	1 character	1 character
0	1	16 characters	32 characters
1	0	32 characters	64 characters
1	1	56 characters	120 characters

Table 13-50. FIFO Trigger Level of Transmitter

FCR Code		FIFO Trigger Level of 16-byte Transmitter	FIFO Trigger Level of 32-byte Transmitter
Bit 5	**Bit 4**		
0	0	1 character	1 character
0	1	3 characters	6 characters
1	0	9 characters	18 characters
1	1	13 characters	26 characters
FCR Code		**FIFO Trigger Level of 64-byte Transmitter**	**FIFO Trigger Level of 128-byte Transmitter**
Bit 5	**Bit 4**		
0	0	1 character	1 character
0	1	16 characters	32 characters
1	0	32 characters	64 characters
1	1	56 characters	120 characters

13.4.3.2.6 Line Control Register (LCR, Offset = 0x0C)

This register controls in a way that the transmitted characters are serialized and the received characters are assembled and checked.

Table 13-51. Line Control Register (LCR, Offset = 0x0C)

Bit	Name	Type	Function
7	DLAB	R/W	Divisor Latch Access Bit (DLAB) This bit must be set in order to access the DLL, DLM, and PSR registers, which program the division constants for the baud rate divider and prescaler.
6	Set Break	R/W	This bit causes a break condition to be transmitted to the receiving UART. When it is set to logic 1, the serial output (**io_irda_sout**) is forced to the Spacing (Logic 0) state. The break is disabled by setting bits[6:0]. The Break Control bit acts only on **io_irda_sout** and has no effect on the transmitter logic, so if several characters are stored in the transmit FIFO, they will be removed from this FIFO and passed sequentially to the Transmitter Shift Register which serializes them, even if Set Break is set. This fact can be useful to establish the break time making use of the THR Empty and Transmitter Empty flags of the LSR. Firmware can follow the sequence below to assure no erroneous or extraneous characters will be transmitted because of the break: Set break when transmitter is idle (LSR bit[6]). Write a character with any value to THR. Wait for the transmitter to become idle (LSR bit[6]), and clear break when normal transmission has to be restored.
5	Stick Parity	R/W	When bits 3, 4, and 5 are logic 1, the Parity bit is transmitted and checked as 0. If bits 3 and 5 are 1 and bit 4 is 0, the Parity bit is transmitted and checked as 1. If bit 5 is 0, Stick Parity is disabled. Please refer to Table 13-52.
4	Even Parity	R/W	This bit is the Even Parity Select bit. When bit 3 is 1 and bit 4 is 0, an odd number of logic 1s is transmitted or checked in the data word bits and Parity bit. Please refer to Table 13-52.
3	Parity Enable	R/W	This bit is the Parity Enable bit. When this bit is a 1, a Parity bit is generated (Transmit data) or checked (Receive data) between the last data word bit and Stop bit of the serial data. When bit 3 is 1 and bit 4 is a 1, an even number of 1s is transmitted or checked. Please refer to Table 13-52.
2	Stop Bits	R/W	This bit selects the number of stop bits to be transmitted. If cleared, only one stop bit will be transmitted. If set, two stop bits (1.5 with 5bit data) will be transmitted before the start bit of the next character. The receiver always checks only one stop bit. Please refer to Table 13-53.
1	WL1	R/W	This bit along with WL0 defines the word length of the data being transmitted and received. Please refer to Table 13-53 for the possible selections.
0	WL0	R/W	This bit along with WL1 defines the word length of the data being transmitted and received. Please refer to Table 13-53 for the possible selections.

Table 13-52. Parity Setting

LCR Code			Parity Bit (Transmitted or Checked)
Bit 5 Stick Parity	Bit 4 Even Parity	Bit 3 Parity Enable	
X	X	0	Not transmitted or checked
0	1	1	Even parity
0	0	1	Odd parity
1	0	1	1
1	1	1	0

Table 13-53. Word Length and Stop Bit Setting

LCR Code			Character Length (Bit)	Stop Bit
Bit 2 (Stop Bit)	Bit 1 (WL1)	Bit 0 (WL0)		
0	0	0	5	1
	0	1	6	
	1	0	7	
	1	1	8	
1	0	0	5	1.5
	0	1	6	2
	1	0	7	
	1	1	8	

13.4.3.2.7 Modem Control Register (MCR, Offset = 0x10)

By writing to this register, users can set the modem control outputs (**io_irda_ndtr** and **io_irda_nrts**). This register also controls the loopback mode and provides the general-purpose outputs.

Table 13-54. Modem Control Register r (MCR, Offset = 0x10)

Bit	Name	Type	Function
7	Reserved	-	-
6	Out3	R/W	This bit controls the general-purpose active-low output, io_irda_nout3, in the same way as bit 0 controls io_irda_ndtr. This bit is not found in the standard 16550 UART.
5	DMAmode2	R/W	This bit selects the UART/SIR DMA mode. The DMA mode2 affects the way in which the DMA signaling output pins (irda_nrxrdy and irda_ntxrdy) behave.

Bit	Name	Type	Function
4	Loop	R/W	Loopback mode control bit Loopback mode is intended to test the UART or SIR communication.
3	Out2	R/W	This bit controls the general-purpose, active-low output, io_irda_nout2, in the same way as bit 0 controls io_irda_ndtr.
2	Out1	R/W	This bit controls the general-purpose, active-low output, io_irda_nout1, in the same way as bit 0 controls io_irda_ndtr.
1	RTS (Request to Send)	R/W	This bit controls the request-to-send, active-low output, io_irda_nrts, in the same way as bit 0 controls io_irda_ndtr.
0	DTR (Data Terminal Ready)	R/W	This bit controls the data-terminal-ready, active-low output, io_irda_ndtr. A 1 in this bit makes the io_irda_ndtr output a 0. When this bit is cleared, io_irda_ndtr will output a 1.

When UART is set in the loopback mode, the following items will occur:

- The serial output will be internally connected to the serial input even that the sent character is looped back and received.
- The input pin, io_irda_sin, is not used and the output pin, io_irda_sout, is set to '1' (Inactive state).
- Four modem control inputs are internally connected to two modem control outputs plus the general-purpose outputs. In such a way, io_irda_ncts will be internally controlled by io_irda_nrts, io_irda_ndsr is controlled by io_irda_ndtr, io_irda_nri is controlled by io_irda_nout1, and io_irda_ndcd is controlled by io_irda_nout2. That is to say; there is a non-ordered correspondence between four least significant bits of MCR and four most significant bits of MSR. Please refer to Figure 13-2 for connections. The modem control output pins are forced to the inactive state (High).
- Four modem control input pins, io_irda_ncts, io_irda_ndsr, io_irda_nri, and io_irda_ndcd, are not used. Two modem control output pins, io_irda_ndtr and io_irda_nrts, and three user outputs, io_irda_nout1, io_irda_nout2, and io_irda_nout3, are set to '1' (Inactive state).

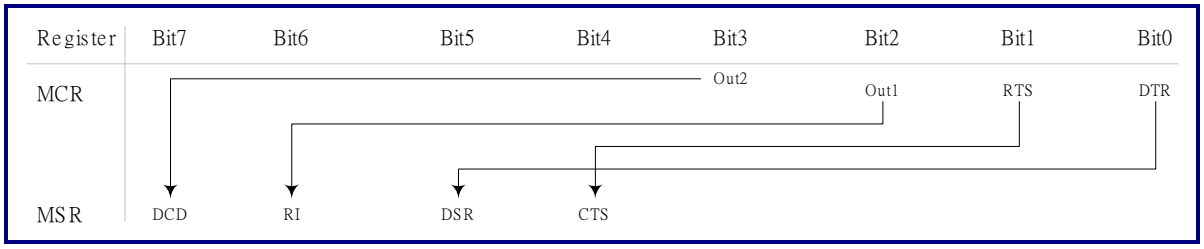


Figure 13-2. Interconnection between MCR and MSR in Loopback Mode

13.4.3.2.8 Line Status Register (LSR, Offset = 0x14 for Read)

This register informs users about the status of the transmitter and receiver. In order to get information about a received characters, LSR must be read before reading the received characters from RBR.

Table 13-55. Line Status Register (LSR, Offset = 0x14 for Read)

Bit	Name	Type	Function
7	FIFO Data Error	R	If FIFO is disabled (16450 mode), this bit will always be zero. If FIFO is active, this bit will be set as soon as any data character in the receive FIFO has parity or framing error or the break indication active. This bit will be cleared when the CPU reads LSR and the rest of data in the receive FIFO do not have any of these three associated flags on.
6	Transmitter Empty	R	This bit will be '1' when both THR (Or TX FIFO) and TSR (Transmitter Shift Register) are empty. Reading this bit as '1' means that no transmission is currently taking place in the io_irda_sout output pin, and that the transmission line is idle. As soon as new data is written in THR, this bit will be cleared.
5	THR Empty	R	This bit indicates that UART is ready to accept a new character for transmission. In addition, this bit causes UART to issue an interrupt to the CPU when the Transmit Holding Register Empty Interrupt enable bit (IER [1]) is set to high. In the non-FIFO mode, this bit will be set when the 1-byte THR is empty. If THR holds data to be transmitted, this bit will immediately be set when data is passed to TSR. In the FIFO mode, this bit will be set when the transmit FIFO is completely empty. This be will be '0' if there is at least one byte in FIFO that is waiting to be passed to TSR for the transmission.
4	Break Interrupt	R/C	This bit will be set to '1' if the receiver line input, io_irda_sin, is held as '0' for a complete character time. That is to say, the positions corresponding to the start bit, data, parity bit (If any), and the first stop bit were all detected as '0'. Please note that a Framing Error flag always accompanies this flag. This bit is queued in the receive FIFO in the same way as the Parity Error bit. When break occurs, only one zero character will be loaded into FIFO. The next character transfer will be enabled after io_irda_sin goes to the marking state and receives the next valid start bit. This bit will be cleared as soon as LSR is read.
3	Framing Error	R	This bit indicates that the received character does not have a valid stop bit (i.e., a 0 was detected in the (First) stop bit position instead of a 1). This bit is queued in the receive FIFO in the same way as the Parity Error bit. When a framing error is detected, the receiver tries to resynchronize. If the next sample is again a zero, it will be taken as the beginning of a possible new start bit. This bit will be cleared as soon as LSR is read.

Bit	Name	Type	Function
2	Parity Error	R	When this bit is set, it indicates that the parity of the received character is wrong according to the current setting in LCR. This bit is queued in the receive FIFO, so it is associated with particular character that had the error. Therefore, LSR must be read before RBR. Each time a character is read from RBR, the next character passes to the top of FIFO and LSR is loaded with the queued error flags corresponding to the top-of-the-FIFO character. This bit will be cleared as soon as LSR is read.
1	Overrun Error	R	When this bit is set, a character will be completely assembled in the Receiver Shift Register without free space to put the character in the receive FIFO or holding register. When an overrun condition appears, the result will be different depending on the activation of 16byte FIFO. When FIFO is not active, only a 1-character Receiver Holing Register will be available. The unread data in RBR will not be overwritten with new character received. When FIFO is active, the character received in the Receiver Shift Register will be overwritten, but the data already presented in FIFO will not be changed. The Overrun Error flag will be set as soon as the overrun condition appears. This bit is not queued in FIFO if it is active. This bit is cleared as soon as LSR is read.
0	Data Ready	R	This bit will be set if one or more characters have been received and are waiting in the receive FIFO to be accessed by users. It is cleared to logic 0 by reading all data in the Receiver Buffer Register or FIFO.

13.4.3.2.9 Testing Register (TST, Offset = 0x14 for Write)

This register will provide the internal diagnostic capabilities if the circuit is hardware implemented. The loopback mode is supported only in the UART and SIR modes. For FIR, because there is only one CRC module in the UART controller, FIR transmit and receive cannot be tested with the loopback mode.

Table 13-56. Testing Register (TST, Offset = 0x14 for Write)

Bit	Name	Type	Function
[7:5]	Reserved	-	-
4	TEST_CRC_ERR	W	When this bit is set, the UART controller will generate incorrect CRC during the FIR transmit.
3	TEST_PHY_ERR	W	When this bit is set, the UART controller will generate the incorrect 4PPM encoding chips during the FIR transmit.
2	TEST_BAUDGEN	W	This bit is used to improve the toggle rate of the baud rate generator.
1	TEST_FRM_ERR	W	When this bit is set, the UART controller will generate the logic 0 STOP bit during the UART transmit.

Bit	Name	Type	Function
0	TEST_PAR_ERR	W	When this bit is set, the UART controller will generate incorrect parity during the UART transmit.

13.4.3.2.10 Modem Status Register (MSR, Offset = 0x18)

This register provides information about the status of four modem-control input pins. Four most significant bits directly provide the statuses of these pins, while four least significant bits give information about the changes of these pins.

Four least significant bits can generate an interrupt (Modem Status interrupt) if it is enabled by the corresponding bit in the IER register. The interrupt will be generated as soon as any of these four significant bits is '1'. They are reset to logic 0 when the Modem Status Register is read.

Table 13-57. Modem Status Register (MSR, Offset = 0x18)

Bit	Name	Type	Function
7	DCD	R	Data Carrier Detect (DCD) is the complement of the io_irda_ndcd input.
6	RI	R	Ring Indicator (RI) is the complement of the io_irda_nri input.
5	DSR	R	Data Set Ready (DSR) is the complement of the io_irda_ndsr input.
4	CTS	R	Clear To Send (CTS) is the complement of the io_irda_ncts input.
3	Delta DCD	R	The delta-DCD flag If this bit is set, it means that the io_irda_ndcd input will be changed since last time the microprocessor read this bit
2	Trailing edge R1	R	This bit will be set when a trailing edge is detected in the io_irda_nri input pin when io_irda_nri changes from '0' to '1'.
1	Delta DSR	R	If this bit is set, it means that the io_irda_ndsr input will be changed since the last time the microprocessor read this bit.
0	Delta CTS	R	If this bit is set, it means that the io_irda_ncts input will be changed since the last time the microprocessor read this bit.

13.4.3.2.11 Scratch Pad Register (SPR, Offset = 0x1C)

Table 13-58. Scratch Pad Register (SPR, Offset = 0x1C)

Bit	Name	Type	Function
[7:0]	User Data	R/W	This 8bit read/write register has no effect on the operation of the serial port. It is intended as a scratchpad register to be used by programmers to temporarily hold data.

13.4.3.2.12 Baud Rate Divisor Latch (DL, Offset = 0x00, 0x04 when DLAB = 1)

This 16bit register holds the most significant byte in DLM and holds the least significant byte in DLL. The division factors can be programmed from 1 to 65535. To access these two registers, located at addresses 1 and 0 respectively, and is conditioned based on the value of the DLAB bit in the LCR register. These two registers can be written and read only if this bit is set to '1'. Otherwise, IER, RBR, and THR will be accessed.

These two registers (DLL and DLM) with the Prescaler Register (PSR) can select the speed when the communication occurs. This is the baud rate when the characters are transmitted and the expected baud rate for the characters are received. Only one baud rate can be defined for both the transmission and reception.

The baud rate is defined as the io_irda_uclk frequency divided by 16, divided by the contents of the PSR register, and divided by the contents of the Baud Rate Divisor Latch register. When DLM and DLL are programmed as zeroes, there will not be the output clock. It is recommended programing DLL and DLM as zeroes for saving power in the FIR mode.

Table 13-59. Baud Rate Divisor Latch LSB

Bit	Name	Type	Function
[7:0]	DLL	R/W	Baud Rate Divisor Latch Least Significant Byte

Table 13-60. Baud Rate Divisor Latch MSB

Bit	Name	Type	Function
[7:0]	DLM	R/W	Baud Rate Divisor Latch Most Significant Byte

13.4.3.2.13 Prescaler Register (PSR, Offset = 0x08 when DLAB = 1)

This 5bit register (PSR[4:0]) adds a second programmable division factor to obtain the desired baud rate (Please refer to the description of the Divisor Latch Register). Because the division factor is the value hold in this register, the maximum factor will be 31 and the minimum factor will be 0. Bits 5 to 7 are always zeroes. This is a non-standard register (It is not presented in the industry standard 16550 UART). The input clock, io_irda_uclk, is divided by an integer ranging from 1 to 31. When PSR is '0', there will not be the input clock to the divisor latch unit. Therefore, programming DLL and DLM are useless when PSR is set to '0'. The default value for the PSR register is '1'. This register can only be accessible when the DLAB bit in LCR is set. Otherwise, the Line Status Register will be accessed.

Table 13-61. Prescaler Register (PSR, Offset = 0x08 when DLAB = 1)

Bit	Name	Type	Function
[7:5]	Reserved	-	-
[4:0]	PSR	R/W	Prescaler Value

13.5 PWM Controller

13.5.1 General Description

PWM provides up to eight independent sets of timers. Each timer can use the internal system clock (PCLK of APB) or external clock. These timers can be used to generate the internal interrupts to CPU. They can also be used to trigger the DMA transfers. In addition, each timer supports the Pulse Width Modulation (PWM) function, which can generate the PWM signals for the motor control or power-level control.

13.5.2 Features

- Supports up to 8 independent 32bit timers
- Internal or external clock source selection for each timer
- Supports PWM generation for each timer
- Programmable duty control and polarity of PWM
- Supports auto-reload mode and one-shot pulse mode
- Dead-zone generator on PWM outputs
- Supports timer based on DMA transfer request

13.5.3 Programming Model

13.5.3.1 Summary of Timer Registers

Table 13-62 lists and describes the offsets, types, widths, reset values, names, and descriptions of the timer registers. Besides, the global registers, INT_STAT, INT_CTRL, and TMR_REV, are shared by all timers and each timer has its own register set. The settings of these registers depend on the presence of the corresponding timers.

Table 13-62. Summary of Timer Registers

Offset	Name	Type	Width	Reset	Description
0x000	INT_CSTAT	R/WC	8	0x0	Interrupt status and control register
0x004	START_CTRL	R/W	8	0x0	Timers 1 ~ 8 start control register
0x010	T1_CTRL	R/W	16	0x0	Timer 1 control register
0x014	T1_CNTB	R/W	32	0x0	Timer 1 count buffer register
0x018	T1_CMPB	R/W	32	0x0	Timer 1 compare buffer register
0x01C	T1_CNTO	R	32	0x0	Timer 1 observation register
0x020	T2_CTRL	R/W	16	0x0	Timer 2 control register
0x024	T2_CNTB	R/W	32	0x0	Timer 2 count buffer register
0x028	T2_CMPB	R/W	32	0x0	Timer 2 compare buffer register
0x02C	T2_CNTO	R	32	0x0	Timer 2 observation register
0x030	T3_CTRL	R/W	16	0x0	Timer 3 control register
0x034	T3_CNTB	R/W	32	0x0	Timer 3 count buffer register
0x038	T3_CMPB	R/W	32	0x0	Timer 3 compare buffer register
0x03C	T3_CNTO	R	32	0x0	Timer 3 observation register
0x040	T4_CTRL	R/W	16	0x0	Timer 4 control register
0x044	T4_CNTB	R/W	32	0x0	Timer 4 count buffer register
0x048	T4_CMPB	R/W	32	0x0	Timer 4 compare buffer register
0x04C	T4_CNTO	R	32	0x0	Timer 4 observation register
0x050	T5_CTRL	R/W	16	0x0	Timer 5 control register
0x054	T5_CNTB	R/W	32	0x0	Timer 5 count buffer register
0x058	T5_CMPB	R/W	32	0x0	Timer 5 compare buffer register
0x05C	T5_CNTO	R	32	0x0	Timer 5 observation register
0x060	T6_CTRL	R/W	16	0x0	Timer 6 control register
0x064	T6_CNTB	R/W	32	0x0	Timer 6 count buffer register
0x068	T6_CMPB	R/W	32	0x0	Timer 6 compare buffer register

Offset	Name	Type	Width	Reset	Description
0x06C	T6_CNTO	R	32	0x0	Timer 6 observation register
0x070	T7_CTRL	R/W	16	0x0	Timer 7 control register
0x074	T7_CNTB	R/W	32	0x0	Timer 7 count buffer register
0x078	T7_CMPB	R/W	32	0x0	Timer 7 compare buffer register
0x07C	T7_CNTO	R	32	0x0	Timer 7 observation register
0x080	T8_CTRL	R/W	16	0x0	Timer 8 control register
0x084	T8_CNTB	R/W	32	0x0	Timer 8 count buffer register
0x088	T8_CMPB	R/W	32	0x0	Timer 8 compare buffer register
0x08C	T8_CNTO	R	32	0x0	Timer 8 observation register
0x090	TMR_REV	R	32	0x100	FTTMR020 revision
0x0a0	T1_REP_CNTB	R/W	10	0x0	Timer 1 repeat/ pattern mode count buffer register
0x0a4	T2_REP_CNTB	R/W	10	0x0	Timer 2 repeat/pattern mode count buffer register
0x0a8	T3_REP_CNTB	R/W	10	0x0	Timer 3 repeat/pattern mode count buffer register
0x0ac	T4_REP_CNTB	R/W	10	0x0	Timer 4 repeat/pattern mode count buffer register
0x0b0	T5_REP_CNTB	R/W	10	0x0	Timer 5 repeat/pattern mode count buffer register
0x0b4	T6_REP_CNTB	R/W	10	0x0	Timer 6 repeat/pattern mode count buffer register
0x0b8	T7_REP_CNTB	R/W	10	0x0	Timer 7 repeat/pattern mode count buffer register
0x0bc	T8_REP_CNTB	R/W	10	0x0	Timer 8 repeat/pattern mode count buffer register
0x0c0	T1_PAT0	R/W	32	0x0	Timer1 pattern-mode data pattern [31:0]
0x0c4	T1_PAT1	R/W	32	0x0	Timer1 pattern-mode data pattern [63:32]
0x0c8	T1_PAT2	R/W	32	0x0	Timer1 pattern-mode data pattern [95:64]
0x0cc	T1_PAT3	R/W	32	0x0	Timer1 pattern-mode data pattern [127:96]
0x0d0	T2_PAT0	R/W	32	0x0	Timer2 pattern-mode data pattern [31:0]
0x0d4	T2_PAT1	R/W	32	0x0	Timer2 pattern-mode data pattern [63:32]
0x0d8	T2_PAT2	R/W	32	0x0	Timer2 pattern-mode data pattern [95:64]
0x0dc	T2_PAT3	R/W	32	0x0	Timer2 pattern-mode data pattern [127:96]
0x0e0	T3_PAT0	R/W	32	0x0	Timer3 pattern-mode data pattern [31:0]
0x0e4	T3_PAT1	R/W	32	0x0	Timer3 pattern-mode data pattern [63:32]
0x0e8	T3_PAT2	R/W	32	0x0	Timer3 pattern-mode data pattern [95:64]
0x0ec	T3_PAT3	R/W	32	0x0	Timer3 pattern-mode data pattern [127:96]
0x0f0	T4_PAT0	R/W	32	0x0	Timer4 pattern-mode data pattern [31:0]
0x0f4	T4_PAT1	R/W	32	0x0	Timer4 pattern-mode data pattern [63:32]
0x0f8	T4_PAT2	R/W	32	0x0	Timer4 pattern-mode data pattern [95:64]
0x0fc	T4_PAT3	R/W	32	0x0	Timer4 pattern-mode data pattern [127:96]
0x100	T5_PAT0	R/W	32	0x0	Timer5 pattern-mode data pattern [31:0]
0x104	T5_PAT1	R/W	32	0x0	Timer5 pattern-mode data pattern [63:32]

Offset	Name	Type	Width	Reset	Description
0x108	T5_PAT2	R/W	32	0x0	Timer5 pattern-mode data pattern [95:64]
0x10c	T5_PAT3	R/W	32	0x0	Timer5 pattern-mode data pattern [127:96]
0x110	T6_PAT0	R/W	32	0x0	Timer6 pattern-mode data pattern [31:0]
0x114	T6_PAT1	R/W	32	0x0	Timer6 pattern-mode data pattern [63:32]
0x118	T6_PAT2	R/W	32	0x0	Timer6 pattern-mode data pattern [95:64]
0x11c	T6_PAT3	R/W	32	0x0	Timer6 pattern-mode data pattern [127:96]
0x120	T7_PAT0	R/W	32	0x0	Timer7 pattern-mode data pattern [31:0]
0x124	T7_PAT1	R/W	32	0x0	Timer7 pattern-mode data pattern [63:32]
0x128	T7_PAT2	R/W	32	0x0	Timer7 pattern-mode data pattern [95:64]
0x12c	T7_PAT3	R/W	32	0x0	Timer7 pattern-mode data pattern [127:96]
0x130	T8_PAT0	R/W	32	0x0	Timer8 pattern-mode data pattern [31:0]
0x134	T8_PAT1	R/W	32	0x0	Timer8 pattern-mode data pattern [63:32]
0x138	T8_PAT2	R/W	32	0x0	Timer8 pattern-mode data pattern [95:64]
0x13c	T8_PAT3	R/W	32	0x0	Timer8 pattern-mode data pattern [127:96]
0x140	T1_PAT_LEN	R/W	7	0x0	Timer1 pattern-mode output (length+1) bits in the 128bit pattern register
0x144	T2_PAT_LEN	R/W	7	0x0	Timer2 pattern-mode output (length+1) bits in the 128bit pattern register
0x148	T3_PAT_LEN	R/W	7	0x0	Timer3 pattern-mode output (length+1) bits in the 128bit pattern register
0x14c	T4_PAT_LEN	R/W	7	0x0	Timer4 pattern-mode output (length+1) bits in the 128bit pattern register
0x150	T5_PAT_LEN	R/W	7	0x0	Timer5 pattern-mode output (length+1) bits in the 128bit pattern register
0x154	T6_PAT_LEN	R/W	7	0x0	Timer6 pattern-mode output (length+1) bits in the 128bit pattern register
0x158	T7_PAT_LEN	R/W	7	0x0	Timer7 pattern-mode output (length+1) bits in the 128bit pattern register
0x15c	T8_PAT_LEN	R/W	7	0x0	Timer8 pattern-mode output (length+1) bits in the 128bit pattern register
0x160	INT_ID	R	32	0x0	Timer1 ~ Timer8 priority interrupt ID

13.5.3.2 Register Description

The following subsections describe the timer registers in details.

13.5.3.2.1 Interrupt Status and Control Register (INT_CSTAT)

The INT_CSTAT register shows the interrupt status of each timer. If the global interrupt of PWM is used by the system, CPU will check the timers that issue the interrupts by reading the INT_CSTAT register. The interrupt enable/disable control of each timer can also be set in this register.

Table 13-63. Interrupt Status and Control Register (INT_CSTAT)

Bit	Name	Type	Reset	Description
0	TM1_INT_STAT	R/WC	0	Timer 1 interrupt status Clear by writing '1' on this bit 0: No effect 1: Clear
1	TM2_INT_STAT	R/WC	0	Timer 2 interrupt status Clear by writing '1' on this bit 0: No effect 1: Clear
2	TM3_INT_STAT	R/WC	0	Timer 3 interrupt status Clear by writing '1' on this bit 0: No effect 1: Clear
3	TM4_INT_STAT	R/WC	0	Timer 4 interrupt status Clear by writing '1' on this bit 0: No effect 1: Clear
4	TM5_INT_STAT	R/WC	0	Timer 5 interrupt status Clear by writing '1' on this bit 0: No effect 1: Clear

Bit	Name	Type	Reset	Description
5	TM6_INT_STAT	R/WC	0	Timer 6 interrupt status Clear by writing ‘1’ on this bit 0: No effect 1: Clear
6	TM7_INT_STAT	R/WC	0	Timer 7 interrupt status Clear by writing ‘1’ on this bit 0: No effect 1: Clear
7	TM8_INT_STAT	R/WC	0	Timer 8 interrupt status Clear by writing ‘1’ on this bit 0: No effect 1: Clear
31:8	-	-	-	Reserved.

13.5.3.2.2 Timer Start Control Register (START_CTRL)

This register controls the start/stop behavior of timers 1 ~ 8.

Table 13-64. Timer Start Control Register (START_CTRL)

Bit	Name	Type	Reset	Description
0	TM1_START	R/W	0	Timer 1 start/stop 0: Stop 1: Start
1	TM2_START	R/W	0	Timer 2 start/stop 0: Stop 1: Start
2	TM3_START	R/W	0	Timer 3 start/stop 0: Stop 1: Start
3	TM4_START	R/W	0	Timer 4 start/stop 0: Stop 1: Start

Bit	Name	Type	Reset	Description
4	TM5_START	R/W	0	Timer 5 start/stop 0: Stop 1: Start
5	TM6_START	R/W	0	Timer 6 start/stop 0: Stop 1: Start
6	TM7_START	R/W	0	Timer 7 start/stop 0: Stop 1: Start
7	TM8_START	R/W	0	Timer 8 start/stop 0: Stop 1: Start
31:8	-	-	-	Reserved.

13.5.3.2.3 Timer Control Register (TMX_CTRL, X = 1, 2, …, 8)

This register controls the behavior of a timer, including start, update, invert output, auto-reload, DMA, and dead-zone.

Table 13-65. Timer Control Register (TMX_CTRL, X = 1, 2, …, 8)

Bit	Name	Type	Reset	Description
0	TM**X**_CLK_SRC	R/W	0	Timer**X** clock source 0: PCLK 1: ext_clk
1	TM**X**_UPDATE	R/W	0	TimerX updates TM**X**_CNTB and TM**X**_CMPB. Write '1' to this register will trigger a manual update. This bit will be automatically cleared after the operation. 0: No operation 1: Manual updates
2	TM**X**_OUT_INV	R/W	0	TimerX output inverter on/off 0: Inverter off 1: Inverter on Note: The pattern mode will force the inverter to be off.

Bit	Name	Type	Reset	Description
4:3	TM**X**_SET_MODE	R/W	0	Timer**X** setup mode 00: One shot 01: Interval mode (Auto-reload) 10: Repeat mode (Need setup TM**X**_REP_CNTB) 11: Pattern mode (Need setup TM**X**_REP_CNTB)
5	TM**X**_INT_EN	R/W	0	Timer**X** Interrupt mode enable/disable 0: Disable 1: Enable
6	TM**X**_INT_MO	R/W	0	Timer**X** Interrupt mode 0: Level (Clear the interrupt by writing '1' on TM**X**_INT_STAT) 1: Pulse
7	TM**X**_DMA_EN	R/W	0	Timer**X** DMA request mode enable/disable 0: Disable 1: Enable
8	TM**X**_PWM_EN	R/W	0	Timer**X** PWM function enable/disable 0: Disable 1: Enable
23:9	-	-	-	Reserved
31:24	TM**X**_DZ	R/W	0	Timer**X** dead-zone period

13.5.3.2.4 Timer Counting Value Buffer Register (TMX_CNTB, X = 1, 2, …, 8)

This register will store the start value of the counter. The counter in the timer will load this value when:

1. Manual update is triggered by TM**X**_CTRL[TM**X**_UPDATE].
2. Auto-reload is enabled and the counter value reaches zero.

Table 13-66. Timer Counting Value Buffer Register (TMX_CNTB, X = 1, 2, …, 8)

Bit	Name	Type	Reset	Description
31:0	TM**X**_CNTB	R/W	0	Start value of the Timer**X** counter

13.5.3.2.5 Timer Counter Compare Buffer Register (TMX_CMPB, X = 1, 2, …, 8)

This register stores the comparison value of the counter. When the counter value equals to the comparison value, the level of the timer output, tmr**X**_out, will be changed. The comparator in the timer will load this value when:

- Manual update is triggered by TM**X**_CTRL[TM**X**_UPDATE].
- Auto-reload is enabled and the counter value reaches zero.

Table 13-67. Timer Counter Compare Buffer Register (TMX_CMPB, X = 1, 2, …, 8)

Bit	Name	Type	Reset	Description
[31:0]	TM**X**_CMPB	R/W	0	Compare value of Timer**X**

13.5.3.2.6 Timer Count Observation Register (TMX_CNTO, X = 1, 2, …, 8)

CPU can use this register to observe the counter value.

Table 13-68. Timer Count Observation Register (TMX_CNTO, X = 1, 2, …, 8)

Bit	Name	Type	Reset	Description
[31:0]	TM**X**_CNTO	R	0	Compare value of Timer**X**

13.5.3.2.7 Revision Register (TMR_REV)

Users can use this register to get the revision of TIMER.

Table 13-69. Revision Register (TMR_REV)

Bit	Name	Type	Reset	Description
[31:0]	REV	R	0x100	Revision of FTTMR020

13.5.3.2.8 Timer Count for Repeat/Pattern Mode Register (TMX_REP_CNTB, X = 1, 2, …, 8)

Users can use this register to setup the repeat/pattern mode.

Table 13-70. Timer Count for Repeat/Pattern Mode Register (TMX_REP_CNTB, X = 1, 2, …, 8)

Bit	Name	Type	Reset	Description
[15:0]	TM**X**_REP_CNTB	R/W	0x0	Timer repeat/pattern mode count buffer is used to count the (TM**X**_REP_CNTB+1) times. When reading this value, it will send the interrupt in the repeat/pattern mode.

13.5.3.2.9 Timer Pattern Mode Data Register (TMX_PAT0 ~ 3, X = 1, 2, …, 8)

Users can use this register to setup the pattern-mode data in 128bits.

Table 13-71. Timer Pattern Mode Data Register (TMX_PAT0 ~ 3, X = 1, 2, …, 8)

Bit	Name	Type	Reset	Description
[31:0]	TM**X**_PAT0	R/W	0x0	Timer**X** data pattern [31:0]
[31:0]	TM**X**_PAT1	R/W	0x0	Timer**X** data pattern [63:32]
[31:0]	TM**X**_PAT2	R/W	0x0	Timer**X** data pattern [95:64]
[31:0]	TM**X**_PAT3	R/W	0x0	Timer**X** data pattern [127:96]

13.5.3.2.10 Timer Pattern Mode Output Length Register (TMX_PAT_LEN, X = 1, 2, …, 8)

Users can used this register to setup the output length of the pattern mode.

Table 13-72. Timer Pattern Mode Output Length Register (TMX_PAT_LEN, X = 1, 2, …, 8)

Bit	Name	Type	Reset	Description
[6:0]	TM**X**_PAT_LEN	R/W	0x0	Timer**X** pattern-mode output (Length + 1) bits in the 128bit pattern register.

13.5.3.2.11 Timer1 ~ Timer8 Interrupt Priority ID Register (INT_ID)

Based on the interrupt priority ID, users can update new parameters in order. If the interrupt occurs, it will look this register to identify which timer interrupt occurred.

Table 13-73. Timer1 ~ Timer8 Interrupt Priority ID Register (INT_ID)

Bit	Name	Type	Reset	Description
[3:0]	INT_ID	R	0x0	Which timer interrupt (1 ~ 8) occurs.

13.5.4 Function Description

13.5.4.1 Auto-reload and Double Buffering

The timers have a double buffering feature, which can change the reload value for the next timer operation without stopping the current timer operation. Consequently, a current timer operation can be completed successfully even if the new time value is set.

The timer value can be written into TM**X**_CNTB (Timer count buffer register) and the current counter value of the timer can be read from TM**X**_CNTO (Timer count observation register). When TM**X**_CNTB reads, the read value will be the reload value for the next timer duration, not the current state of the counter.

The auto-reload feature is the operation that copies TM**X**_CNTB into the counter when the counter reaches '0' and when the auto-reload feature is enabled. Under this condition, the counter will reach '0' and the Auto-reload feature will be disabled, and the counter will stop operating. This is called "one-shot operation".

Figure 13-3 shows an example of double buffering feature.

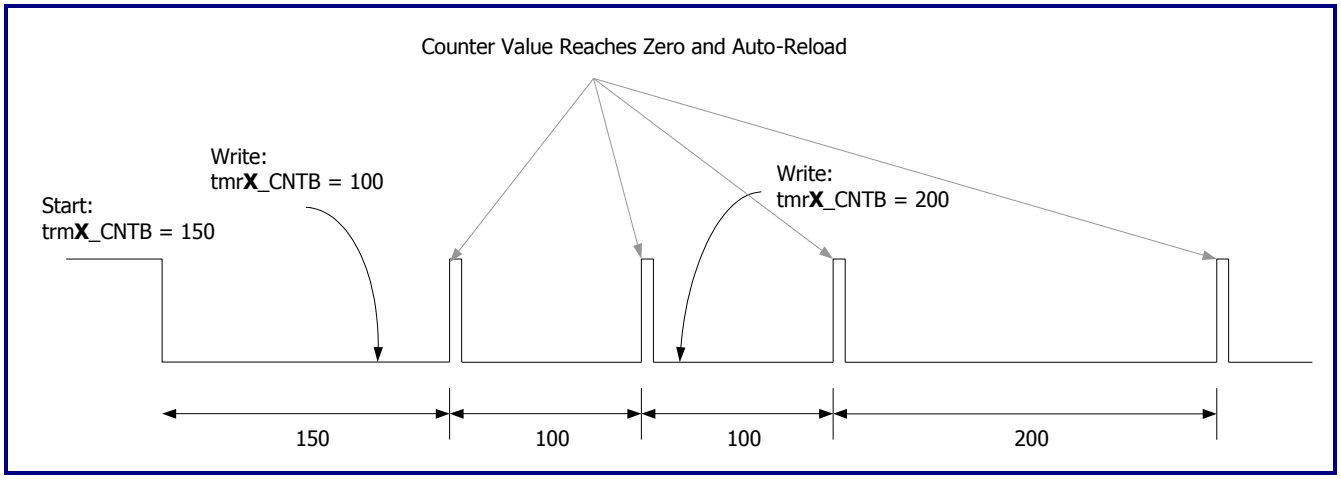


Figure 13-3. Double Buffering Feature

13.5.4.2 PWM Function

The PWM function can be implemented by using TM**X**_CMPB. When the output inverter, TM**X**_CTRL[TM**X**_OUT_INV], is disabled, the timer output, tmr**X**_out, will go to low level whenever the manual update or auto-reload of the counter occurs. The output will go to high level whenever the counter value equals to the compare value. This mechanism implies that the PWM period is determined by TM**X**_CNTB and the pulse width is determined by TM**X**_CMPB.

To obtain the higher PWM value, users should increase the TM**X**_CMPB value. To obtain the lower PWM values, users should decrease the TM**X**_CMPB value. However, if the output inverter is enabled, the increment/decrement may be opposite.

Because of the double buffering feature, TM**X**_CMPB, the next PWM cycle can be written into any point of the current PWM cycle by ISR.

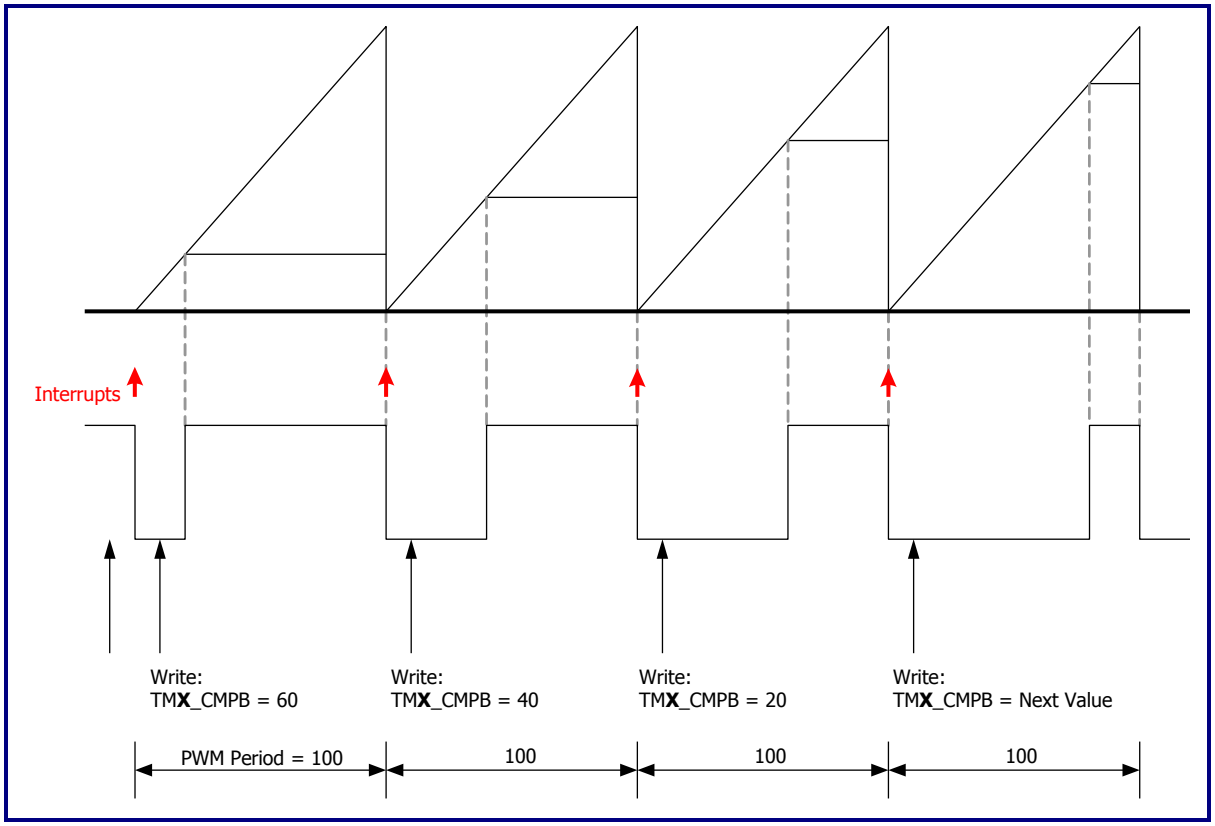


Figure 13-4. Example of PWM

13.5.4.3 Dead-zone Generation

The dead-zone is for the PWM control of a power device. This feature is used to insert the time gap between a turning-off switching device and other turning-on switching device. This time gap prohibits the two switching devices to be simultaneously turned on even for a short period of time.

Figure 13-5 shows the waveform when the dead-zone feature is enabled, where tmr**X**_out is the PWM output and tmr**X**_out_n is the inversion of tmr**X**_out. When the dead-zone feature is enabled, the output waveforms of tmr**X**_out and tmr**X**_out_n will be tmr**X**_out_DZ and tmr**X**_out_n_DZ. These two signals can never be simultaneously turned on by the dead-zone interval. The dead-zone period can be specified by the register, TM**X**_DZ (TM**X**_CFG[31:24]). The period is the setting value minus 1. The dead-zone function can be disabled by setting the value to '0' or '1'. For the functional correctness, the dead-zone period must be set to be smaller than the compare counter value; otherwise, the pulses with a pulse width of less than the dead-zone will vanish.

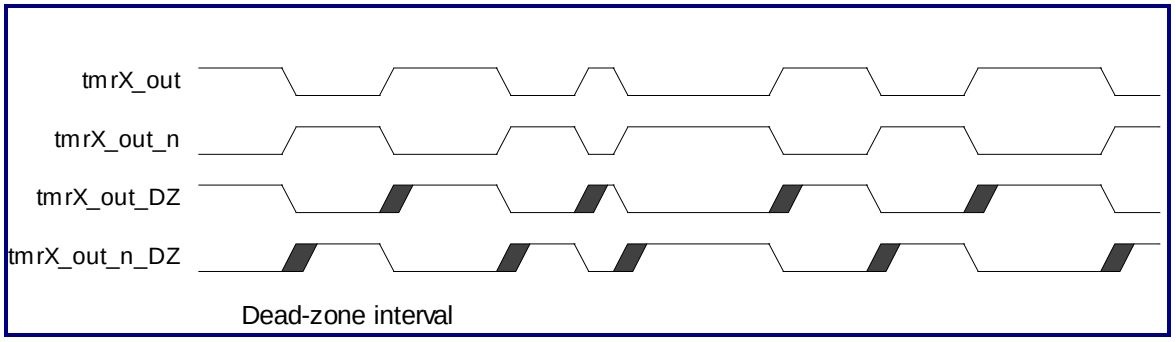


Figure 13-5. Waveform of Enabled Dead-zone Feature

13.5.4.4 DMA Request Mode

Instead of sending an interrupt at the end of the down-counter cycle, a timer can be configured to send a DMA request signal to one of the DMA channels. This mode allows the occurrence of the DMA transfers between a source and a destination at the regular intervals. The idea is not to use the timer as the source or destination, but rather using the timer to simply the timing control of the data movements between two other devices.

The DMA request signal, tmr**X**_req, will be active high by the timer until the timer receives an ACK signal from the DMA unit. When the timer receives an ACK signal, the request signal will become inactive.

If the timer is in the enabled DMA request mode, it will not issue any interrupt, even when the interrupt mode is turned on. Figure 13-6 shows how the DMA request remains active before the ACKJ signal is sent by DMA.

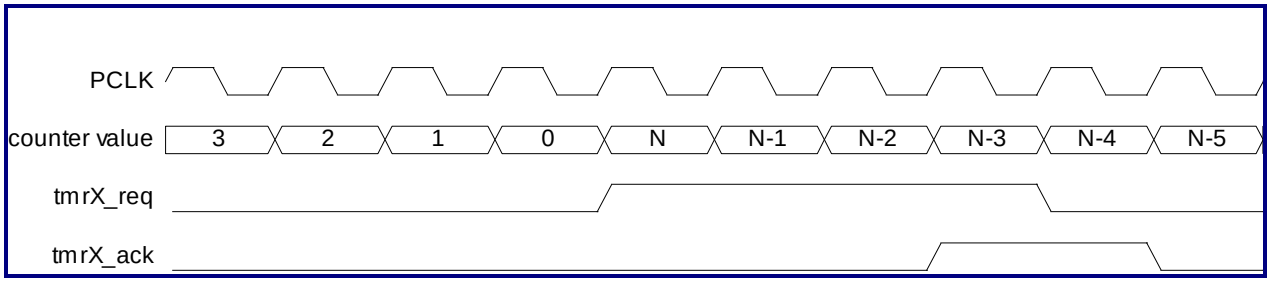
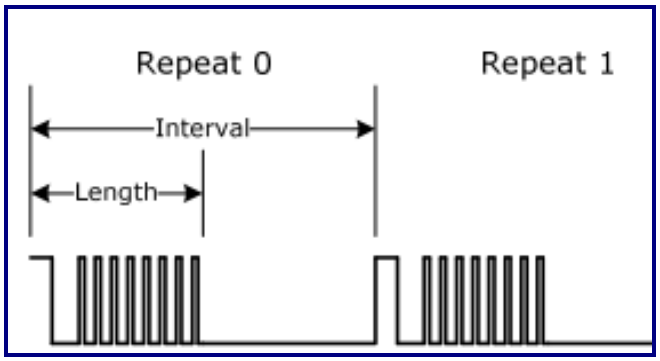
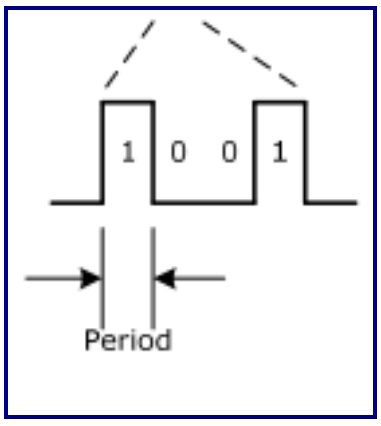


Figure 13-6. DMA Request and ACK

13.5.4.5 PWM Repeat Mode and Pattern Mode

The PWM repeat mode can repeat the one-shot mode between 1 time and 32768 times. The PWM pattern mode can send the user required waveform and repeat it many times.





Chapter 14

SAR ADC Controller

This chapter contains the following sections:

14.1 General Description

The ADC controller is designed with three main functions. The first function is the sampling clock generated by the ADC controller according to the user settings. The second function is the ADC valid data kept in a CPU readable register until the occurrence of the next valid data. The third function is the interrupt triggered according to the user-programmed threshold and trigger method.

14.2 Features

- Programmable sampling clock rate and sampling clock width
- Separated registers readable by CPU to store 3-channel ADC valid data
- Separated programmable interrupt thresholds of different applications (Key detection, Dry battery detection, Li-battery detection, and TV composite signal plug-in/plug-out detection)
- Four different ways to trigger interrupt by user selection
- Separated programmable request intervals of different channels when H/W control is enabled
- ADC channel control signals controlled by S/W or H/W

14.3 Programming Model

14.3.1 Summary of ADC Controller Registers

Table 14-1 provides a summary of the ADC controller registers.

Table 14-1. Summary of ADC Controller Registers

Offset	Name	Reset Value
+0x00	ADC_CTRL_Reg	0x0000 0000
+0x04	ADC_INTR_Reg	0x0000 0000
+0x08	ADC_DAT0_Reg	0x0000 0000
+0x0C	ADC_DAT1_Reg	0x0000 0000
+0x10	ADC_DAT2_Reg	0x0000 0000
+0x14	ADC_THD0_Reg	0x0000 0000
+0x18	ADC_THD1_Reg	0x0000 0000
+0x1C	ADC_THD2_Reg	0x0000 0000
+0x20	ADC_REQ0_Reg	0x0000 0000
+0x24	ADC_REQ1_Reg	0x0000 0000
+0x28	ADC_REQ2_Reg	0x0000 0000
+0x2C	ADC_XTP_Reg	0x0000 0000

14.3.2 Register Descriptions

The following subsections describe the ADC controller registers in more details.

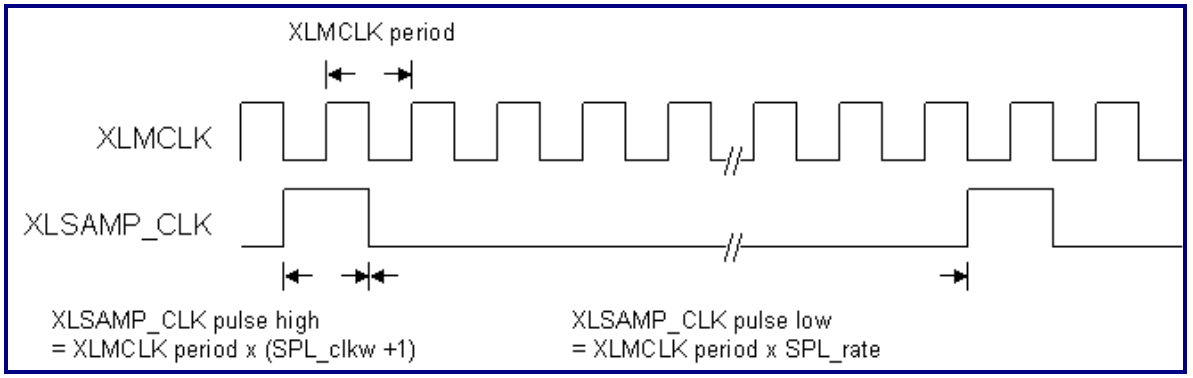
The abbreviations below represent the access types used throughout the register descriptions:

- R/W: Read/Write
- RO: Read Only
- W1C: Write 1 Clear
- HR: Hardware Reset (X_RST_N is asserted.)

14.3.2.1 ADC_CTRL_Reg (Offset = 0x00)

Table 14-2. ADC_CTRL_Reg (Offset = 0x00)

Bit	Name	Type	Description
[31:24]	SPL_rate	R/W	ADC sampling clock rate XLSAMP_CLK pulse low = XLMCLK period * SPL_rate
[23:22]	SPL_clkw	R/W	ADC sampling clock width XLSAMP_CLK pulse high = XLMCLK period * (SPL_clkw + 1)
[21:20]	XLSEL	R/W	SAR ADC XLSEL[1:0] Channel selection if XLSEL_MODE = '1' 0x: Select channel 0 10: Select channel 1 11: Select channel 2
19	XLSEL_MODE	R/W	XLREF_SEL[1:0] and XLSEL are controlled by software.
18	XAIN2_INTEN	R/W	SAR ADC XAIN2 interrupt enable
17	XAIN1_INTEN	R/W	SAR ADC XAIN1 interrupt enable
16	XAIN0_INTEN	R/W	SAR ADC XAIN0 interrupt enable
15	ADC_INTEN	R/W	SAR ADC interrupt enable
14	XAIN2_LTEN	R/W	XAIN2 low-threshold compare enable
13	XAIN2_HTEN	R/W	XAIN2 high-threshold compare enable
12	XAIN1_LTEN	R/W	XAIN1 low-threshold compare enable
11	XAIN1_HTEN	R/W	XAIN1 high-threshold compare enable
10	XAIN0_LTEN	R/W	XAIN0 low-threshold compare enable
9	XAIN0_HTEN	R/W	XAIN0 high-threshold compare enable
8	ADC_PD	R/W	SAR ADC power down
[7:6]	XLREF_SEL	R/W	SAR ADC XLREF_SEL[1:0] Reference control signal if XLSEL_MODE = '1' 00: XAIN range is between GND33A and VCC33A. 01: XAIN range is between 0.1 * VCC33A and 0.9 * VCC33A. 10: XAIN range is between 0.4 * GND33A and 0.8 * VCC33A. 11: XAIN range is between 0.05 * GND33A and 0.25 * VCC33A (When XTP2 = '0').
[5:3]	EOC_DRP_CNT	R/W	EOC drop count
2	XAIN2_EN	R/W	ADC XAIN2 enable
1	XAIN1_EN	R/W	ADC XAIN1 enable
0	XAIN0_EN	R/W	ADC XAIN0 enable



14.3.2.2 ADC_INTR_Reg (Offset = 0x04)

Table 14-3. ADC_INTR_Reg (Offset = 0x04)

Bit	Name	Type	Description
[31:6]	-	-	Reserved
5	XAIN2_LT_INT	W1C	XAIN2 below the low-threshold interrupt flag
4	XAIN2_HT_INT	W1C	XAIN2 above the high-threshold interrupt flag
3	XAIN1_LT_INT	W1C	XAIN1 below the low-threshold interrupt flag
2	XAIN1_HT_INT	W1C	XAIN1 above the high-threshold interrupt flag
1	XAIN0_LT_INT	W1C	XAIN0 below the low-threshold interrupt flag
0	XAIN0_HT_INT	W1C	XAIN0 above the high-threshold interrupt flag

14.3.2.3 ADC_DAT0_Reg (Offset = 0x08)

Table 14-4. ADC_DAT0_Reg (Offset = 0x08)

Bit	Name	Type	Description
[7:0]	ADC_Data0	RO	8bit digital data of XAIN0 (Key detection) sampled by the sampling clock

14.3.2.4 ADC_DAT1_Reg (Offset = 0x0C)

Table 14-5. ADC_DAT1_Reg (Offset = 0x0C)

Bit	Name	Type	Description
[7:0]	ADC_Data1	RO	8bit digital data of XAIN1 (Battery detection) sampled by the sampling clock

14.3.2.5 ADC_DAT2_Reg (Offset = 0x10)

Table 14-6. ADC_DAT2_Reg (Offset = 0x10)

Bit	Name	Type	Description
[7:0]	ADC_Data2	RO	8bit digital data of XAIN2 (Video cable in detection) sampled by the sampling clock

14.3.2.6 ADC_THD0_Reg (Offset = 0x14)

Table 14-7. ADC_THD0_Reg (Offset = 0x14)

Bit	Name	Type	Description
[31:24]	-	-	Reserved
[23:16]	ADC_THD0H	R/W	High threshold of XAIN0
[15:8]	-	-	Reserved
[7:0]	ADC_THD0L	R/W	Low threshold of XAIN0

14.3.2.7 ADC_THD1_Reg (Offset = 0x18)

Table 14-8. ADC_THD1_Reg (Offset = 0x18)

Bit	Name	Type	Description
[31:24]	-	-	Reserved
[23:16]	ADC_THD1H	R/W	High threshold of XAIN1
[15:8]	-	-	Reserved
[7:0]	ADC_THD1L	R/W	Low threshold of XAIN1

14.3.2.8 ADC_THD2_Reg (Offset = 0x1C)

Table 14-9. ADC_THD2_Reg (Offset = 0x1C)

Bit	Name	Type	Description
[31:24]	-	-	Reserved
[23:16]	ADC_THD2H	R/W	High threshold of XAIN2
[15:8]	-	-	Reserved
[7:0]	ADC_THD2L	R/W	Low threshold of XAIN2

14.3.2.9 ADC_REQ0_Reg (Offset = 0x20)

Table 14-10. ADC_REQ0_Reg (Offset = 0x20)

Bit	Name	Type	Description
[31:0]	ADC_REQ0	R/W	XAIN0 request interval

14.3.2.10 ADC_REQ1_Reg (Offset = 0x24)

Table 14-11. ADC_REQ1_Reg (Offset = 0x24)

Bit	Name	Type	Description
[31:0]	ADC_REQ1	R/W	XAIN1 request interval

14.3.2.11 ADC_REQ2_Reg (Offset = 0x28)

Table 14-12. ADC_REQ2_Reg (Offset = 0x28)

Bit	Name	Type	Description
[31:0]	ADC_REQ2	R/W	XAIN2 request interval

14.3.2.12 ADC_XTP_Reg (Offset = 0x2C)

Table 14-13. ADC_XTP_Reg (Offset = 0x2C)

Bit	Name	Type	Description
[31:3]	-	-	Reserved
[2:0]	ADC_XTP	R/W	XTP[2] is used for RD test. XTP[1:0]: Current control for the loading detect circuit The typical mapping values are listed below: 00: 6mA 01: 8mA 10: 10mA 11: 12mA

Chapter 15

DC/AC Characteristics

This chapter contains the following sections:

15.1 Absolute Maximum Ratings

Table 15-1. Absolute Maximum Ratings

Symbol	Parameter	Rating	Unit
VCCK	1.15V core power supply	-0.3 ~ 1.155	V
VCC11A_MPRX	1.15V power supply of the serial interface	-0.3 ~ 1.155	V
VCCDDR	1.5V/1.8V power supply of DDRx	-0.3 ~ 1.575/-0.3 ~ 1.89	V
VCC15O_DDRCKA	1.5V/1.8V power supply of DDRx	-0.3 ~ 1.575/-0.3 ~ 1.89	V
VCC15O_DDRCKD	1.5V/1.8V power supply of DDRx	-0.3 ~ 1.575/-0.3 ~ 1.89	V
VCCIO	3.3V power supply of the I/O pin	-0.3 ~ 3.465	V
VCC3IO_BAYER	3.3V/2.5V power supply of Bayer I/O	-0.3 ~ 3.465/-0.3 ~ 2.625	
VCC3IO_CAP0	3.3V/2.5V power supply of CAP0 I/O	-0.3 ~ 3.465/-0.3 ~ 2.625	
VCC33A_OSCL	3.3V power supply of the OSC pin	-0.3 ~ 3.465	V
VCC33A_REG	3.3V power supply of the internal regulator	-0.3 ~ 3.465	V
VCC33_HSRT	3.3V power supply of OTG	-0.3 ~ 3.465	V
VCC33A_SPK_ADDA	3.3V power supply of the speaker	-0.3 ~ 3.465	V
VCC33A_ADDA	3.3V power supply of the audio codec	-0.3 ~ 3.465	V
VCC33A_DAC	3.3V power supply of TVE	-0.3 ~ 3.465	V
V_{IN3}	Input voltage of 3.3V I/O	-0.3 ~ 3.465	V
V_{IN15}	Input voltage of 1.5V I/O	-0.3 ~ 1.465	V
V_{IN18}	Input voltage of 1.8V I/O	-0.3 ~ 1.89	V
V_{IN5VT}	Input voltage of 3.3V with 5V tolerance I/O Only for X_UART2_SIN and X_UART2_SOUT	-0.3 ~ 5.25	V
T_{STG}	Storage temperature	-40 ~ 150	°C
I_{IN}	DC input current	40	mA
I_{OUT}	Output short circuit current	40	mA

15.2 Recommended Operating Conditions

Table 15-2. Recommended Operating Conditions

Symbol	Parameter	Min.	Typ.	Max.	Unit
VCCK	1.1V core power supply	1.045	1.1	1.155	V
VCC11A_MPRX	1.1V power supply of the serial interface	1.045	1.1	1.155	V
VCCDDR VCC15O_DDRCKA VCC15O_DDRCKD	1.5V/1.8V power supply of DDR	1.425/ 1.71	1.5/1.8	1.575/ 1.89	V
VCCIO	3.3V power supply of the I/O pin	3.135	3.3	3.465	V
VCC3IO_BAYER VCC3IO_CAP0	3.3V/2.5V/1.8V power supply of Bayer or CAP0 I/O pin	3.135/ 2.375/ 1.71	3.3/2.5/ 1.8	3.465/ 2.625/ 1.89	V
VCC33A_OSCL	3.3V power supply of the OSC pin	3.135	3.3	3.465	V
VCC33A_REG	3.3V power supply of the internal regulator	3.135	3.3	3.465	V
VCC33_HSRT	3.3V power supply of OTG	3.135	3.3	3.465	V
VCC33A_SPK_ADDA	3.3V power supply of the speaker	3.135	3.3	3.465	V
VCC33A_ADDA	3.3V power supply of the audio codec	3.135	3.3	3.465	V
VCC33A_DAC	3.3V power supply of TVE	3.135	3.3	3.465	V
T_c	Case temperature	0	-	85	°C

15.3 Leakage Current and Capacitance

Table 15-3. Leakage Current and Capacitance

Symbol	Description	Condition	Typ.	Unit
I_{IN}	Input leakage current for the multi-voltage I/O cells	V_{in} = VCCIO (3.3V), VCC3IO_BAYER/VCC3IO_CAP0 (3.3V, 2.5V, and 1.8V) or 0V	< ±1.0	μA
C_{IN}	Input capacitance	1.8V/2.5V/3.3V I/O cells	2.3	pF

Note: "C_{IN}" includes the cell layout capacitance and the pad capacitance (Estimated to be 0.5 pF).

15.4 DC Characteristics of 3.3V I/O Buffer Cells

Table 15-4. DC Characteristics of 3.3V I/O Buffer Cells (2.0mA ~ 8mA)

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
VCC3I	Power supply for the 3.3V input buffers and the output pre-drivers	3.3V power supply	3.135	3.3	3.465	V
VCC3O	Power supply for the 3.3V output buffers and the input ESD protection	-	3.135	3.3	3.465	V
V_{il}	Input low voltage	LVTTL	-	-	0.8	V
V_{ih}	Input high voltage		2.0	-	-	V
Vt-	Schmitt-trigger negative-to-threshold voltage	LVTTL	0.8	1.1	-	V
V_{t+}	Schmitt-trigger positive-to-threshold voltage		-	1.6	2.0	V
V_{ol}	Output low voltage	I_{ol} = 2.0mA ~ 8.0mA	-	-	0.4	V
V_{oh}	Output high voltage	I_{oh} = 2.0mA ~ 8.0mA	2.4	-	-	V
R_{PU}	Input pull-up resistance	V_{in} = 0V	40	75	190	kΩ
R_{PD}	Input pull-down resistance	V_{in} = VCC3O	30	75	190	kΩ
I_{in}	Input leakage current	V_{in} = VCC3O or 0V	-	±1.0	±10	μA
I_{oz}	Tri-state output leakage current	-	-	±1.0	±10	μA
T_J	Operating junction temperature	-	-40	25	125	°C
V_{det}	Core detection voltage for power-on control	VCC3O = 3.3V	0.24	-	0.84	V

15.5 AC Timing Characteristics

15.5.1 I²C Timing

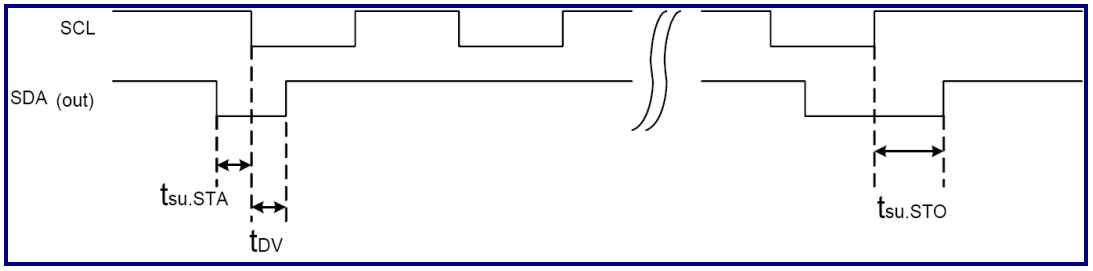


Figure 15-1. I²C Timing Diagram

Parameter	Description	Min.	Max.	Unit
$t_{su.STA}$	START setup time	Note 1	Note 1	ns
t_{DV}	SDA valid time	Note 2	Note 2	ns
$t_{su.STO}$	STOP setup time	Note 3	Note 3	ns

Notes:

1. START setup time = tpclk x count[9:0], where tpclk is the clock cycle time of PCLK and count[9:0] is the CDR register.
2. SDA valid time = tpclk x TSR[9:0], where tpclk is the clock cycle time of PCLK and TSR[9:0] is the TGSR register.
3. STOP setup time = tpclk x count[9:0], where tpclk is the clock cycle time of PCLK and count[9:0] is the CDR register.

15.5.2 RMII MAC Timing

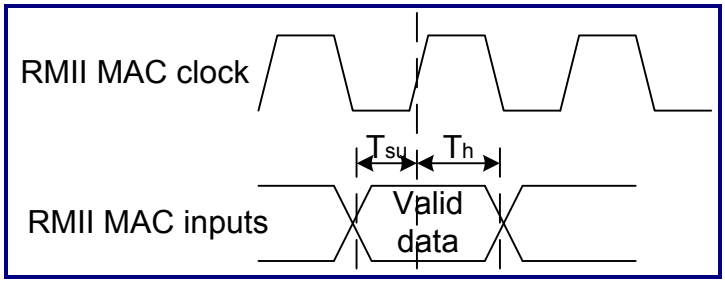


Figure 15-2. RMII MAC Input Timing

Parameter	Description	Min.	Max.	Unit
T_{su}	Setup time of the input signals	2.0	-	ns
T_h	Hold time of the input signals	0	-	ns

Please note that the input signals include X_RMII_RX_CRS_DV, X_RMII_RXD[1:0], X_RMII_RX_ER, X_RMII_MDIO, and X_RMII_PHYLINK.

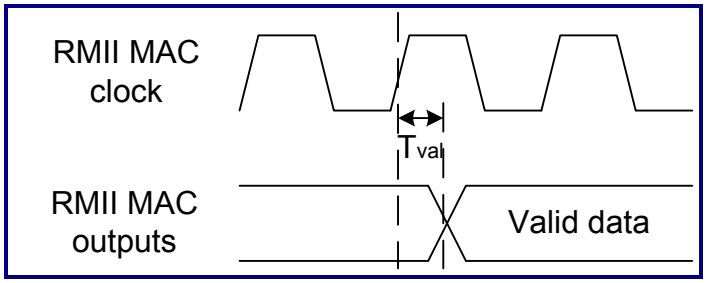


Figure 15-3. RMII MAC Output Timing

Parameter	Description	Min.	Max.	Unit
T_{val}	CLK to data valid delay	2.0	7.6	ns

Please note that the output signals include X_RMII_TXD[1:0], X_RMII_TX_EN, X_RMII_MDC, and X_RMII_MDIO.

15.5.3 SPI Timing

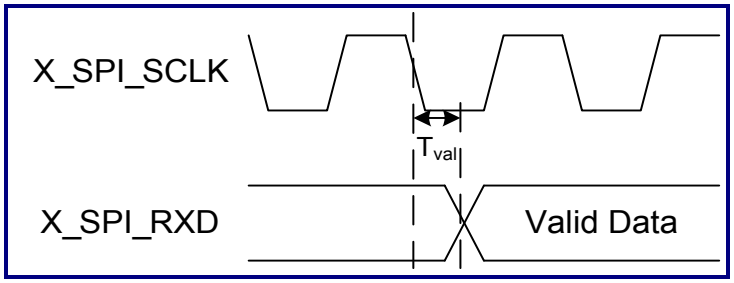


Figure 15-4. SPI Input Timing

Parameter	Description	Min.	Max.	Unit
T_{val}	CLK to data valid delay	2		ns

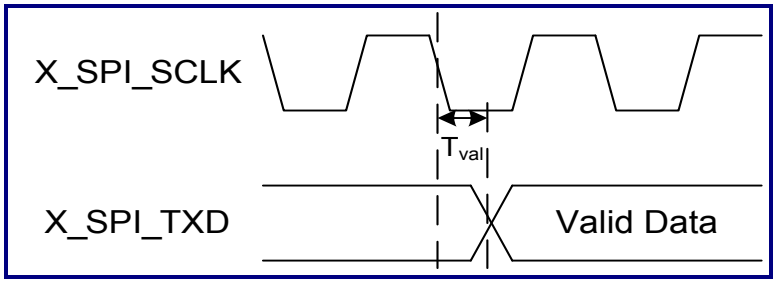


Figure 15-5. SPI Output Timing

Parameter	Description	Min.	Max.	Unit
T_{val}	CLK to data valid delay		8.4	ns

15.5.4 I²S Timing

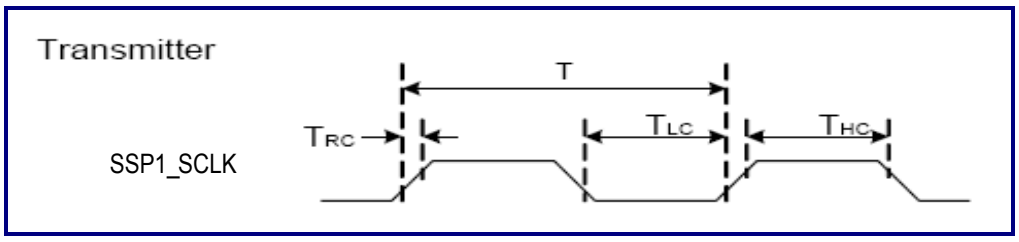


Figure 15-6. I²S Clock Timing

Parameter	Description	Min.	Typ.	Max.	Unit
T	Clock period	-	Note 1	-	ns
T_{HC}	Clock high	-	Note 2	-	ns
T_{LC}	Clock low	-	Note 2	-	ns
T_{RC}	Clock rise time	-	-	-	ns

Notes:

1. T = (sspxclk period) x 2 (SCLKDIV + 1)
2. T_{HC} or T_{LC} = (sspxclk period) x (SCLKDIV + 1)

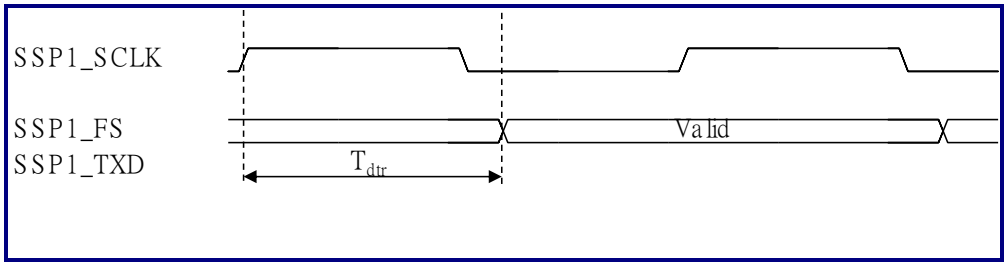


Figure 15-7. I²S Transmitter Timing

Parameter	Description	Min.	Max.	Unit
T_{dtr}	Signal delay time	-	T/2 + 0.6	ns

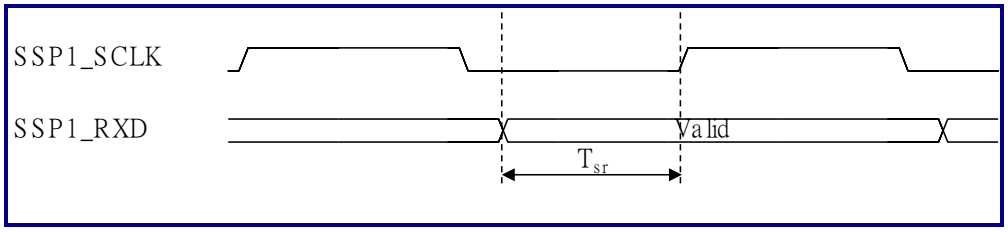


Figure 15-8. I²S Receiver Timing

Parameter	Description	Min.	Max.	Unit
T_{sr}	Signal delay time	3.5	-	ns

15.5.5 BT.656/BT.1120 Input Timing

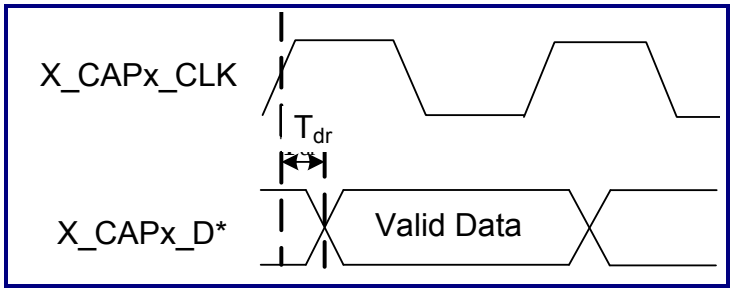


Figure 15-9. BT.656/BT.1120 Input Timing

Please note that the input signals include X_MPRX_CKN, X_MPRX_CKP, X_MPRX_DN0, X_MPRX_DP0, X_MPRX_DN1, X_MPRX_DP1, X_CAP0_D[7:0], and X_BAYER_D7 ~ 4.

Parameter	Description	Min.	Max.	Unit	Note
T_{dr}	Delay time of the input signals	0.5	-	ns	Note 1
T_{dr}	Delay time of the input signals	-	1/2T + 1.0	ns	Note 2
T_{dr}	Delay time of the input signals	0.5	-	ns	Note 3
		-	1.0		

Notes:

1. Latch data at the clock rising edge
2. Latch data at the clock falling edge
3. Latch data at the clock dual edges

15.5.6 Parallel Bayer Input Timing

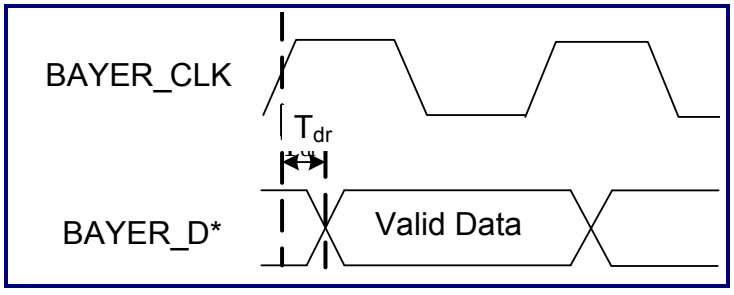


Figure 15-10. Parallel Bayer Input Timing

Please note that the input signals include X_MPRX_CKN, X_MPRX_CKP, X_MPRX_DN0, X_MPRX_DP0, X_MPRX_DN1, X_MPRX_DP1, X_CAP0_D[7:4], and X_BAYER_D7 ~ 4.

Parameter	Description	Min.	Max.	Unit	Note
T_{dr}	Delay time of the input signals	0.5	-	ns	Note 1
T_{dr}	Delay time of the input signals	-	1/2T + 1	ns	Note 2

Notes:

1. Latch data at the clock rising edge
2. Latch data at the clock falling edge

15.5.7 BT.1120/RGB565/BT.656 Output Timing

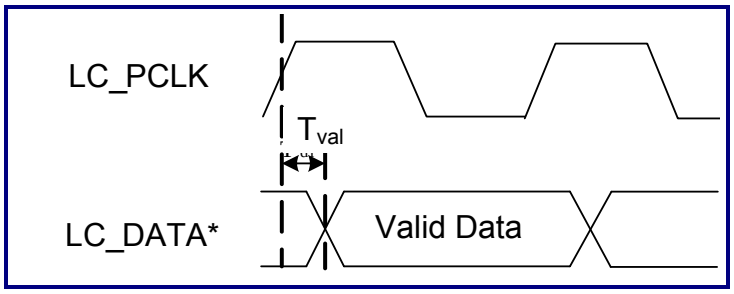


Figure 15-11. BT.1120/RGB565/BT.656 Output Timing

Parameter	Description	Min.	Max.	Unit	Note
T_{val}	CLK to data valid delay	-0.3	-1.0	ns	Note 1
T_{val}	CLK to data valid delay	1/2T - 0.3	1/2T - 1	ns	Note 2

Notes:

1. Data valid at the clock rising edge (LC_PCLK output is at the same phase with the internal pixel clock.)
2. Data valid at the clock falling edge (LC_PCLK output inverts to the internal pixel clock.)

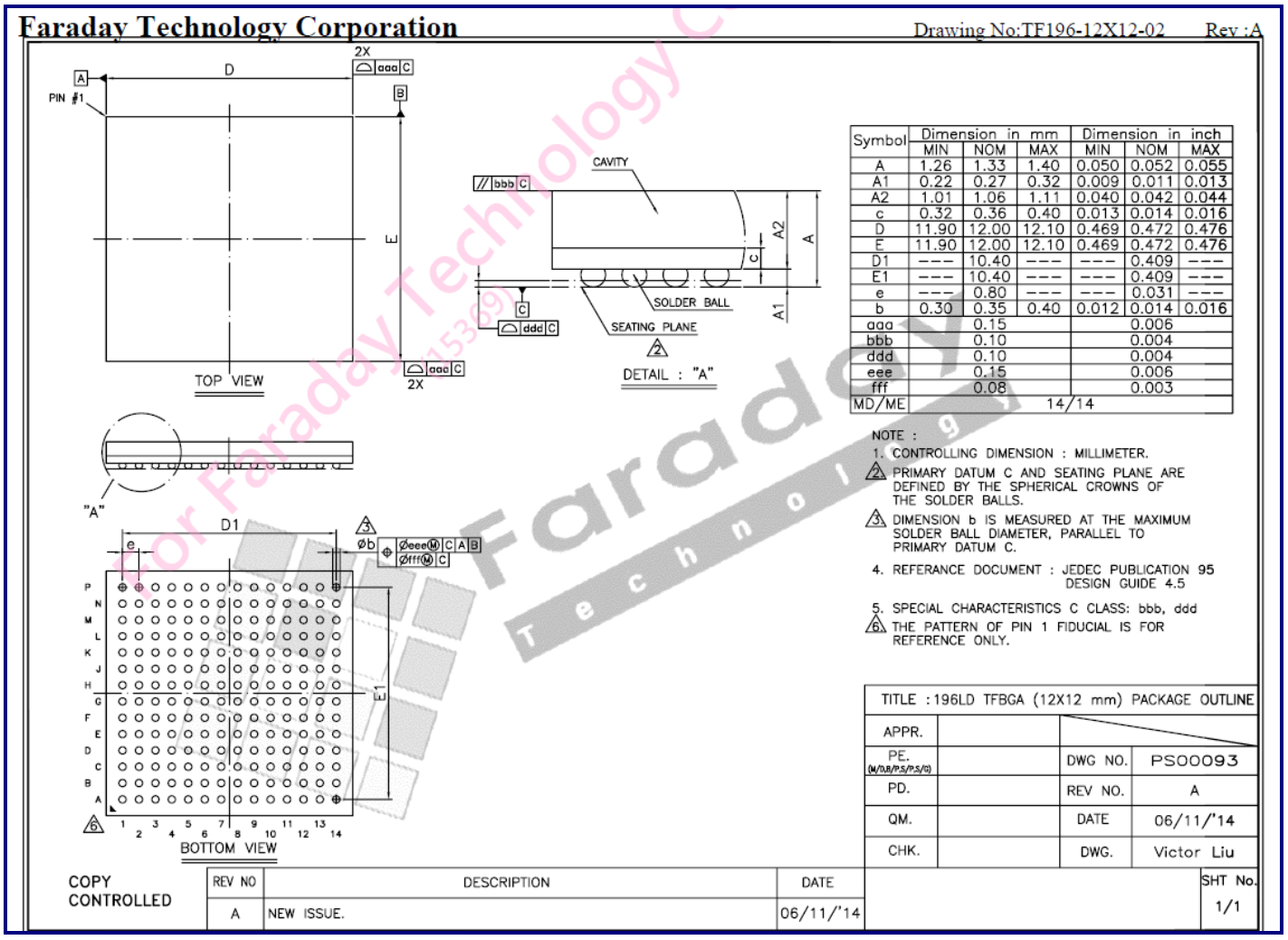
Chapter 16

Package Information

This chapter contains the following section:

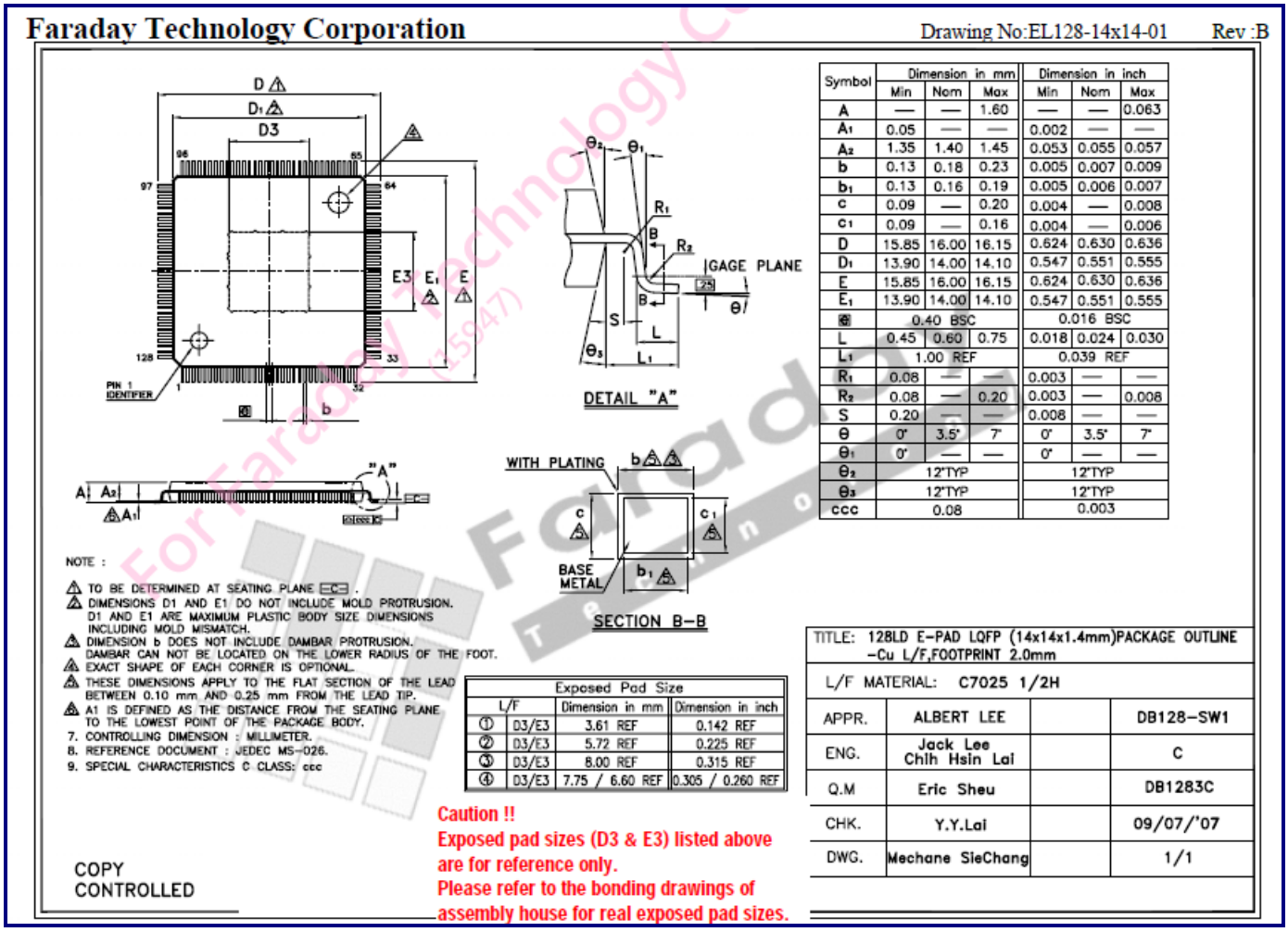
16.1 Package Dimensions

196 TFBGA (12x12)



Symbol	Dimension in mm			Dimension in inch		
	MIN	NOM	MAX	MIN	NOM	MAX
A	1.26	1.33	1.40	0.050	0.052	0.055
A1	0.22	0.27	0.32	0.009	0.011	0.013
A2	1.01	1.06	1.11	0.040	0.042	0.044
c	0.32	0.36	0.40	0.013	0.014	0.016
D	11.90	12.00	12.10	0.469	0.472	0.476
E	11.90	12.00	12.10	0.469	0.472	0.476
D1	---	10.40	---	---	0.409	---
E1	---	10.40	---	---	0.409	---
e	---	0.80	---	---	0.031	---
b	0.30	0.35	0.40	0.012	0.014	0.016
aaa		0.15			0.006	
bbb		0.10			0.004	
ddd		0.10			0.004	
eee		0.15			0.006	
fff		0.08			0.003	
MD/ME		14/14				

Figure 16-1. Package Outline of 196 TFBGA



Symbol	Dimension in mm			Dimension in inch		
	Min	Nom	Max	Min	Nom	Max
A	—	—	1.60	—	—	0.063
A_1	0.05	—	—	0.002	—	—
A_2	1.35	1.40	1.45	0.053	0.055	0.057
b	0.13	0.18	0.23	0.005	0.007	0.009
b_1	0.13	0.16	0.19	0.005	0.006	0.007
c	0.09	—	0.20	0.004	—	0.008
c_1	0.09	—	0.16	0.004	—	0.006
D	15.85	16.00	16.15	0.624	0.630	0.636
D_1	13.90	14.00	14.10	0.547	0.551	0.555
E	15.85	16.00	16.15	0.624	0.630	0.636
E_1	13.90	14.00	14.10	0.547	0.551	0.555
e	0.40 BSC			0.016 BSC		
L	0.45	0.60	0.75	0.018	0.024	0.030
L_1	1.00 REF			0.039 REF		
R_1	0.08	—	—	0.003	—	—
R_2	0.08	—	0.20	0.003	—	0.008
S	0.20	—	—	0.008	—	—
θ	0°	3.5°	7°	0°	3.5°	7°
θ_1	0°	—	—	0°	—	—
θ_2	12°TYP			12°TYP		
θ_3	12°TYP			12°TYP		
ccc	0.08			0.003		

Figure 16-2. Package Outline of 128 LQFP