# Allwinner A83T User Manual

**Revision V1.5.1**

**2015/05/11**

# Declaration

This documentation is the original work and copyrighted property of Allwinner Technology ("Allwinner"). Reproduction in whole or in part must obtain the written approval of Allwinner and give clear acknowledgement to the copyright owner.

The information furnished by Allwinner is believed to be accurate and reliable. Allwinner reserves the right to make changes in circuit design and/or specifications at any time without notice. Allwinner does not assume any responsibility and liability for its use. Nor for any infringements of patents or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Allwinner. This documentation neither states nor implies warranty of any kind, including fitness for any particular application.

Third party licences may be required to implement the solution/product. Customers shall be solely responsible to obtain all appropriately required third party licences. Allwinner shall not be liable for any licence fee or royalty due in respect of any required third party licence. Allwinner shall have no warranty, indemnity or other obligations with respect to matters covered under any required third party licence.

# Revision History

| Version | Date | Description |
| --- | --- | --- |
| V1.0 | 2014/08/25 | Initial release version |
| V1.1 | 2014/09/24 | Correct Port Controller Description |
| V1.2 | 2014/10/23 | Add SD/MMC,NAND,EMAC Register Description |
| V1.3 | 2014/12/30 | Add R_timer,R_watchdog Description |
| V1.4 | 2015/03/16 | Add R_PRCM Module Description |
| V1.5 | 2015/04/23 | Add the programming guide of the Security System |
| V1.5.1 | 2015/05/11 | Modify Security System Register |

# Table of Contents

# Chapter 1 About This Documentation

## 1.1. Documentation Overview

This documentation provides an overall description of the Allwinner Octa-core A83T application processor, which will provide instructions to programmers from several sections, including system, memory, graphic, image, display and interface.

## 1.2. Acronyms and abbreviations

The table below contains acronyms and abbreviations used in this document.

| A | | |
|---|---|---|
| AES | Advanced Encryption Standard | A specification for the encryption of electronic data established by the U.S. National Institute of Standards and Technology (NIST) in 2001 |
| AGC | Automatic Gain Control | An adaptive system found in electronic devices that automatically controls the gain of a signal: the average output signal level is fed back to adjust the gain to an appropriate level for a range of input signal levels |
| AHB | AMBA High-speed Bus | A bus protocol introduced in Advanced Microcontroller Bus Architecture version 2 published by ARM Ltd company |
| APB | Advanced Peripheral Bus | APB is designed for low bandwidth control accesses, which has an address and data phase similar to AHB, but a much reduced, low complexity signal list (for example no bursts) |
| AVS | Audio Video Standard | A compression standard for digital audio and video |
| C | | |
| CIR | Consumer IR | The CIR (Consumer IR) interface is used for remote control through infra-red light |

| CRC | Cyclic Redundancy Check | A type of hash function used to produce a checksum in order to detect errors in data storage or transmission |
|---|---|---|
| CSI | CMOS Sensor Interface | The hardware block that interfaces with different image sensor interfaces and provides a standard output that can be used for subsequent image processing |
| D | | |
| DES | Data Encryption Standard | A previously predominant algorithm for the encryption of electronic data |
| DLL | Delay-Locked Loop | A digital circuit similar to a phase-locked loop (PLL), with the main difference being the absence of an internal voltage-controlled oscillator, replaced by a delay line |
| DRC | Dynamic Range Compression | It reduces the volume of loud sounds or amplifies quiet sounds by narrowing or "compressing" an audio signal's dynamic range. |
| DVFS | Dynamic Voltage and Frequency Scaling | Dynamic voltage scaling is a power management technique where the voltage used in a component is increased or decreased, depending on circumstances. Dynamic frequency scaling is a technique whereby the frequency of a microprocessor can be automatically adjusted on the fly so that the power consumption or heat generated by the chip can be reduced. These two are often used together to save power in mobile devices. |
| E | | |
| EHCI | Enhanced Host Controller Interface | The register-level interface for a Host Controller for the USB Revision 2.0. |
| eMMC | Embedded Multi-Media Card | An architecture consisting of an embedded storage solution with MMC interface, flash memory and controller, all in a small BGA package. |
| F | | |
| FBGA | Fine Ball Grid Array | FBGA is based on BGA technology, but comes with thinner contacts and is mainly used in SoC design |
| G | | |

| | | |
|---|---|---|
| GIC | Generic Interrupt Controller | A centralized resource for supporting and managing interrupts in a system that includes at least one processor |
| H | | |
| HDMI | High-Definition Multimedia Interface | A compact audio/video interface for transmitting uncompressed digital data |
| I | | |
| I2S | Inter-IC Sound | An electrical serial bus interface standard used for connecting digital audio devices together |
| L | | |
| LSB | Least Significant Bit | The bit position in a binary integer giving the units value, that is, determining whether the number is even or odd. It is sometimes referred to as the right-most bit, due to the convention in positional notation of writing less significant digits further to the right. |
| LRADC | Analog to Digital Converter | Used for KEY Application |
| M | | |
| MAC | Media Access Control | A sublayer of the data link layer, which provides addressing and channel access control mechanisms that make it possible for several terminals or network nodes to communicate within a multiple access network that incorporates a shared medium, e.g. Ethernet. |
| MII | Media Independent Interface | An interface originally designed to connect a fast Ethernet MAC-block to a PHY chip, which now has been extended to support reduced signals and increased speeds. |
| MIPI | Mobile Industry Processor Interface | MIPI alliance is an open membership organization that includes leading companies in the mobile industry that share the objective of defining and promoting open specifications for interfaces inside mobile terminals. |
| MIPI DSI | MIPI Display Serial Interface | A specification by the Mobile Industry Processor Interface (MIPI) Alliance aimed at reducing the cost of display sub-systems in a mobile device |

| MSB | Most Significant Bit | The bit position in a binary number having the greatest value, which is sometimes referred to as the left-most bit due to the convention in positional notation of writing more significant digits further to the left |
|---|---|---|
| N | | |
| NTSC | National Television System Committee | An analog television system that is used in most of North America, and many other countries |
| O | | |
| OHCI | Open Host Controller Interface | A register-level interface that enables a host controller for USB to communicate with a host controller driver in software |
| P | | |
| PCM | Pulse Code Modulation | A method used to digitally represent sampled analog signals |
| S | | |
| SPI | Synchronous Peripheral Interface | A synchronous serial data link standard that operates in full duplex mode. Devices communicate in master/slave mode where the master device initiates the data frame. |
| U | | |
| USB DRD | Universal Serial Bus Dual Role Device | A Dual Role Device controller, which supports both USB Host and USB Device functions . |

# Chapter 2  Overview

The Allwinner's A83T is a remarkably lower power,high performance octa-core mobile application processor based on ARM Cortex$^{TM}$-A7 CPU along with SGX544MP1 GPU architecture. It is also highly competitive in term of system cost thanks to its high system integration and is capable of delivering excellent user experience while maintaining low power consumption.

Main feature of A83T include:

- **CPU architecture**:Based on an octa-core Cortex$^{TM}$-A7 CPU architecture,the most power efficient CPU core ARM's ever developed.
- **Graphics**: A83T adopts the extensively implemented and technically mature PowerVR SGX544MP1 to provide mobile users with superior experience in web browsing, video playback and games; OpenGL ES1.1/2.0 ,OpenCL1.1 and DirectX 9.3 standards are supported.
- **Video**: Multi-format playback of up to 1080P high-definition video, and support MPEG1/2, MPEG4 SP/ASP GMC, H.263 including Sorenson Spark, H.264 BP/MP/HP, VP8, WMV9/VC1, JPEG/MJPEG standards with dedicated hardware, and HEVC/H.265 decoder 1080p@30fps with software.
- **Display**: Support RGB/LVDS/DSI/HDMI/CVBS interface to 1920x1200 resolution.Four-lane MIPI DSI is integrated as well.
- **Memory**: Support LPDDR2/LPDDR3/DDR3/DDR3L SDRAM,NAND Flash,SD/eMMC,Nor Flash.
- **HawKView$^{TM}$ ISP**: Support camera up to 8MPixels@30fps,better spatial de-noise and chrominance de-noise, Zone-based AE/AF/AWB statistics,Programmable color correction,Anti-flick detection statistics.

To reduce total system cost and enhance overall functionality, in addition to these major elements,A83T has a broad range of hardware peripherals such as MIPI CSI,LCD controller,Power management,DMA,Timers,High Speed Timer,Security System,GPIO,Digital Audio,UART,SPI,CIR,USB2.0,TWI etc.

A83T application usage is extremely diverse on tablets: users listen to music,watch movies,browse the web,share photos,play games,send emails,and more. When combined with Allwinner PMIC chip that supports CoolFlex technology, A83T enables devices to run different applications more efficiently on different CPU cores ,which build a range of low-power devices and a better user experience.

## 2.1.  Processor Features

### 2.1.1.  CPU Architecture

The A83T platform is based on octa-core Cortex<sup>TM</sup>-A7 CPU architecture.

- ARMv7 ISA standard instruction set plus Thumb-2 and Jazeller RCT
- NEON with SIMD and VFPv4 support
- Support LPAE
- 32KB I-cache and 32KB D-cache per CPU
- 1MB L2-cache(512KB per Cluster)

### 2.1.2.  GPU Architecture

- PowerVR SGX544MP1
- Support OpenGL ES 1.1/2.0
- Support OpenCL 1.1
- Support DirectX 9.3 standards
- Support RenderScript

### 2.1.3.  Memory Subsystem

#### 2.1.3.1.  Boot ROM

- On-chip ROM boot loader
- Support secure and non-secure boot
- secure boot ROM : 64KB,non-secure boot ROM : 32KB
- Support system boot from NAND Flash, SD/TF card, eMMC, SPI NOR Flash.
- Support system code download through USB DRD(Dual Role Device)

#### 2.1.3.2.  SDRAM

- Compatible with the JEDEC standard LPDDR2/LPDDR3/DDR3/DDR3L SDRAM
- Up to 2GB address space

- 32-bit bus width
- Clock frequency up to 800MHz(DDR3)/667MHz(LPDDR3)/533MHz(LPDDR2)
- Support Memory Dynamic Frequency Scale
- 16 address signal lines and 3 bank signal lines
- Automatically generates initialization and refresh sequences
- Runtime-configurable parameters setting for application flexibility
- Random read or write operation is supported

### 2.1.3.3. NAND Flash

- 8-bit data BUS width
- Up to 64-bits ECC per 1024 bytes
- Up to 4 flash chips
- Support 1024, 2048, 4096, 8192, 16K bytes size per page
- Support SDR, ONFI DDR and Toggle DDR NAND

### 2.1.3.4. SD/MMC

- Up to three SD/MMC controllers
- Compatible with eMMC standard specification V4.41, SD physical layer specification V2.0, SDIO card specification V2.0
- 1-bit ,4-bit or 8-bit data bus transfer mode
- Up to 50MHz in both SDR and DDR modes
- Support SDIO suspend and resume operation
- Support hardware CRC generation and error detection
- Support SDIO interrupt detection
- Support block size of 1 to 65535 bytes

## 2.1.4. System Peripheral

### 2.1.4.1. Timer

- Up to two timers
- Two 33-bit AVS(audio and video synchronize) counters
- One watchdog to generate reset signal or interrupts

---

### 2.1.4.2. High Speed Timer

- Counters up to 56 bits
- Clock source is synchronized with AHB clock, much more accurate than other timers

### 2.1.4.3. OSC24M

- Support 1.8v oscillator
- Support internal RC oscillator

### 2.1.4.4. GIC

- 16 SGIs(Software Generated Interrupt)
- 16 PPIs(Private Peripheral Interrupt)
- 125 SPIs(Shared Peripheral Interrupts)

### 2.1.4.5. DMA

- 8-channel DMA
- Support data width of 8/16/32 bits
- Support linear and IO address modes
- Support data transfer types with memory-to-memory,memory-to-peripheral,peripheral-to-memory

### 2.1.4.6. CCU

- 11 PLLs
- Support a 24MHz oscillator and an on-chip RC oscillator
- Support clock configuration for corresponding modules
- Support software-controlled clock gating and software-controlled reset for corresponding modules

### 2.1.4.7. PWM

- Support outputting two kinds of waveform:continuous waveform and pulse waveform
- 0% to 100% adjustable duty cycle
- Up to 24MHz output frequency

#### 2.1.4.8.  Security System

- Support symmetrical algorithm: AES,DES,TDES
- Support hash algorithm: MD5,SHA1,SHA224,SHA256
- Support asymmetrical algorithm: RSA 512/1024/2048/3072-bits
- Support 160-bit hardware PRNG with 175-bit seed
- Support 256bits TRNG
- Support ECB,CBC,CTR,CTS modes for AES
- 128-bit, 192-bit and 256-bit key size for AES
- Support ECB,CBC,CTR modes for DES/TDES
- Support 2Kbit EFUSE for chip ID and security application

#### 2.1.4.9.  TrustZone

- Support TrustZone technology
- Support 96KB secure SRAM

#### 2.1.4.10.  CPU Configuration

- Support power clamp
- Support flexible CPU configuration

#### 2.1.4.11.  Power Management

- Support DVFS for CPU frequency and voltage adjustment
- Support flexible clock gate and module reset
- Support dynamic frequency adjustment for external DRAM
- Support multiple power domains

## 2.1.5.  Display Subsystem

#### 2.1.5.1.  Display Engine2.0

- Support input layer size up to 2048x2048, and output size up to 2048x2048.
- Support four alpha blending channel for main display,two channel for aux display.
- Support four overlay layer in each channel, and has a independent scaler.
- Support potter-duff compatible blending operation.

- Support input format YUV422/YUV420/YUV411/ARGB8888/XRGB8888/RGB888/ARGB4444/ARGB1555 and RGB565
- Support Frame Packing/Top-and-Bottom/Side-by-Side Full/Side-by-Side Half 3D format data
- Support display enhancement 2.0 for excellent display experience
    - Adaptive edge sharping
    - Adaptive color enhancement
    - Adaptive contrast enhancement and fresh tone rectify
- Support write back & rotation for high efficient dual display and miracast

### 2.1.5.2. Video Output

- Support two independent display channels
- Support RGB up to 1920x1200@60Hz resolution
- Support RGB666/656 dither function
- Support LVDS up to 1366x768@60Hz resolution
- Support 4-lane MIPI DSI (V1.0) up to 1920x1200@60Hz resolution
- Support HDMI V1.4 output with HDCP1.2

## 2.1.6. Video Engine

### 2.1.6.1. Video Decoding

- Support video playback up to 1080p@60fps
- Support multi-format video playback, including MPEG1/2, MPEG4 SP/ASP GMC, H.263 including Sorenson Spark, H.264 BP/ MP/HP ,VP8, WMV9/VC-1, JPEG/MJPEG, etc
- HEVC/H.265 decoder(software),Main Profile,1080p@30fps

### 2.1.6.2. Video Encoding

- Support H.264 video encoding up to 1080p@60fps, 720p@120fps
- JPEG baseline: picture size up to 8192x8192
- Support input picture size up to 4800x4800
- Support input format: tiled (128x32)/YU12/YV12/NU12/NV12/ARGB/YUYV
- Support Alpha blending
- Support thumb generation
- Support 4x2 scaling ratio: from 1/16 to 64 arbitrary non-integer ratio
- Support rotated input

## 2.1.7. Image Subsystem

### 2.1.7.1. CSI

- Up to 5M pixel camera sensor
- Video shot up to 720p@30fps
- Support 10-bit parallel camera sensor
- Support CCIR656 protocol for NTSC and PAL

### 2.1.7.2. MIPI CSI

- 4-lane MIPI CSI ,Compliant with MIPI-CSI2 V1.00 and MIPI DPHY V1.00.00
- Up to 1Gbps per Lane in HS Transmission
- Support video shot up to 1080p@60fps
- Maximum to 8M@30fps with 4 data lane
- Supports format: YUV422-8bit/10bit,YUV420-8bit/10bit,RAW-8,RAW-10,RAW-12,RGB888,RGB565

### 2.1.7.3. ISP

- Support input formats: 8/10-bit RAW RGB, 8-bit YCbCr
- Support output formats: YCbCr420 semi-planar,YCrCb420 semi-planar, YCbCr422 semi-planar,YCrCb422 semi-planar,YUV420 planar, YUV422 planar
- Support image mirror flip and rotation
- Support two output channels
- Support speed up to 8MPixels@30fps
- Defect pixel correction
- Super lens shading correction
- Anisotropic non-linear Bayer interpolation with false color suppression
- Programmable color correction
- Advanced contrast enhance and sharping
- Advanced saturation adjust
- Advanced spatial (2D) de-noise filter
- Advanced chrominance noise reduction
- DRC(dynamic range compression)
- Zone-based AE/AF/AWB statistics
- Anti-flick detection statistics
- Histogram statistics

## 2.1.8. External Peripherals

### 2.1.8.1. USB

- USB 2.0 DRD
    - Complies with USB2.0 Specification
    - Support High-Speed(HS,480Mbps),Full-Speed(FS,12Mbps),and Low-Speed(LS,1.5Mbps) in host mode
    - Up to 10 User-Configurable Endpoints
    - Support point-to-point and point-to-multipoint transfer in both Host and Peripheral mode
- Two EHCI/OHCI compliant Host SIE multiplexed with one USB 2.0 analog PHY, one HSIC PHY
    - Support High-Speed(HS,480Mbps),Full-Speed(FS,12Mbps),and Low-Speed(LS,1.5Mbps) Device
    - An internal DMA Controller for data transfer with memory

### 2.1.8.2. EMAC

- Support 10/100/1000Mbps data transfer rate
- Support MII/RGMII PHY interface
- Support full-duplex and half-duplex operation
- Programmable frame length
- Flexible address filtering modes
- Automatic CRC and pad generation controllable on a per-frame basis
- Options for Automatic Pad/CRC Stripping on receive frames
- Programmable frame length to support Standard or Jumbo Ethernet frames with sizes up to 16 KB
- Programmable Inter Frame Gap (40-96 bit times in steps of 8)
- Supports a variety of flexible address filtering modes

### 2.1.8.3. ADC

- LRADC with 6-bit resolution
- Support hold key and continuous key
- Support single key, normal key and continuous key

### 2.1.8.4. Digital Audio

- I2S/PCM
    - I2S and PCM are configurable through software
    - Support I2S formats:normal,left-justified,right-justified
    - Audio data resolution:16bits, 20bits, 24bits
    - Audio sample rate up to 192KHz

- Master and slave work mode are configurable
- TDM(Time Division Multiplexing)
  - Master/Slave mode
  - Audio sample resolution from 8bits to 32bits
  - Sample rate from 8KHz to 192KHz
  - 4 data output pin
  - DMA-based or interrupt-based operation

### 2.1.8.5. CIR

- Support a flexible receiver for IR remote
- Programmable FIFO thresholds

### 2.1.8.6. UART

- Up to six UART controllers
- 64-Bytes Transmit and receive data FIFOs for all UART
- Compliant with industry-standard 16550 UARTs
- Support Infrared Data Association(IrDA) 1.0 SIR
- Interrupt support for FIFOs,Status Change

### 2.1.8.7. SPI

- Up to two SPI controllers
- Master/Slave configurable
- Polarity and phase of the chip select(SPI_SS) and SPI_Clock(SPI_CLK) are configurable
- Two 64-Bytes FIFO for SPI-TX and SPI-RX operation
- DMA-based or interrupt-based operation

### 2.1.8.8. TWI

- Up to four TWI(Two Wire Interface) controllers
- Support Standard mode(up to 100K bps) and Fast mode(up to 400K bps)
- Master/Slave configurable
- Allows 10-bit addressing transactions

### 2.1.8.9. RSB^{TM} (Reduced Serial Bus)

- A simplified two wire protocol
- Support master mode

- Support multi-slaves
- Speed up to 20Mbps

## 2.1.9.    Package

- FCBGA 345 balls, 0.65mm ball pitch, 14mm x 14mm

## 2.2. System Block Diagram
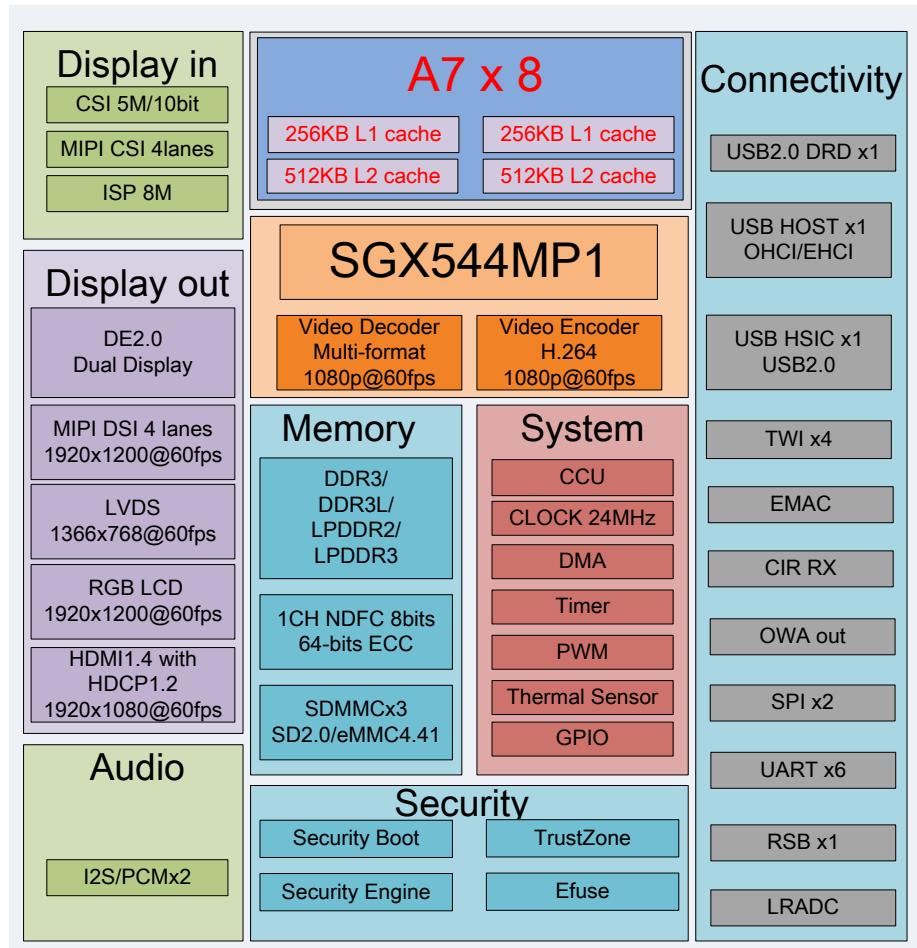


Figure 2-1. System Block Diagram

# Chapter 3 System

The chapter describes the A83T system from following sections:

- Memory Mapping
- Boot System
- CCU
- CPU Configuration
- TimeStamp
- System Control
- Timer
- Trusted Watchdog
- High-speed Timer
- PWM
- DMA
- GIC
- Message Box
- Spinlock
- Security System
- Security ID
- Secure Memory Controller
- Secure Memory Touch Arbiter
- Thermal Sensor Controller
- LRADC
- R_timer
- R_watchdog
- R_PRCM
- Port Controller

## 3.1. Memory Mapping

| Module | Address (It is for Cluster CPU) | Size (byte) |
|---|---|---|
| SRAM A1 | 0x0000 0000---0x0000 7FFF | 32K |
| SRAM A2 | 0x0004 4000---0x0005 3FFF | 64K |
| SRAM B(Secure) | 0x0002 0000---0x0003 7FFF | 96K |
| DE | 0x0100 0000---0x013F FFFF | 4M |
| CS Debug | 0x0140 0000---0x0141 FFFF | 128K |
| TSGEN RO | 0x0150 6000---0x0150 6FFF | 4K |
| TSGEN CTRL | 0x0160 7000---0x0160 7FFF | 4K |
| CPU_CFG | 0x0170 0000---0x0170 03FF | 1K |
| TIMESTAMP_STA | 0x0171 0000---0x0171 0FFF | 4K |
| TIMESTAMP_CTRL | 0x0172 0000---0x0172 0FFF | 4K |
| Cluster 0 MBIST | 0x0150 2000---0x0150 2FFF | 4K |
| Cluster 1 MBIST | 0x0160 2000---0x0160 2FFF | 4K |
| CCI-400 | 0x0179 0000---0x0179 FFFF | 64K |
| SYS_CTRL | 0x01C0 0000---0x01C0 0FFF | 4K |
| DMA | 0x01C0 2000---0x01C0 2FFF | 4K |
| NDFC | 0x01C0 3000---0x01C0 3FFF | 4K |
| Key Memory Space | 0x01C0 B000---0x01C0 BFFF | 4K |
| LCD0 | 0x01C0 C000---0x01C0 CFFF | 4K |
| LCD1 | 0x01C0 D000---0x01C0 DFFF | 4K |
| VE | 0x01C0 E000---0x01C0 EFFF | 4K |
| SD/MMC 0 | 0x01C0 F000---0x01C0 FFFF | 4K |
| SD/MMC 1 | 0x01C1 0000---0x01C1 0FFF | 4K |
| SD/MMC 2 | 0x01C1 1000---0x01C1 1FFF | 4K |
| SID | 0x01C1 4000---0x01C1 43FF | 1K |
| SS | 0x01C1 5000---0x01C1 5FFF | 4K |
| MSG_BOX | 0x01C1 7000---0x01C1 7FFF | 4K |
| SPINLOCK | 0x01C1 8000---0x01C1 8FFF | 4K |
| USB-DRD | 0x01C1 9000---0x01C1 9FFF | 4K |
| USB-EHCI0/OHCI0 | 0x01C1 A000---0x01C1 AFFF | 4K |
| USB-EHCI1 | 0x01C1 B000---0x01C1 BFFF | 4K |
| SMC | 0x01C1 E000---0x01C1 EFFF | 4K |
| CCU | 0x01C2 0000---0x01C2 03FF | 1K |
| PIO | 0x01C2 0800---0x01C2 0BFF | 1K |
| TIMER | 0x01C2 0C00---0x01C2 0FFF | 1K |
| OWA | 0x01C2 1000---0x01C2 13FF | 1K |
| PWM | 0x01C2 1400---0x01C2 17FF | 1K |
| DAUDIO-0 | 0x01C2 2000---0x01C2 23FF | 1K |
| DAUDIO-1 | 0x01C2 2400---0x01C2 27FF | 1K |
| DAUDIO-2 | 0x01C2 2800---0x01C2 2BFF | 1K |

| TDM | 0x01C2 3000---0x01C2 33FF | 1K |
|---|---|---|
| SMTA | 0x01C2 3400---0x01C2 37FF | 1K |
| DSI | 0x01C2 6000---0x01C2 6FFF | 4K |
| UART 0 | 0x01C2 8000---0x01C2 83FF | 1K |
| UART 1 | 0x01C2 8400---0x01C2 87FF | 1K |
| UART 2 | 0x01C2 8800---0x01C2 8BFF | 1K |
| UART 3 | 0x01C2 8C00---0x01C2 8FFF | 1K |
| UART 4 | 0x01C2 9000---0x01C2 93FF | 1K |
| TWI 0 | 0x01C2 AC00---0x01C2 AFFF | 1K |
| TWI 1 | 0x01C2 B000---0x01C2 B3FF | 1K |
| TWI 2 | 0x01C2 B400---0x01C2 B7FF | 1K |
| EMAC | 0x01C3 0000---0x01C3 FFFF | 64K |
| GPU | 0x01C4 0000---0x01C4 FFFF | 64K |
| HSTMR | 0x01C6 0000---0x01C6 0FFF | 4K |
| DRAMCOM | 0x01C6 2000---0x01C6 2FFF | 4K |
| DRAMCTL0 | 0x01C6 3000---0x01C6 3FFF | 4K |
| DRAMPHY0 | 0x01C6 5000---0x01C6 5FFF | 4K |
| SPI0 | 0x01C6 8000---0x01C6 8FFF | 4K |
| SPI1 | 0x01C6 9000---0x01C6 9FFF | 4K |
| GIC | 0x01C8 0000 | |
| CSI | 0x01CB 0000---0x01CE FFFF | 256K |
| VE Memory Space | 0x01D0 0000 | |
| HDMI | 0x01EE 0000---0x01EF FFFF | 128K |
| R_TIMER | 0x01F0 0800---0x01F0 0BFF | 1K |
| R_INTC | 0x01F0 0C00---0x01F0 0FFF | 1K |
| R_WDOG | 0x01F0 1000---0x01F0 13FF | 1K |
| R_PRCM | 0x01F0 1400---0x01F0 17FF | 1K |
| R_TWD | 0x01F0 1800---0x01F0 1BFF | 1K |
| R_CPUS_CFG | 0x01F0 1C00---0x01F0 1FFF | 1K |
| R_CIR-RX | 0x01F0 2000---0x01F0 23FF | 1K |
| R_TWI | 0x01F0 2400---0x01F0 27FF | 1K |
| R_UART | 0x01F0 2800---0x01F0 2BFF | 1K |
| R_PIO | 0x01F0 2C00---0x01F0 2FFF | 1K |
| R_RSB | 0x01F0 3400---0x01F0 37FF | 1K |
| R_PWM | 0x01F0 3800---0x01F0 3BFF | 1K |
| R_LRADC | 0x01F0 3C00---0x01F0 3FFF | 1K |
| R_TH | 0x01F0 4000---0x01F0 43FF | 1K |
| DDR Space | 0x4000 0000---0xBFFF FFFF | 2G |
| S-BROM | 0xFFFF 0000—0xFFFF FFFF | 64K |
| NS-BROM | 0xFFFF 0000—0xFFFF 7FFF | 32K |

## 3.2. Boot System

The Boot System includes the following features:

- The system will boot in different ways based on whether its security features are enabled
- Support CPU-0 boot process and CPU-0+ boot process
- Support CPU hot plug process
- Support super standby wakeup process
- Support mandatory upgrade process through SDC0 and USB DRD
- Support fast boot process from Raw NAND,eMMC,SD/TF card ,and SPI NOR Flash

## 3.3.  CCU

### 3.3.1.  Overview

The CCU controls the PLLs configuration and most of the clock generation, division, distribution, synchronization and gating. CCU input signals include the external clock for the reference frequency (24MHz). The outputs from CCU are mostly clocks to other blocks in the system.

The CCU includes the following features:
- 11 PLLs, independent PLL for CPUX
- Bus Source and Divisions
- PLLs Bias Control
- PLLs Tunning Control
- PLLs Pattern Control
- Configuring Modules Clock
- Bus Clock Gating
- Bus Software Reset

## 3.3.2.  Functionalities Description

### 3.3.2.1.  System Bus
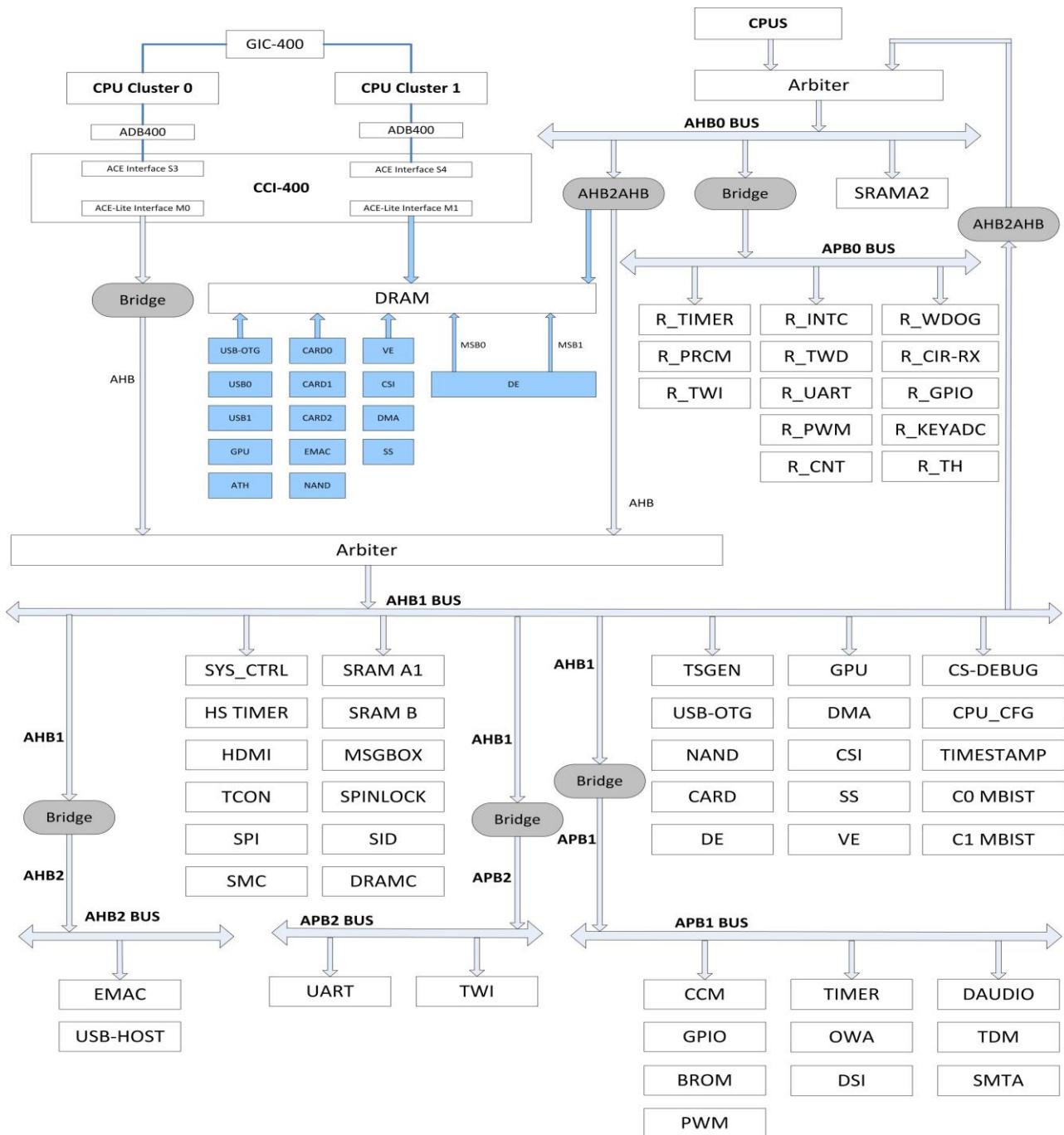


Figure 3-1. System Bus Tree

### 3.3.2.2.  Bus clock tree
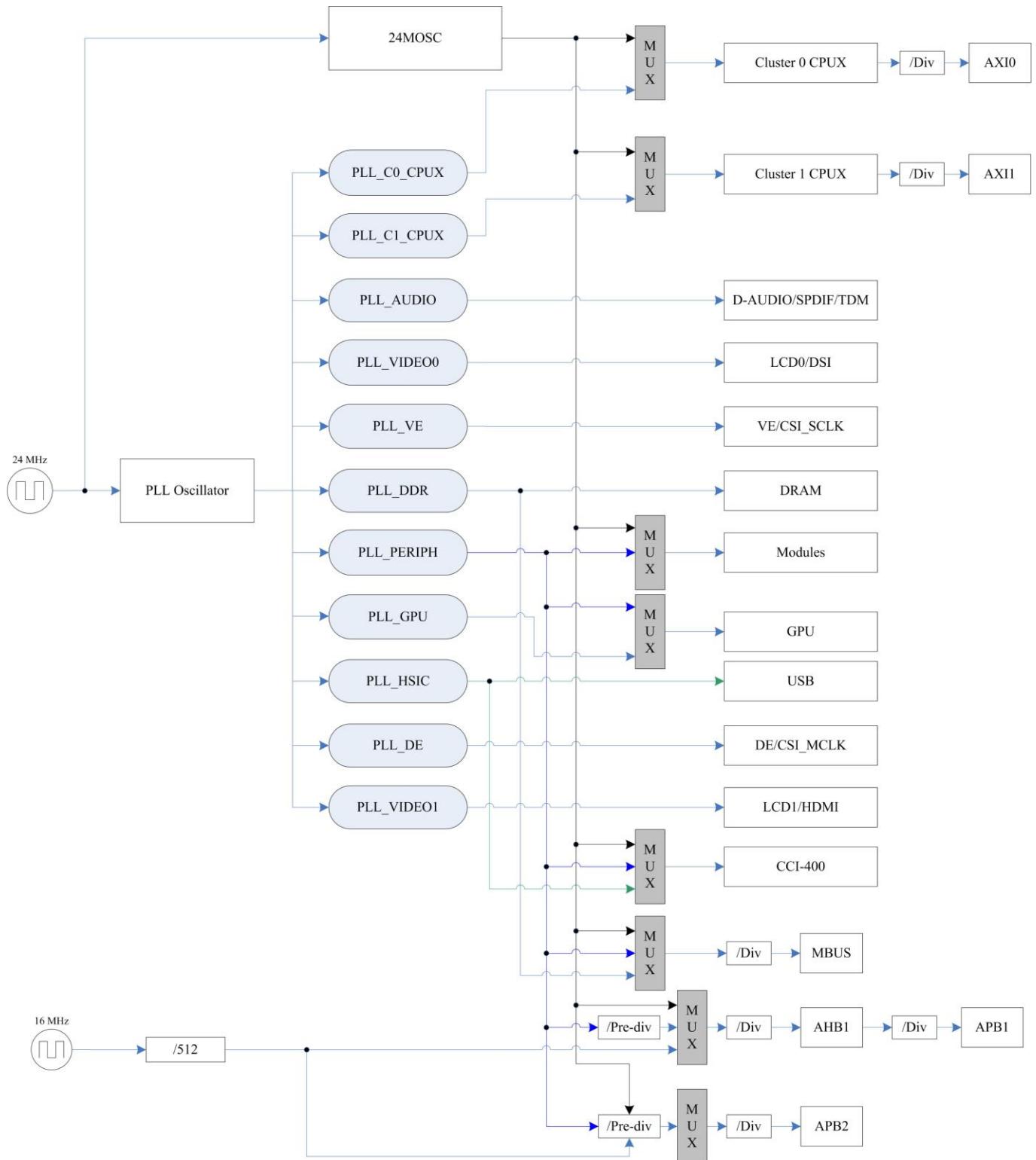


Figure 3-2. Bus Clock Tree

### 3.3.3.  Typical Applications

Clock output of **PLL_C0CPUX** is used only for Cluster 0 CPUX, and the frequency factor can be dynamically modified for DVFS;

Clock output of **PLL_C1CPUX** is used only for Cluster 1 CPUX, and the frequency factor can be dynamically modified for DVFS;

Clock output of **PLL_AUDIO** can be used for DAUDIO, TDM, OWA etc, and dynamic frequency scaling is not supported;

Clock output of **PLL_PERIPH** can be used for MBUS/AHB1/AHB2/APB1/APB2/CCI-400 and NAND/GPU_Memory/MMC/ SS/SPI/CSI,etc, and dynamic frequency scaling is not supported;

Clock output of **PLL_VE** can be used for CSI and VE , and dynamic frequency scaling is not supported;

Clock output of **PLL_DDR** can be used for MBUS and DRAM, and dynamic frequency scaling is supported;

Clock output of **PLL_VIDEO0** can be used for TCON0 and MIPI_DSI, and dynamic frequency scaling is not supported;

Clock output of **PLL_VIDEO1** can be used for TCON1 and HDMI, and dynamic frequency scaling is not supported;

Clock output of **PLL_DE** can be used for CSI, and dynamic frequency scaling is not supported;

Clock output of **PLL_HSIC** can be used for CCI-400 and USBPHY, and dynamic frequency scaling is not supported;

Clock output of **PLL_GPU** can be used for GPU_Core/GPU_Memory, and dynamic frequency scaling is not supported;

### 3.3.4.  Register List

| Module Name | Base Address |
|---|---|
| CCU | 0x01C20000 |

| Register Name | Offset | Description |
|---|---|---|
| PLL_C0CPUX_CTRL_REG | 0x0000 | PLL_C0CPUX Control Register |
| PLL_C1CPUX_CTRL_REG | 0x0004 | PLL_C1CPUX Control Register |
| PLL_AUDIO_CTRL_REG | 0x0008 | PLL_AUDIO Control Register |
| PLL_VIDEO0_CTRL_REG | 0x0010 | PLL_VIDEO0 Control Register |
| PLL_VE_CTRL_REG | 0x0018 | PLL_VE Control Register |
| PLL_DDR_CTRL_REG | 0x0020 | PLL_DDR Control Register |
| PLL_PERIPH_CTRL_REG | 0x0028 | PLL_PERIPH Control Register |
| PLL_GPU_CTRL_REG | 0x0038 | PLL_GPU Control Register |
| PLL_HSIC_CTRL_REG | 0x0044 | PLL_HSIC Control Register |
| PLL_DE_CTRL_REG | 0x0048 | PLL_DE Control Register |
| PLL_VIDEO1_CTRL_REG | 0x004C | PLL_VIDEO1 Control Register |
| CPUX_AXI_CFG_REG | 0x0050 | CPUX/AXI Configuration Register |
| AHB1_APB1_CFG_REG | 0x0054 | AHB1/APB1 Configuration Register |
| APB2 _CFG_REG | 0x0058 | APB2 Configuration Register |
| AHB2_CFG_REG | 0x005C | AHB2 Configuration Register |
| BUS_CLK_GATING_REG0 | 0x0060 | Bus Clock Gating Register 0 |
| BUS_CLK_GATING_REG1 | 0x0064 | Bus Clock Gating Register 1 |

| BUS_CLK_GATING_REG2 | 0x0068 | Bus Clock Gating Register 2 |
|---|---|---|
| BUS_CLK_GATING_REG3 | 0x006C | Bus Clock Gating Register 3 |
| CCI400_CFG_REG | 0x0078 | CCI-400 Configuration Register |
| NAND_CLK_REG | 0x0080 | NAND Clock Register |
| SDMMC0_CLK_REG | 0x0088 | SDMMC0 Clock Register |
| SDMMC1_CLK_REG | 0x008C | SDMMC1 Clock Register |
| SDMMC2_CLK_REG | 0x0090 | SDMMC2 Clock Register |
| SS_CLK_REG | 0x009C | SS Clock Register |
| SPI0_CLK_REG | 0x00A0 | SPI0 Clock Register |
| SPI1_CLK_REG | 0x00A4 | SPI1 Clock Register |
| DAUDIO0_CLK_REG | 0x00B0 | DAUDIO0 Clock Register |
| DAUDIO1_CLK_REG | 0x00B4 | DAUDIO1 Clock Register |
| DAUDIO2_CLK_REG | 0x00B8 | DAUDIO2 Clock Register |
| TDM_CLK_REG | 0x00BC | TDM Clock Register |
| OWA_CLK_REG | 0x00C0 | OWA Clock Register |
| USBPHY_CFG_REG | 0x00CC | USBPHY Configuration Register |
| DRAM_CFG_REG | 0x00F4 | DRAM Configuration Register |
| PLL_DDR_CFG_REG | 0x00F8 | PLL_DDR Configuration Register |
| MBUS_RST_REG | 0x00FC | MBUS Reset Register |
| DRAM_CLK_GATING_REG | 0x0100 | DRAM Clock Gating Register |
| TCON0_CLK_REG | 0x0118 | TCON0 Clock Register |
| TCON1_CLK_REG | 0x011C | TCON1 Clock Register |
| MIPI_CSI_CLK_REG | 0x0130 | MIPI_CSI Clock Register |
| CSI_CLK_REG | 0x0134 | CSI Clock Register |
| VE_CLK_REG | 0x013C | VE Clock Register |
| AVS_CLK_REG | 0x0144 | AVS Clock Register |
| HDMI_CLK_REG | 0x0150 | HDMI Clock Register |
| HDMI_SLOW_CLK_REG | 0x0154 | HDMI Slow Clock Register |
| MBUS_CLK_REG | 0x015C | MBUS Clock Register |
| MIPI_DSI_CLK_REG | 0x0168 | MIPI_DSI Clock Register |
| GPU_CORE_CLK_REG | 0x01A0 | GPU Core Clock Register |
| GPU_MEM_CLK_REG | 0x01A4 | GPU Memory Clock Register |
| GPU_HYD_CLK_REG | 0x01A8 | GPU HYD Clock Register |
| PLL_STABLE_TIME_REG0 | 0x0200 | PLL Stable Time Register 0 |
| PLL_STABLE_TIME_REG1 | 0x0204 | PLL Stable Time Register 1 |
| PLL_STB_STATUS_REG | 0x020C | PLL Stable Status Register |
| PLL_C0CPUX_BIAS_REG | 0x0220 | PLL_C0CPUX Bias Register |
| PLL_AUDIO_BIAS_REG | 0x0224 | PLL_AUDIO Bias Register |
| PLL_VIDEO0_BIAS_REG | 0x0228 | PLL_VIDEO0 Bias Register |
| PLL_VE_BIAS_REG | 0x022C | PLL_VE Bias Register |
| PLL_DDR_BIAS_REG | 0x0230 | PLL_DDR Bias Register |
| PLL_PERIPH_BIAS_REG | 0x0234 | PLL_PERIPH Bias Register |
| PLL_C0CPUX_BIAS_REG | 0x0238 | PLL_C1CPUX Bias Register |
| PLL_GPU_BIAS_REG | 0x023C | PLL_GPU Bias Register |

| PLL_HSIC_BIAS_REG | 0x0244 | PLL_HSIC Bias Register |
|---|---|---|
| PLL_DE_BIAS_REG | 0x0248 | PLL_DE Bias Register |
| PLL_VIDEO1_BIAS_REG | 0x024C | PLL_VIDEO1 Bias Register |
| PLL_C0CPUX_TUN_REG | 0x0250 | PLL_C0CPUX Tuning Register |
| PLL_C1CPUX_TUN_REG | 0x0254 | PLL_C1CPUX Tuning Register |
| PLL_AUDIO_PAT_CTRL_REG0 | 0x0284 | PLL_AUDIO Pattern Control Register 0 |
| PLL_VIDEO0_PAT_CTRL_REG0 | 0x0288 | PLL_VIDEO0 Pattern Control Register 0 |
| PLL_DDR_PAT_CTRL_REG0 | 0x0290 | PLL_DDR Pattern Control Register 0 |
| PLL_AUDIO_PAT_CTRL_REG1 | 0x02A4 | PLL_AUDIO Pattern Control Register 1 |
| PLL_VIDEO0_PAT_CTRL_REG1 | 0x02A8 | PLL_VIDEO0 Pattern Control Register 1 |
| PLL_DDR_PAT_CTRL_REG1 | 0x02B0 | PLL_DDR Pattern Control Register 1 |
| BUS_SOFT_RST_REG0 | 0x02C0 | Bus Software Reset Register 0 |
| BUS_SOFT_RST_REG1 | 0x02C4 | Bus Software Reset Register 1 |
| BUS_SOFT_RST_REG2 | 0x02C8 | Bus Software Reset Register 2 |
| BUS_SOFT_RST_REG3 | 0x02D0 | Bus Software Reset Register 3 |
| BUS_SOFT_RST_REG4 | 0x02D8 | Bus Software Reset Register 4 |

## 3.3.5.    Register Description

### 3.3.5.1.   PLL_C0CPUX Control Register (Default Value: 0x02001100)

| Offset: 0x0000 | | | Register Name: **PLL_C0CPUX_CTRL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x0 | PLL_ENABLE<br>0: Disable<br>1: Enable.<br>The PLL Output= 24MHz*N/P.<br>The PLL output is for the Cluster 0 CPU0/1/2/3 Clock.<br>Note: the PLL output clock must be in the range of 200MHz~3GHz.<br>Its default is 408MHz. |
| 30:27 | / | / | / |
| 26:24 | R/W | 0x2 | PLL_LOCK_TIME<br>PLL Lock Time. |
| 23:21 | / | / | / |
| 20 | R/W | 0x0 | CLOCK_OUTPUT<br>0: Enable<br>1: Disable. |
| 19:17 | / | / | / |
| 16 | R/W | 0x0 | PLL_OUT_EXT_DIVP |

| | | | PLL Output External Divider P. |
|---|---|---|---|
| | | | 0: /1 |
| | | | 1: /4. |
| | | | Note:The P factor only use in the condition that PLL output less than 288 MHz. |
| 15:8 | R/W | 0x11 | PLL_FACTOR_N |
| | | | PLL Factor N. |
| | | | The range is from 0 to 255 (In application, Factor N should be no less than 12 ) |
| 7:2 | / | / | / |
| 1:0 | R/W | 0x0 | PLL_POSTDIV_M |
| | | | Post Divider Factor M. |
| | | | The range is from 0 to 3. This factor is only for test. |

### 3.3.5.2. PLL_C1CPUX Control Register (Default Value: 0x02001100)

| Offset: 0x0004 | | | Register Name: **PLL_C1CPUX_CTRL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x0 | PLL_ENABLE |
| | | | 0: Disable. |
| | | | 1: Enable. |
| | | | The PLL Output= 24MHz*N/P. |
| | | | The PLL output is for the Cluster 1 CPU0/1/2/3 Clock. |
| | | | Note: the PLL output clock must be in the range of 200MHz~3GHz. |
| | | | Its default is 408MHz. |
| 30:27 | / | / | / |
| 26:24 | R/W | 0x2 | PLL_LOCK_TIME |
| | | | PLL Lock Time. |
| 23:21 | / | / | / |
| 20 | R/W | 0x0 | CLOCK_OUTPUT |
| | | | 0: Enable |
| | | | 1: Disable. |
| 19:17 | / | / | / |
| 16 | R/W | 0x0 | PLL_OUT_EXT_DIVP |
| | | | PLL Output External Divider P. |
| | | | 0: /1 |
| | | | 1: /4. |
| | | | Note:The P factor only use in the condition that PLL output less than 288 MHz. |
| 15:8 | R/W | 0x11 | PLL_FACTOR_N. |
| | | | PLL Factor N. |
| | | | The range is from 0 to 255 (In application, Factor N should be no less than 12 ) |
| 7:2 | / | / | / |

| 1:0 | R/W | 0x0 | PLL_POSTDIV_M |
| | | | Post Divider Factor M. |
| | | | The range is from 0 to 3. This factor is only for test. |

### 3.3.5.3.   PLL_Audio Control Register (Default Value: 0x00042B14)

| Offset: 0x0008 | | | Register Name: **PLL_AUDIO_CTRL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x0 | PLL_ENABLE |
| | | | 0: Disable. |
| | | | 1: Enable. |
| | | | This PLL is for Audio. |
| | | | The PLL Output = 24MHz*N/(Div1+1)/(Div2+1)/(P+1). |
| | | | Its default is 24.5714 MHz. |
| 30:25 | / | / | / |
| 24 | R/W | 0x0 | PLL_SDM_EN |
| | | | 0: Disable. |
| | | | 1: Enable. |
| 23:21 | / | / | / |
| 20 | R/W | 0x0 | CLOCK_OUTPUT |
| | | | 0: Enable |
| | | | 1: Disable. |
| 19 | / | / | / |
| 18 | R/W | 0x1 | PLL_DIV2 |
| | | | PLL Div2 Factor = 0 or 1 (Output Div). |
| 17 | / | / | / |
| 16 | R/W | 0x0 | PLL_DIV1 |
| | | | PLL Div1 Factor = 0 or 1 (Input Div). |
| 15:8 | R/W | 0x2B | PLL_FACTOR_N |
| | | | PLL Factor N. |
| | | | The range is from 0 to 255 (In application, Factor N should be no less than 12 ) |
| 7:6 | / | / | / |
| 5:0 | R/W | 0x14 | PLL_POSTDIV_P |
| | | | Post Divider Factor P. |
| | | | The range is from 0 to 63 |

### 3.3.5.4.   PLL_VIDEO0 Control Register (Default Value: 0x00016300)

| Offset: 0x0010 | | | Register Name: **PLL_VIDEO0_CTRL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x0 | PLL_ENABLE. |

| | | | 0: Disable |
|---|---|---|---|
| | | | 1: Enable. |
| | | | This PLL is for the Video . |
| | | | The PLL Output = 24MHz*N/(Div+1)/P. |
| | | | Its default is 1188 MHz. |
| 30:25 | / | / | / |
| 24 | R/W | 0x0 | PLL_SDM_EN. |
| | | | 0: Disable |
| | | | 1: Enable. |
| 23:21 | / | / | / |
| 20 | R/W | 0x0 | CLOCK_OUTPUT |
| | | | 0: Enable |
| | | | 1: Disable. |
| 19 | / | / | / |
| 18 | R/W | 0x0 | PLL_DIV2. |
| | | | PLL Div2 Factor = 0 or 1 (Output Div).This factor is only for test. |
| 17 | / | / | / |
| 16 | R/W | 0x1 | PLL_DIV. |
| | | | PLL Div Factor = 0 or 1 (Input Div). |
| 15:8 | R/W | 0x63 | PLL_FACTOR_N. |
| | | | PLL Factor N. |
| | | | The range is from 0 to 255 (In application, Factor N should be no less than 12 ). |
| 7:2 | / | / | / |
| 1:0 | R/W | 0x0 | PLL_OUT_EXT_DIVP. |
| | | | PLL Output External Divider P. |
| | | | 00: /1 |
| | | | 01: /2 |
| | | | 10: /4 |
| | | | 11: /8. |

### 3.3.5.5. PLL_VE Control Register (Default Value: 0x00042400)

| Offset: 0x0018 | | | Register Name: **PLL_VE_CTRL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x0 | PLL_ENABLE. |
| | | | 0: Disable |
| | | | 1: Enable. |
| | | | This PLL is for the VE. |
| | | | The PLL Output = 24MHz*N/(Div1+1)/(Div2+1). |
| | | | Its default is 432 MHz. |
| 30:21 | / | / | / |
| 20 | R/W | 0x0 | CLOCK_OUTPUT |
| | | | 0: Enable |

| Bit | R/W | Default/Hex | Description |
|---|---|---|---|
| | | | 1: Disable. |
| 19 | / | / | / |
| 18 | R/W | 0x1 | PLL_DIV2. <br> PLL Div2 Factor = 0 or 1(Output Div). |
| 17 | / | / | / |
| 16 | R/W | 0x0 | PLL_DIV1. <br> PLL Div1 Factor = 0 or 1.(Input Div). |
| 15:8 | R/W | 0x24 | PLL_FACTOR_N. <br> PLL Factor N. <br> The range is from 0 to 255 (In application, Factor N should be no less than 12 ). |
| 7:0 | / | / | / |

### 3.3.5.6. PLL_DDR Control Register (Default Value: 0x00042400)

| Offset: 0x0020 | | | Register Name: **PLL_DDR_CTRL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x0 | PLL_ENABLE. <br> 0: Disable <br> 1: Enable. <br> This PLL is for the DRAM. <br> The PLL Output = 24MHz*(N+1)/(Div1+1)/(Div2+1). <br> Its default is 432 MHz. |
| 30 | R/W | 0x0 | SDRPLL_UPD. <br> SDRPLL Configuration Update. <br> Note: When PLL_DDR has changed, this bit should be set to 1 to validate the PLL, otherwise the change is invalid. It will be auto cleared after the PLL is valid. <br> 0: No effect. <br> 1: To validate the PLL_DDR. |
| 29:23 | / | / | / |
| 24 | R/W | 0x0 | PLL_SDM_EN. <br> 0: Disable <br> 1: Enable. |
| 23:22 | / | / | / |
| 21 | R/W | 0x0 | DDR_CLOCK_Select <br> 0:PLL_DDR <br> 1:PLL_PERIPH |
| 20 | R/W | 0x0 | CLOCK_OUTPUT <br> 0: Enable <br> 1: Disable. |
| 19 | / | / | / |
| 18 | R/W | 0x1 | PLL_DIV2. |

| | | | PLL Div2 Factor = 0 or 1 (Output Div). |
|---|---|---|---|
| 17 | / | / | / |
| 16 | R/W | 0x0 | PLL_DIV1. <br> PLL Div1 Factor = 0 or 1 (Input Div). |
| 15:14 | / | / | / |
| 13:8 | R/W | 0x24 | PLL_FACTOR_N. <br> PLL Factor N. <br> The range is from 0 to 255 (In application, Factor N should be no less than 12 ). |
| 7:0 | / | / | / |

### 3.3.5.7. PLL_PERIPH Control Register (Default Value: 0x00001900)

| Offset: 0x0028 | | | Register Name: **PLL_PERIPH_CTRL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x0 | PLL_ENABLE. <br> 0: Disable <br> 1: Enable. <br> This PLL is for the peripheral. <br> The PLL Output = 24MHz*N/(Div1+1)/(Div2+1). <br> Note: The PLL_PERIPH Output should be fixed to 600MHz, it is not recommended to vary this value arbitrarily. Its default value is 600 MHz. |
| 30:21 | / | / | / |
| 20 | R/W | 0x0 | CLOCK_OUTPUT <br> 0: Enable <br> 1: Disable. |
| 19 | / | / | / |
| 18 | R/W | 0x0 | PLL_DIV2. <br> PLL Div2 Factor = 0 or 1.(Output Div) |
| 17 | / | / | / |
| 16 | R/W | 0x0 | PLL_DIV1. <br> PLL Div1 Factor = 0 or 1.(Input Div) |
| 15:8 | R/W | 0x19 | PLL_FACTOR_N. <br> PLL Factor N. <br> The range is from 0 to 255 (In application, Factor N should be no less than 12 ). |
| 7:0 | / | / | / |

### 3.3.5.8. PLL_GPU Control Register (Default Value: 0x00042400)

| Offset: 0x0038 | | | Register Name: **PLL_GPU_CTRL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |

| 31 | R/W | 0x0 | PLL_ENABLE. 0: Disable 1: Enable. This PLL is for the GPU. The PLL Output = 24MHz*N/(Div1+1)/(Div2+1). Its default is 432 MHz. |
|---|---|---|---|
| 30:21 | / | / | / |
| 20 | R/W | 0x0 | CLOCK_OUTPUT 0: Enable 1: Disable. |
| 19 | / | / | / |
| 18 | R/W | 0x1 | PLL_DIV2. PLL Div2 Factor = 0 or 1(Output Div). |
| 17 | / | / | / |
| 16 | R/W | 0x0 | PLL_DIV1. PLL Div1 Factor = 0 or 1 (Input Div). |
| 15:8 | R/W | 0x24 | PLL_FACTOR_N. PLL Factor N. The range is from 0 to 255 (In application, Factor N should be no less than 12 ) |
| 7:0 | / | / | / |

### 3.3.5.9.  PLL_HSIC Control Register (Default Value: 0x00042800)

| Offset: 0x0044 | | | Register Name: **PLL_HSIC_CTRL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x0 | PLL_ENABLE. 0: Disable 1: Enable. This PLL is for the HSIC. The PLL Output = 24MHz*N/(Div1+1)/(Div2+1). Its default is 480 MHz. |
| 30:21 | / | / | / |
| 20 | R/W | 0x0 | CLOCK_OUTPUT 0: Enable 1: Disable. |
| 19 | / | / | / |
| 18 | R/W | 0x1 | PLL_DIV2. PLL Div2 Factor = 0 or 1(Output Div). |
| 17 | / | / | / |
| 16 | R/W | 0x0 | PLL_DIV1. PLL Div1 Factor = 0 or 1(Input Div). |
| 15:8 | R/W | 0x28 | PLL_FACTOR_N. |

| | | | PLL Factor N. The range is from 0 to 255 (In application, Factor N should be no less than 12 ). |
|---|---|---|---|
| 7:0 | / | / | / |

### 3.3.5.10. PLL_DE Control Register (Default Value: 0x00042400)

| Offset: 0x0048 | | | Register Name: **PLL_DE_CTRL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x0 | PLL_ENABLE. 0: Disable 1: Enable. This PLL is for the DE. The PLL Output = 24MHz*N/(Div1+1)/(Div2+1). Its default is 432 MHz. |
| 30:25 | / | / | / |
| 24 | R/W | 0x0 | PLL_SDM_EN. 0: Disable 1: Enable. |
| 23:21 | / | / | / |
| 20 | R/W | 0x0 | CLOCK_OUTPUT 0: Enable 1: Disable. |
| 19 | / | / | / |
| 18 | R/W | 0x1 | PLL_DIV2. PLL Div2 factor = 0 or 1(Output Div). |
| 17 | / | / | / |
| 16 | R/W | 0x0 | PLL_DIV1. PLL Div1 factor = 0 or 1(Input Div). |
| 15:8 | R/W | 0x24 | PLL_FACTOR_N. PLL Factor N. The range is from 0 to 255 (In application, Factor N should be no less than 12 ) |
| 7:0 | / | / | / |

### 3.3.5.11. PLL_VIDEO1 Control Register (Default Value: 0x00016300)

| Offset: 0x004C | | | Register Name: **PLL_VIDEO1_CTRL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x0 | PLL_ENABLE. 0: Disable 1: Enable. |

| | | | This PLL is for the Video . |
|---|---|---|---|
| | | | The PLL Output = 24MHz*N/(Div+1)/P. |
| | | | Its default is 1188 MHz. |
| 30:25 | / | / | / |
| 24 | R/W | 0x0 | PLL_SDM_EN. |
| | | | 0: Disable |
| | | | 1: Enable. |
| 23:21 | / | / | / |
| 20 | R/W | 0x0 | CLOCK_OUTPUT |
| | | | 0: Enable |
| | | | 1: Disable. |
| 19 | / | / | / |
| 18 | R/W | 0x0 | PLL_DIV2. |
| | | | PLL Div2 Factor = 0 or 1 (Output Div).This factor is only for test. |
| 17 | / | / | / |
| 16 | R/W | 0x1 | PLL_DIV. |
| | | | PLL Div Factor = 0 or 1 (Input Div). |
| 15:8 | R/W | 0x63 | PLL_FACTOR_N. |
| | | | PLL Factor N. |
| | | | The range is from 0 to 255 (In application, Factor N should be no less than 12 ) |
| 7:2 | / | / | / |
| 1:0 | R/W | 0x0 | PLL_OUT_EXT_DIVP. |
| | | | PLL Output External Divider P. |
| | | | 00: /1 |
| | | | 01: /2 |
| | | | 10: /4 |
| | | | 11: /8. |

### 3.3.5.12.  CPUX/AXI Configuration Register (Default Value: 0x00000000)

| Offset: 0x0050 | | | Register Name: **CPUX_AXI_CFG_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:29 | / | / | / |
| 28 | R/W | 0x0 | C1_CPUX_CLK_SRC_SEL. |
| | | | 0: OSC24M |
| | | | 1: PLL_C1CPUX. |
| 27:18 | / | / | / |
| 17:16 | R/W | 0x0 | AXI1_CLK_DIV_RATIO. |
| | | | AXI1 Clock Divide Ratio. |
| | | | AXI1 Clock source is C1_CPUX clock. |
| | | | 00: /1 |
| | | | 01: /2 |

| | | | 10: /3 |
| | | | 11: /4. |
|---|---|---|---|
| 15:13 | / | / | / |
| 12 | R/W | 0x0 | C0_CPUX_CLK_SRC_SEL. |
| | | | 0: OSC24M |
| | | | 1: PLL_C0CPUX. |
| 11:2 | / | / | / |
| 1:0 | R/W | 0x0 | AXI0_CLK_DIV_RATIO. |
| | | | AXI0 Clock Divide Ratio. |
| | | | AXI0 Clock source is C0_CPUX clock. |
| | | | 00: /1 |
| | | | 01: /2 |
| | | | 10: /3 |
| | | | 11: /4. |

### 3.3.5.13. AHB1/APB1 Configuration Register (Default Value: 0x00000000)

| Offset: 0x0054 | | | Register Name: **AHB1_APB1_CFG_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:14 | / | / | / |
| 13:12 | R/W | 0x1 | AHB1_CLK_SRC_SEL. |
| | | | 00: Internal OSC / 512 |
| | | | 01: OSC24M |
| | | | 1X: PLL_PERIPH/ AHB1_PRE_DIV. |
| 11:10 | / | / | / |
| 9:8 | R/W | 0x0 | APB1_CLK_RATIO. |
| | | | APB1 Clock Divide Ratio. APB1 clock source is AHB1 clock. |
| | | | 00: /1 |
| | | | 01: /2 |
| | | | 10: /3 |
| | | | 11: /4. |
| 7:6 | R/W | 0x0 | AHB1_PRE_DIV |
| | | | AHB1 Clock Pre Divide Ratio |
| | | | 00: /1 |
| | | | 01: /2 |
| | | | 10: /3 |
| | | | 11: /4. |
| 5:4 | R/W | 0x1 | AHB1_CLK_DIV_RATIO. |
| | | | AHB1 Clock Divide Ratio. |
| | | | 00: /1 |
| | | | 01: /2 |
| | | | 10: /4 |
| | | | 11: /8. |
| 3:0 | / | / | / |

### 3.3.5.14. APB2 Configuration Register (Default Value: 0x01000000)

| Offset: 0x0058 | | | Register Name: **APB2_CFG_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:26 | / | / | / |
| 25:24 | R/W | 0x1 | APB2_CLK_SRC_SEL.<br>APB2 Clock Source Select<br>00: Internal OSC / 512<br>01: OSC24M<br>1X: PLL_PERIPH.<br>This clock is used for some special module apbclk(UART、TWI). Because these modules need special clock rate even if the apb1clk changed. |
| 23:18 | / | / | / |
| 17:16 | R/W | 0x0 | CLK_RAT_N<br>Clock Pre Divide Ratio (n)<br>The select clock source is pre-divided by 2^n. The divider is 1/2/4/8. |
| 15:5 | / | / | / |
| 4:0 | R/W | 0x0 | CLK_RAT_M.<br>Clock Divide Ratio (m)<br>The Pre Divide clock is divided by (m+1). The divider M is from 1 to 32. |

### 3.3.5.15. AHB2 Configuration Register (Default Value: 0x00000000)

| Offset: 0x005C | | | Register Name: **AHB2_CFG_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:2 | / | / | / |
| 1:0 | R/W | 0x0 | AHB2_CLK_CFG.<br>00: AHB1 Clock<br>01: PLL_PERIPH / 2<br>1X: /.<br>EMAC and USBHOST clock source is AHB2 Clock. |

### 3.3.5.16. Bus Clock Gating Register0 (Default Value: 0x00000000)

| Offset: 0x0060 | | | Register Name: **BUS_CLK_GATING_REG0** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:30 | / | / | / |
| 29 | R/W | 0x0 | USB_OHCl0_GATING.<br>0: Mask<br>1: Pass. |

| 28 | / | / | / |
|---|---|---|---|
| 27 | R/W | 0x0 | USB_EHCI1_GATING. <br> 0: Mask <br> 1: Pass. |
| 26 | R/W | 0x0 | USB_EHCI0_GATING. <br> 0: Mask <br> 1: Pass. |
| 25 | / | / | / |
| 24 | R/W | 0x0 | USB_DRD_GATING. <br> 0: Mask <br> 1: Pass. |
| 23:22 | / | / | / |
| 21 | R/W | 0x0 | SPI1_GATING. <br> 0: Mask <br> 1: Pass. |
| 20 | R/W | 0x0 | SPI0_GATING. <br> 0: Mask <br> 1: Pass. |
| 19 | R/W | 0x0 | HSTIMER_GATING. <br> 0: Mask <br> 1: Pass. |
| 18:15 | / | / | / |
| 17 | R/W | 0x0 | EMAC_GATING. <br> 0: Mask <br> 1: Pass. |
| 16:15 | / | / | / |
| 14 | R/W | 0x0 | DRAM_GATING. <br> 0: Mask <br> 1: Pass. |
| 13 | R/W | 0x0 | NAND_GATING. <br> 0: Mask <br> 1: Pass. |
| 12:11 | / | / | / |
| 10 | R/W | 0x0 | MMC2_GATING. <br> 0: Mask <br> 1: Pass. |
| 9 | R/W | 0x0 | MMC1_GATING. <br> 0: Mask <br> 1: Pass. |
| 8 | R/W | 0x0 | MMC0_GATING. <br> 0: Mask <br> 1: Pass. |
| 7 | / | / | / |
| 6 | R/W | 0x0 | DMA_GATING. <br> 0: Mask |

| | | | 1: Pass. |
|---|---|---|---|
| 5 | R/W | 0x0 | SS_GATING.<br>0: Mask<br>1: Pass. |
| 4:2 | / | / | / |
| 1 | R/W | 0x0 | MIPIDSI_GATING.<br>0: Mask<br>1: Pass. |
| 0 | / | / | / |

### 3.3.5.17. Bus Clock Gating Register1 (Default Value: 0x00000000)

| Offset: 0x0064 | | | Register Name: **BUS_CLK_GATING_REG1** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:23 | / | / | / |
| 22 | R/W | 0x0 | SPINLOCK_GATING.<br>0: Mask<br>1: Pass. |
| 21 | R/W | 0x0 | MSGBOX_GATING.<br>0: Mask<br>1: Pass. |
| 20 | R/W | 0x0 | GPU_GATING.<br>0: Mask<br>1: Pass. |
| 19:13 | / | / | / |
| 12 | R/W | 0x0 | DE_GATING.<br>0: Mask<br>1: Pass. |
| 11 | R/W | 0x0 | HDMI_GATING.<br>0: Mask<br>1: Pass. |
| 10:9 | / | / | / |
| 8 | R/W | 0x0 | CSI_GATING.<br>0: Mask<br>1: Pass. |
| 7:6 | / | / | / |
| 5 | R/W | 0x0 | TCON1_GATING.<br>0: Mask<br>1: Pass. |
| 4 | R/W | 0x0 | TCON0_GATING.<br>0: Mask<br>1: Pass. |
| 3:1 | / | / | / |

| 0 | R/W | 0x0 | VE_GATING.<br>0: Mask<br>1: Pass. |
|---|-----|-----|-----------------------------------|

### 3.3.5.18. Bus Clock Gating Register2 (Default Value: 0x00000000)

| Offset: 0x0068 | | | Register Name: **BUS_CLK_GATING_REG2** |
|----------------|-----|-------------|----------------------------------------|
| Bit | R/W | Default/Hex | Description |
| 31:16 | / | / | / |
| 15 | R/W | 0x0 | TDM_GATING.<br>0: Mask<br>1: Pass. |
| 14 | R/W | 0x0 | DAUDIO2_GATING.<br>0: Mask<br>1: Pass. |
| 13 | R/W | 0x0 | DAUDIO1_GATING.<br>0: Mask<br>1: Pass. |
| 12 | R/W | 0x0 | DAUDIO0_GATING.<br>0: Mask<br>1: Pass. |
| 11:6 | / | / | / |
| 5 | R/W | 0x0 | PIO_GATING.<br>0: Mask<br>1: Pass. |
| 4:2 | / | / | / |
| 1 | R/W | 0x0 | OWA_GATING.<br>0: Mask<br>1: Pass. |
| 0 | / | / | / |

### 3.3.5.19. Bus Clock Gating Register3 (Default Value: 0x00000000)

| Offset: 0x006C | | | Register Name: **BUS_CLK_GATING_REG3** |
|----------------|-----|-------------|----------------------------------------|
| Bit | R/W | Default/Hex | Description |
| 31:21 | / | / | / |
| 20 | R/W | 0x0 | UART4_GATING.<br>0: Mask<br>1: Pass. |
| 19 | R/W | 0x0 | UART3_GATING.<br>0: Mask<br>1: Pass. |

| 18 | R/W | 0x0 | UART2_GATING.<br>0: Mask<br>1: Pass. |
|---|---|---|---|
| 17 | R/W | 0x0 | UART1_GATING.<br>0: Mask<br>1: Pass. |
| 16 | R/W | 0x0 | UART0_GATING.<br>0: Mask<br>1: Pass. |
| 15:3 | / | / | / |
| 2 | R/W | 0x0 | TWI2_GATING.<br>0: Mask<br>1: Pass. |
| 1 | R/W | 0x0 | TWI1_GATING.<br>0: Mask<br>1: Pass. |
| 0 | R/W | 0x0 | TWI0_GATING.<br>0: Mask<br>1: Pass. |

### 3.3.5.20. CCI-400 Configuration Register (Default Value: 0x00000000)

| Offset: 0x0078 | | | Register Name: **CCI400_CLK_CFG_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:26 | / | / | / |
| 25:24 | R/W | 0x0 | CCI400_CLK_SRC_SEL.<br>Clock Source Select<br>00: OSC24M<br>01: PLL_PERIPH<br>10: PLL_HSIC<br>11: /. |
| 23:2 | / | / | / |
| 1:0 | R/W | 0x0 | CCI400_CLK_DIV_RATIO.<br>CCI-400 Clock Divide Ratio.<br>00: /1<br>01: /2<br>10: /3<br>11: /4.<br>Note: When the clock source changed from OSC24M to PLL_PERIPH or PLL_HSIC, it must follow these procedure:<br>1. The CCI-400 clock divide ratio should be changed at first.<br>2. It must wait for at least 10 CCI-400 clock cycles before changing clock source. |

### 3.3.5.21.  NAND Clock Register (Default Value: 0x00000000)

| Offset: 0x0080 | | | Register Name: **NAND_CLK_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x0 | SCLK_GATING. Gating Special Clock(Max Clock = 200MHz) 0: Clock is OFF 1: Clock is ON. SCLK = Clock Source/Divider N/Divider M. |
| 30:26 | / | / | / |
| 25:24 | R/W | 0x0 | CLK_SRC_SEL. Clock Source Select 00: OSC24M 01: PLL_PERIPH 1X: /. |
| 23:18 | / | / | / |
| 17:16 | R/W | 0x0 | CLK_DIV_RATIO_N. Clock Pre Divide Ratio (n) 00: /1 01: /2 10: /4 11: /8. |
| 15:4 | / | / | / |
| 3:0 | R/W | 0x0 | CLK_DIV_RATIO_M Clock Divide Ratio (m) The pre-divided clock is divided by (m+1). The divider M is from 1 to 16. |

### 3.3.5.22.  SDMMC0 Clock Register (Default Value: 0x00000000)

| Offset: 0x0088 | | | Register Name: **SDMMC0_CLK_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x0 | SCLK_GATING. Gating Special Clock(Max Clock = 200MHz) 0: Clock is OFF 1: Clock is ON. SCLK = Clock Source/Divider N/Divider M. |
| 30:26 | / | / | / |
| 25:24 | R/W | 0x0 | CLK_SRC_SEL. Clock Source Select 00: OSC24M 01: PLL_PERIPH 1X: /. |
| 23 | / | / | / |

| 22:20 | R/W | 0x0 | SAMPLE_CLK_PHASE_CTR. |
| | | | Sample Clock Phase Control. |
| | | | The sample clock phase delay is based on the number of source clock that is from 0 to 7. |
| 19:18 | / | / | / |
| 17:16 | R/W | 0x0 | CLK_DIV_RATIO_N. |
| | | | Clock Pre Divide Ratio (n) |
| | | | 00: /1 |
| | | | 01: /2 |
| | | | 10: /4 |
| | | | 11: /8. |
| 15:11 | / | / | / |
| 10:8 | R/W | 0x0 | OUTPUT_CLK_PHASE_CTR. |
| | | | Output Clock Phase Control. |
| | | | The output clock phase delay is based on the number of source clock that is from 0 to 7. |
| 7:4 | / | / | / |
| 3:0 | R/W | 0x0 | CLK_DIV_RATIO_M. |
| | | | Clock Divide Ratio (m) |
| | | | The pre-divided clock is divided by (m+1). The divider M is from 1 to 16. |

### 3.3.5.23. SDMMC1 Clock Register (Default Value: 0x00000000)

| Offset: 0x008C | | | Register Name: **SDMMC1_CLK_REG** |
| --- | --- | --- | --- |
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x0 | SCLK_GATING. |
| | | | Gating Special Clock(Max Clock = 200MHz) |
| | | | 0: Clock is OFF |
| | | | 1: Clock is ON. |
| | | | SCLK= Clock Source/Divider N/Divider M. |
| 30:26 | / | / | / |
| 25:24 | R/W | 0x0 | CLK_SRC_SEL. |
| | | | Clock Source Select |
| | | | 00: OSC24M |
| | | | 01: PLL_PERIPH |
| | | | 1X: /. |
| 23 | / | / | / |
| 22:20 | R/W | 0x0 | SAMPLE_CLK_PHASE_CTR. |
| | | | Sample Clock Phase Control. |
| | | | The sample clock phase delay is based on the number of source clock that is from 0 to 7. |
| 19:18 | / | / | / |
| 17:16 | R/W | 0x0 | CLK_DIV_RATIO_N. |
| | | | Clock Pre-Divide Ratio (n) |

| | | | 00: /1 |
| --- | --- | --- | --- |
| | | | 01: /2 |
| | | | 10: /4 |
| | | | 11: /8. |
| 15:11 | / | / | / |
| 10:8 | R/W | 0x0 | OUTPUT_CLK_PHASE_CTR. |
| | | | Output Clock Phase Control. |
| | | | The output clock phase delay is based on the number of source clock that is from 0 to 7. |
| 7:4 | / | / | / |
| 3:0 | R/W | 0x0 | CLK_DIV_RATIO_M. |
| | | | Clock Divide Ratio (m) |
| | | | The pre-divided clock is divided by (m+1). The divider M is from 1 to 16. |

### 3.3.5.24. SDMMC2 Clock Register (Default Value: 0x00000000)

| Offset: 0x0090 | | | Register Name: **SDMMC2_CLK_REG** |
| --- | --- | --- | --- |
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x0 | SCLK_GATING. |
| | | | Gating Special Clock(Max Clock = 200MHz) |
| | | | 0: Clock is OFF |
| | | | 1: Clock is ON. |
| | | | If SDMMC2 is in old mode, SCLK = Clock Source/Divider N/Divider M. |
| | | | If SDMMC2 is in new mode, SCLK= Clock Source/Divider N/Divider M/2. |
| 30 | R/W | 0x0 | MMC2_MODE_SELECT. |
| | | | 0: Old Mode |
| | | | 1: New Mode. |
| 29:26 | / | / | / |
| 25:24 | R/W | 0x0 | CLK_SRC_SEL. |
| | | | Clock Source Select |
| | | | 00: OSC24M |
| | | | 01: PLL_PERIPH |
| | | | 1X: /. |
| 23 | / | / | / |
| 22:20 | R/W | 0x0 | CLK_PHASE_CTR. |
| | | | Sample Clock Phase Control. |
| | | | The sample clock phase delay is based on the number of source clock that is from 0 to 7. |
| 19:18 | / | / | / |
| 17:16 | R/W | 0x0 | CLK_DIV_RATIO_N. |
| | | | Clock Pre Divide Ratio (n) |
| | | | 00: /1 |
| | | | 01: /2 |
| | | | 10: /4 |

| | | | 11: /8. |
|---|---|---|---|
| 15:11 | / | / | / |
| 10:8 | R/W | 0x0 | OUTPUT_CLK_PHASE_CTR. Output Clock Phase Control. The output clock phase delay is based on the number of source clock that is from 0 to 7. |
| 7:4 | / | / | / |
| 3:0 | R/W | 0x0 | CLK_DIV_RATIO_M. Clock Divide Ratio (m) The pre-divided clock is divided by (m+1). The divider M is from 1 to 16. |

### 3.3.5.25. SS Clock Register (Default Value: 0x00000000)

| Offset: 0x009C | | | Register Name: **SS_CLK_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x0 | SCLK_GATING. Gating Special Clock(Max Clock = 400MHz) 0: Clock is OFF 1: Clock is ON. SCLK = Clock Source/Divider N/Divider M. |
| 30:26 | / | / | / |
| 25:24 | R/W | 0x0 | CLK_SRC_SEL. Clock Source Select 00: OSC24M 01: PLL_PERIPH 1X: /. |
| 23:18 | / | / | / |
| 17:16 | R/W | 0x0 | CLK_DIV_RATIO_N. Clock Pre Divide Ratio (n) 00: /1 01: /2 10: /4 11: /8. |
| 15:4 | / | / | / |
| 3:0 | R/W | 0x0 | CLK_DIV_RATIO_M. Clock divide ratio (m) The pre-divided clock is divided by (m+1). The divider M is from 1 to 16. |

### 3.3.5.26. SPI0 Clock Register (Default Value: 0x00000000)

| Offset: 0x00A0 | | | Register Name: **SPI0_CLK_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x0 | SCLK_GATING. |

| | | | Gating Special Clock(Max Clock = 200MHz)<br>0: Clock is OFF<br>1: Clock is ON.<br>SCLK = Clock Source/Divider N/Divider M. |
|---|---|---|---|
| 30:26 | / | / | / |
| 25:24 | R/W | 0x0 | CLK_SRC_SEL.<br>Clock Source Select<br>00: OSC24M<br>01: PLL_PERIPH<br>1X: /. |
| 23:18 | / | / | / |
| 17:16 | R/W | 0x0 | CLK_DIV_RATIO_N.<br>Clock Pre Divide Ratio (n)<br>00: /1<br>01: /2<br>10: /4<br>11: /8. |
| 15:4 | / | / | / |
| 3:0 | R/W | 0x0 | CLK_DIV_RATIO_M.<br>Clock Divide Ratio (m)<br>The pre-divided clock is divided by (m+1). The divider M is from 1 to 16. |

### 3.3.5.27. SPI1 Clock Register (Default Value: 0x00000000)

| Offset: 0x00A4 | | | Register Name: **SPI1_CLK_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x0 | SCLK_GATING.<br>Gating Special Clock(Max Clock = 200MHz)<br>0: Clock is OFF<br>1: Clock is ON.<br>SCLK= Clock Source/Divider N/Divider M. |
| 30:26 | / | / | / |
| 25:24 | R/W | 0x0 | CLK_SRC_SEL.<br>Clock Source Select<br>00: OSC24M<br>01: PLL_PERIPH<br>1X: /. |
| 23:18 | / | / | / |
| 17:16 | R/W | 0x0 | CLK_DIV_RATIO_N.<br>Clock Pre Divide Ratio (n)<br>00: /1<br>01: /2<br>10: /4<br>11: /8. |

| 15:4 | / | / | / |
|------|-----|-----|-----|
| 3:0 | R/W | 0x0 | CLK_DIV_RATIO_M. <br> Clock Divide Ratio (m) <br> The pre-divided clock is divided by (m+1). The divider M is from 1 to 16. |

### 3.3.5.28. DAUDIO0 Clock Register (Default Value: 0x00000000)

| Offset: 0x00B0 | | | Register Name: **DAUDIO0_CLK_REG** |
|------|-----|-----|-----|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x0 | SCLK_GATING. <br> Gating Special Clock(Max Clock = 200MHz) <br> 0: Clock is OFF <br> 1: Clock is ON. <br> SCLK= Clock Source PLL_AUDIO/Divider M. |
| 30:4 | / | / | / |
| 3:0 | R/W | 0x0 | CLK_DIV_RATIO_M. <br> Clock Divide Ratio (m) <br> The pre-divided clock is divided by (m+1). The divider M is from 1 to 16. |

### 3.3.5.29. DAUDIO1 Clock Register (Default Value: 0x00000000)

| Offset: 0x00B4 | | | Register Name: **DAUDIO1_CLK_REG** |
|------|-----|-----|-----|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x0 | SCLK_GATING. <br> Gating Special Clock(Max Clock = 200MHz) <br> 0: Clock is OFF <br> 1: Clock is ON. <br> SCLK= Clock Source PLL_AUDIO/Divider M. |
| 30:4 | / | / | / |
| 3:0 | R/W | 0x0 | CLK_DIV_RATIO_M. <br> Clock Divide Ratio (m) <br> The pre-divided clock is divided by (m+1). The divider M is from 1 to 16. |

### 3.3.5.30. DAUDIO2 Clock Register (Default Value: 0x00000000)

| Offset: 0x00B8 | | | Register Name: **DAUDIO2_CLK_REG** |
|------|-----|-----|-----|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x0 | SCLK_GATING. <br> Gating Special Clock(Max Clock = 200MHz) <br> 0: Clock is OFF <br> 1: Clock is ON. |

| | | | SCLK= Clock Source PLL_AUDIO/Divider M. |
|---|---|---|---|
| 30:4 | / | / | / |
| 3:0 | R/W | 0x0 | CLK_DIV_RATIO_M.<br>Clock Divide Ratio (m)<br>The pre-divided clock is divided by (m+1). The divider M is from 1 to 16. |

### 3.3.5.31. TDM Clock Register (Default Value: 0x00000000)

| Offset: 0x00BC | | | Register Name: **TDM_CLK_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x0 | SCLK_GATING.<br>Gating Special Clock(Max Clock = 200MHz)<br>0: Clock is OFF<br>1: Clock is ON.<br>SCLK= PLL_AUDIO/Divider M. |
| 30:4 | / | / | / |
| 3:0 | R/W | 0x0 | CLK_DIV_RATIO_M.<br>Clock Divide Ratio (m)<br>The pre-divided clock is divided by (m+1). The divider M is from 1 to 16. |

### 3.3.5.32. OWA Clock Register (Default Value: 0x00000000)

| Offset: 0x00C0 | | | Register Name: **OWA_CLK_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x0 | SCLK_GATING.<br>Gating Special Clock(Max Clock = 200MHz)<br>0: Clock is OFF<br>1: Clock is ON.<br>SCLK= PLL_AUDIO/Divider M. |
| 30:4 | / | / | / |
| 3:0 | R/W | 0x0 | CLK_DIV_RATIO_M.<br>Clock Divide Ratio (m)<br>The pre-divided clock is divided by (m+1). The divider M is from 1 to 16. |

### 3.3.5.33. USBPHY Clock Register (Default Value: 0x00000000)

| Offset: 0x00CC | | | Register Name: **USBPHY_CLK_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:17 | / | / | / |
| 16 | R/W | 0x0 | SCLK_GATING_OHCI0.<br>Gating Special Clock for OHCI0 |

| | | | 0: Clock is OFF |
|---|---|---|---|
| | | | 1: Clock is ON. |
| 15:12 | / | / | / |
| 11 | R/W | 0 | SCLK_GATING_12M |
| | | | Gating Special 12M Clock for HSIC |
| | | | 0: Clock is OFF |
| | | | 1: Clock is ON. |
| | | | The special 12M clock = OSC24M/2. |
| 10 | R/W | 0 | SCLK_GATING_HSIC |
| | | | Gating Special Clock for HSIC |
| | | | 0: Clock is OFF |
| | | | 1: Clock is ON. |
| | | | The special clock is from PLL_HSIC. |
| 9 | R/W | 0x0 | SCLK_GATING_USBPHY1. |
| | | | Gating Special Clock for USB PHY1 |
| | | | 0: Clock is OFF |
| | | | 1: Clock is ON. |
| 8 | R/W | 0x0 | SCLK_GATING_USBPHY0. |
| | | | Gating Special Clock for USB PHY0(USB DRD) |
| | | | 0: Clock is OFF |
| | | | 1: Clock is ON. |
| 7:3 | / | / | / |
| 2 | R/W | 0 | USBHSIC_RST |
| | | | USB HSIC Reset Control |
| | | | 0: Assert |
| | | | 1: De-assert. |
| 1 | R/W | 0x0 | USBPHY1_RST. |
| | | | USB PHY1 Reset Control |
| | | | 0: Assert |
| | | | 1: De-assert. |
| 0 | R/W | 0x0 | USBPHY0_RST. |
| | | | USB PHY0 Reset Control |
| | | | 0: Assert |
| | | | 1: De-assert. |

### 3.3.5.34. DRAM Clock Register (Default Value: 0x00000001)

| Offset: 0x00F4 | | | Register Name: **DRAM_CLK_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x0 | DRAM_CTR_RST. |
| | | | DRAM Controller Reset For AHB Clock Domain. |
| | | | 0: Assert |
| | | | 1: De-assert. |
| 30:15 | | | |

| 16 | R/W | 0x0 | SDRCLK_UPD.<br>SDRCLK Configuration Update.<br>0:Invalid<br>1:Valid.<br>Note: Set this bit will validate Configuration . It will be auto cleared after the Configuration is valid.<br>The DRAMCLK Source is from PLL_DDR. |
| 15:4 | / | / | / |
| 3:0 | R/W | 0x1 | DRAM_DIV_M.<br>DRAMCLK Divider of Configuration.<br>The clock is divided by (m+1). The divider M should be from 2 to 16. |

### 3.3.5.35. PLL_DDR Configuration Register (Default Value: 0x00000030)

| Offset: 0x00F8 | | | Register Name: **PLL_DDR_CFG_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:13 | / | / | / |
| 12 | R/W | 0x0 | PLL_DDR_MODE.<br>0: Normal Mode<br>1: Continuously Frequency Scale. |
| 11:7 | / | / | / |
| 6:4 | R/W | 0x3 | PLL_DDR_PHASE_COMPENSATE.<br>The value of bit[6:4] is based on 24M clock, then the default PLL_DDR phase compensate is (3/24000000) s. |
| 3:0 | R/W | 0x0 | PLL_DDR_STEP.<br>0000: 0.004MHz/us (576/2^17)<br>0001: 0.008MHz/us (576/2^16)<br>0010: 0.016MHz/us (576/2^15)<br>0011: 0.032MHz/us (576/2^14)<br>0100: 0.064MHz/us (576/2^13)<br>0101: 0.128MHz/us (576/2^12)<br>0110: 0.256MHz/us (576/2^11)<br>0111: 0.512MHz/us (576/2^10)<br>1000: 1.024MHz/us (576/2^9)<br>1001: 2.048MHz/us (576/2^8)<br>Others: 0.004MHz/us (576/2^17). |

### 3.3.5.36. MBUS Reset Register (Default Value: 0x80000000)

| Offset: 0x00FC | | | Register Name: **MBUS_RST_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x1 | MBUS_RESET. |

| | | | 0: Reset Mbus Domain |
| | | | 1: Assert Mbus Domain. |
| 30:0 | / | / | / |

### 3.3.5.37. DRAM Clock Gating Register (Default Value: 0x00000000)

| Offset: 0x0100 | | | Register Name: **DRAM_CLK_GATING_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:2 | / | / | / |
| 1 | R/W | 0x0 | CSI_DCLK_GATING. |
| | | | Gating DRAM Clock For CSI |
| | | | 0: Mask |
| | | | 1: Pass. |
| 0 | R/W | 0x0 | VE_DCLK_GATING. |
| | | | Gating DRAM Clock For VE |
| | | | 0: Mask |
| | | | 1: Pass. |

### 3.3.5.38. TCON0 Clock Register (Default Value: 0x00000000)

| Offset: 0x0118 | | | Register Name: **TCON0_CLK_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x0 | SCLK_GATING. |
| | | | Gating Special Clock |
| | | | 0: Clock is OFF |
| | | | 1: Clock is ON. |
| 30:27 | / | / | / |
| 26:24 | R/W | 0x0 | CLK_SRC_SEL. |
| | | | Clock Source Select |
| | | | 000: PLL_VIDEO0 |
| | | | Others: /. |
| 23:0 | / | / | / |

### 3.3.5.39. TCON1 Clock Register (Default Value: 0x00000000)

| Offset: 0x011C | | | Register Name: **TCON1_CLK_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x0 | SCLK_GATING. |
| | | | Gating Special Clock |
| | | | 0: Clock is OFF |
| | | | 1: Clock is ON. |

| | | | SCLK = Clock Source/ Divider M. |
|---|---|---|---|
| 30:26 | / | / | / |
| 25:24 | R/W | 0x0 | SCLK_SEL.<br>Special Clock Source Select<br>00: PLL_VIDEO1<br>Others: /. |
| 23:4 | / | / | / |
| 3:0 | R/W | 0x0 | CLK_DIV_RATIO_M.<br>Clock Divide Ratio (m)<br>The pre-divided clock is divided by (m+1). The divider M is from 1 to 16. |

### 3.3.5.40. MIPI_CSI Clock Register (Default Value: 0x00000000)

| Offset: 0x0130 | | | Register Name: **MIPI_CSI_CLK_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x0 | MIPI_CSI_CFG.<br>0: Clock is OFF<br>1: Clock is ON.<br>This clock = OSC24M. |
| 30:17 | / | / | / |
| 16 | R/W | 0x0 | CSI_MISC_CLK_GATING.<br>Gating clock for CSI MISC, this clock is 24OSC.<br>0: Clock is OFF<br>1: Clock is ON. |
| 15:0 | / | / | / |

### 3.3.5.41. CSI Clock Register (Default Value: 0x00000000)

| Offset: 0x0134 | | | Register Name: **CSI_CLK_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x0 | CSI_SCLK_GATING.<br>Gating Special Clock<br>0: Clock is OFF<br>1: Clock is ON.<br>SCLK= Special Clock Source/CSI_SCLK_DIV_M. |
| 30:27 | / | / | / |
| 26:24 | R/W | 0x0 | SCLK_SRC_SEL.<br>Special Clock Source Select<br>000: PLL_PERIPH<br>101: PLL_VE<br>Others:/. |
| 23:20 | / | / | / |

| 19:16 | R/W | 0x0 | CSI_SCLK_DIV_M.<br>CSI Clock Divide Ratio (m)<br>The pre-divided clock is divided by (m+1). The divider M is from 1 to 16. |
|-------|-----|-----|------------------------------------------------------------------------------|
| 15 | R/W | 0x0 | CSI_MCLK_GATING.<br>Gating Master Clock<br>0: Clock is OFF<br>1: Clock is ON<br>This clock =Master Clock Source/ CSI_MCLK_DIV_M. |
| 14:11 | / | / | / |
| 10:8 | R/W | 0x0 | MCLK_SRC_SEL.<br>Master Clock Source Select<br>011: PLL_DE<br>101: OSC24M<br>Others: /. |
| 7:5 | / | / | / |
| 4:0 | R/W | 0x0 | CSI_MCLK_DIV_M.<br>CSI Master Clock Divide Ratio (m)<br>The pre-divided clock is divided by (m+1). The divider M is from 1 to 32. |

### 3.3.5.42.  VE Clock Register (Default Value: 0x00000000)

| Offset: 0x013C | | | Register Name: **VE_CLK_REG** |
|----------------|-----|-------------|-------------------------------|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x0 | VE_SCLK_GATING.<br>Gating Special Clock<br>0: Clock is OFF<br>1: Clock is ON.<br>SCLK = PLL_VE /Divider N. |
| 30:19 | / | / | /. |
| 18:16 | R/W | 0x0 | CLK_DIV_RATIO_N.<br>Clock Pre Divide Ratio (N)<br>The select clock source is pre-divided by n+1. The divider N is from 1 to 8. |
| 15:0 | / | / | / |

### 3.3.5.43.  AVS Clock Register (Default Value: 0x00000000)

| Offset: 0x0144 | | | Register Name: **AVS_CLK_REG** |
|----------------|-----|-------------|--------------------------------|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x0 | SCLK_GATING.<br>Gating Special Clock<br>0: Clock is OFF<br>1: Clock is ON. |

| | | | SCLK= OSC24M. |
|---|---|---|---|
| 30:0 | / | / | / |

### 3.3.5.44. HDMI Clock Register (Default Value: 0x00000000)

| Offset: 0x0150 | | | Register Name: **HDMI_CLK_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x0 | SCLK_GATING. Gating Special Clock 0: Clock is OFF 1: Clock is ON. SCLK= Clock Source/ Divider M. |
| 30:26 | / | / | / |
| 25:24 | R/W | 0x0 | SCLK_SEL. Special Clock Source Select 00: PLL_VIDEO1 Others: / |
| 23:4 | / | / | / |
| 3:0 | R/W | 0x0 | CLK_DIV_RATIO_M. Clock Divide Ratio (m) The pre-divided clock is divided by (m+1). The divider M is from 1 to 16. |

### 3.3.5.45. HDMI Slow Clock Register (Default Value: 0x00000000)

| Offset: 0x0154 | | | Register Name: **HDMI_SLOW_CLK_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x0 | HDMI_DDC_CLK_GATING. 0: Clock is OFF 1: Clock is ON. SCLK = OSC24M. |
| 30:0 | / | / | / |

### 3.3.5.46. MBUS Clock Register (Default Value: 0x00000000)

| Offset: 0x015C | | | Register Name: **MBUS_CLK_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x0 | MBUS_SCLK_GATING. Gating Clock for MBUS 0: Clock is OFF 1: Clock is ON. MBUS_CLOCK = Clock Source/Divider M |

| 30:26 | / | / | / |
|---|---|---|---|
| 25:24 | R/W | 0x0 | MBUS_SCLK_SRC<br>Clock Source Select<br>00: OSC24M<br>01: PLL_PERIPH<br>10: PLL_DDR<br>11: /. |
| 23:3 | / | / | / |
| 2:0 | R/W | 0x0 | MBUS_SCLK_RATIO_M<br>Clock Divide Ratio (m)<br>The pre-divided clock is divided by (m+1). The divider M is from 1 to 8.<br>Note: If the clock has been changed ,it must wait for at least 16 cycles. |

### 3.3.5.47. MIPI_DSI Clock Register0 (Default Value: 0x00000000)

| Offset: 0x0168 | | | Register Name: **MIPI_DSI_CLK_REG0** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x0 | DSI_SCLK_GATING.<br>Gating DSI Special Clock 0<br>0: Clock is OFF<br>1: Clock is ON<br>DSI Special Clock 0 = Clock Source/ CLK_DIV_RATIO_M. |
| 30:28 | / | / | / |
| 27:24 | R/W | 0x8 | DSI_SCLK0_SRC_SEL.<br>DSI Special Clock Source Select<br>1000: PLL_VIDEO0<br>1001: / |
| 23:4 | / | / | /. |
| 3:0 | R/W | 0x0 | CLK_DIV_RATIO_M.<br>Clock Divide Ratio (m)<br>The pre-divided clock is divided by (m+1). The divider M is from 1 to 16. |

### 3.3.5.48. MIPI_DSI Clock Register1 (Default Value: 0x00000000)

| Offset: 0x016C | | | Register Name: **MIPI_DSI_CLK_REG1** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x0 | DSI_SCLK_GATING.<br>Gating DSI Special Clock 1<br>0: Clock is OFF<br>1: Clock is ON.<br>DSI Special Clock = Clock Source/DSI_SCLK_DIV_M. |
| 30:28 | / | / | / |

| 27:24 | R/W | 0x0 | DSI_SCLK1_SRC_SEL. <br> DSI Special Clock Source Select <br> 0000: OSC24M <br> 1001: PLL_VIDEO0. |
|---|---|---|---|
| 23:4 | / | / | /. |
| 3:0 | R/W | 0x0 | CLK_DIV_RATIO_M. <br> Clock Divide Ratio (m) <br> The pre-divided clock is divided by (m+1). The divider M is from 1 to 16. |

### 3.3.5.49. GPU Core Clock Register (Default Value: 0x00000000)

| Offset: 0x01A0 | | | Register Name: **GPU_CORE_CLK_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x0 | SCLK_GATING. <br> 0: Clock is OFF <br> 1: Clock is ON. <br> SCLK= PLL-GPU/Divider N. |
| 30:3 | / | / | /. |
| 2:0 | R/W | 0x0 | CLK_DIV_RATIO_N. <br> Clock Pre Divide Ratio (N) <br> The select clock source is pre-divided by( n+1). The divider N is from 1 to 8. |

### 3.3.5.50. GPU Memory Register (Default Value: 0x00000000)

| Offset: 0x01A4 | | | Register Name: **GPU__MEM_CLK_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x0 | SCLK_GATING. <br> 0: Clock is OFF <br> 1: Clock is ON. <br> SCLK= Clock Source/Divider N. |
| 30:25 | / | / | /. |
| 24 | R/W | 0x0 | CLK_SRC_SEL <br> 0: PLL_GPU <br> 1: PLL_PERIPH. |
| 23:3 | / | / | / |
| 2:0 | R/W | 0x0 | CLK_DIV_RATIO_N. <br> Clock Pre Divide Ratio (N) <br> The select clock source is pre-divided by( n+1). The divider N is from 1 to 8. |

### 3.3.5.51. GPU HYD Clock Register (Default Value: 0x00000000)

| Offset: 0x01A8 | | | Register Name: **GPU_HYD_CLK_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x0 | SCLK_GATING. <br> 0: Clock is OFF <br> 1: Clock is ON. <br> SCLK= PLL_GPU/Divider N. |
| 30:3 | / | / | /. |
| 2:0 | R/W | 0x0 | CLK_DIV_RATIO_N. <br> Clock Pre Divide Ratio (N) <br> The select clock source is pre-divided by( n+1). The divider N is from 1 to 8. |

### 3.3.5.52. PLL Stable Time Register0 (Default Value: 0x000000FF)

| Offset: 0x0200 | | | Register Name: **PLL_STABLE_TIME_REG0** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:16 | / | / | / |
| 15:0 | R/W | 0x00FF | PLL_LOCK_TIME <br> PLL Lock Time (Unit: us). <br> Note: When any PLL (except PLL_CPUX) is enabled or changed, the corresponding PLL lock bit will be set after the PLL STABLE Time. |

### 3.3.5.53. PLL Stable Time Register1 (Default Value: 0x000000FF)

| Offset: 0x0204 | | | Register Name: **PLL_STABLE_TIME_REG1** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:16 | / | / | / |
| 15:0 | R/W | 0x00FF | PLL_CPUX_LOCK_TIME <br> PLL_CPUX Lock Time (Unit: us). <br> Note: When PLL_CPUX is enabled or changed, the PLL_CPUX lock bit will be set after the PLL_CPUX STABLE Time. |

### 3.3.5.54. PLL Stable Status Register (Default Value: 0x00000000)

| Offset: 0x020C | | | Register Name: **PLL_STABLE_STATUS_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:11 | / | / | / |
| 10 | R | 0x0 | PLL_VIDEO1_STB <br> 0: Unstable |

| | | | 1: Stable (It indicates that the PLL_VIDEO1 has been stable.) |
|---|---|---|---|
| 9 | R | 0x0 | PLL_DE_STB<br>0: Unstable<br>1: Stable (It indicates that the PLL_DE has been stable.) |
| 8 | R | 0x0 | PLL_HSIC_STB<br>0: Unstable<br>1: Stable (It indicates that the PLL_HSIC has been stable.) |
| 7 | R | 0x0 | PLL_GPU_STB<br>0: Unstable<br>1: Stable (It indicates that the PLL_GPU has been stable.) |
| 6 | R | 0x0 | PLL_PERIPH_STB<br>0: Unstable<br>1: Stable (It indicates that the PLL_PERIPH0 has been stable.) |
| 5 | R | 0x0 | PLL_DDR_STB<br>0: Unstable<br>1: Stable (It indicates that the PLL_DDR has been stable.) |
| 4 | R | 0x0 | PLL_VE_STB<br>0: Unstable<br>1: Stable (It indicates that the PLL_VE has been stable.) |
| 3 | R | 0x0 | PLL_VIDEO0_STB<br>0: Unstable<br>1: Stable (It indicates that the PLL_VIDEO0 has been stable.) |
| 2 | R | 0x0 | PLL_ADUIO_STB<br>0: Unstable<br>1: Stable (It indicates that the PLL_AUDIO has been stable.) |
| 1 | R | 0x0 | PLL_C1CPUX_STB<br>0: Unstable<br>1: Stable (It indicates that the PLL_C1CPUX has been stable.) |
| 0 | R | 0x0 | PLL_C0CPUX_STB<br>0: Unstable<br>1: Stable (It indicates that the PLL_C0CPUX has been stable.) |

### 3.3.5.55. PLL_C0CPUX Bias Register (Default Value: 0x08040000)

| Offset: 0x0220 | | | Register Name: **PLL_C0CPUX_BIAS_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x1 | VCO_RST.<br>VCO Reset In. |
| 30:21 | / | / | / |
| 20:16 | R/W | 0x04 | PLL_BIAS_CUR.<br>PLL Current Bias Control [4:0] ,cpu_cp |
| 15:0 | / | / | / |

### 3.3.5.56. PLL_AUDIO Bias Register (Default Value: 0x00040000)

| Offset: 0x0224 | | | Register Name: **PLL_AUDIO_BIAS_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:21 | / | / | / |
| 20:16 | R/W | 0x04 | PLL_BIAS_CTRL. PLL Current Bias Control[4:0]._cp |
| 15:0 | / | / | / |

### 3.3.5.57. PLL_VIDEO0 Bias Register (Default Value: 0x00040000)

| Offset: 0x0228 | | | Register Name: **PLL_VIDEO0_BIAS_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:21 | / | / | / |
| 20:16 | R/W | 0x04 | PLL_BIAS_CTRL. PLL Current Bias Control[4:0]._cp |
| 15:0 | / | / | / |

### 3.3.5.58. PLL_VE Bias Register (Default Value: 0x00040000)

| Offset: 0x022C | | | Register Name: **PLL_VE_BIAS_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:21 | / | / | / |
| 20:16 | R/W | 0x04 | PLL_BIAS_CTRL. PLL Current Bias Control[4:0]._cp |
| 15:0 | / | / | / |

### 3.3.5.59. PLL_DDR Bias Register (Default Value: 0x00040000)

| Offset: 0x0230 | | | Register Name: **PLL_DDR_BIAS_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:21 | / | / | / |
| 20:16 | R/W | 0x04 | PLL_BIAS_CTRL. PLL Current Bias Control[4:0]._cp |
| 15:0 | / | / | / |

### 3.3.5.60. PLL_PERIPH Bias Register (Default Value: 0x00040000)

| Offset: 0x0234 | Register Name: **PLL_PERIPH_BIAS_REG** |
|---|---|

| Bit | R/W | Default/Hex | Description |
|---|---|---|---|
| 31:21 | / | / | / |
| 20:16 | R/W | 0x04 | PLL_BIAS_CTRL.<br>PLL Current Bias Control[4:0]._cp |
| 15:0 | / | / | / |

### 3.3.5.61.  PLL_C1CPUX Bias Register (Default Value: 0x80040000)

| Offset: 0x0238 | | | Register Name: **PLL_C1CPUX_BIAS_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x1 | VCO_RST.<br>VCO Reset In. |
| 30:21 | / | / | / |
| 20:16 | R/W | 0x04 | PLL_BIAS_CUR.<br>PLL Current Bias Control [4:0] ,cpu_cp |
| 15:0 | / | / | / |

### 3.3.5.62.  PLL_GPU Bias Register (Default Value: 0x00040000)

| Offset: 0x023C | | | Register Name: **PLL_GPU_BIAS_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:21 | / | / | / |
| 20:16 | R/W | 0x04 | PLL_BIAS_CTRL.<br>PLL Current Bias Control[4:0]._cp |
| 15:0 | / | / | / |

### 3.3.5.63.  PLL_HSIC Bias Register (Default Value: 0x00040000)

| Offset: 0x0244 | | | Register Name: **PLL_HSIC_BIAS_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:21 | / | / | / |
| 20:16 | R/W | 0x04 | PLL_BIAS_CTRL.<br>PLL Current Bias Control[4:0]._cp |
| 15:0 | / | / | / |

### 3.3.5.64.  PLL_DE Bias Register (Default Value: 0x00040000)

| Offset: 0x0248 | | | Register Name: **PLL_DE_BIAS_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |

| 31:21 | / | / | / |
|-------|---|---|---|
| 20:16 | R/W | 0x04 | PLL_BIAS_CTRL.<br>PLL Current Bias Control[4:0]._cp |
| 15:0 | / | / | / |

### 3.3.5.65. PLL_VIDEO1 Bias Register (Default Value: 0x00040000)

| Offset: 0x024C | | | Register Name: **PLL_VIDEO1_BIAS_REG** |
|-------|-----|------------|-------------|
| Bit | R/W | Default/Hex | Description |
| 31:21 | / | / | / |
| 20:16 | R/W | 0x04 | PLL_BIAS_CTRL.<br>PLL Current Bias Control[4:0]._cp |
| 15:0 | / | / | / |

### 3.3.5.66. PLL_C0CPUX Tuning Register (Default Value: 0x44404000)

| Offset: 0x0250 | | | Register Name: **PLL_C0CPUX_TUN_REG** |
|-------|-----|------------|-------------|
| Bit | R/W | Default/Hex | Description |
| 31 | / | / | / |
| 30:28 | R/W | 0x4 | VCO_RNG_CTRL.<br>VCO Range Control [2:0]. |
| 27 | / | / | / |
| 26:24 | R/W | 0x4 | VCO_GAIN_CTRL.<br>KVCO Gain Control Bits[2:0]. |
| 23 | / | / | / |
| 22:16 | R/W | 0x40 | CNT_INT_CTRL.<br>Counter Initial Control[6:0]. |
| 15 | R/W | 0x0 | C_OD.<br>C-Reg-Od for verify |
| 14:8 | R/W | 0x40 | C_B_IN.<br>C-B-In[6:0] for verify. |
| 7 | R/W | 0x0 | C_OD1.<br>C-Reg-Od1 for verify |
| 6:0 | RO | 0x0 | C_B_OUT.<br>For verify |

### 3.3.5.67. PLL_C1CPUX Tuning Register (Default Value: 0x44404000)

| Offset: 0x0254 | | | Register Name: **PLL_C1CPUX_TUN_REG** |
|-------|-----|------------|-------------|
| Bit | R/W | Default/Hex | Description |
| 31 | / | / | / |

| 30:28 | R/W | 0x4 | VCO_RNG_CTRL. |
| | | | VCO Range Control [2:0]. |
| 27 | / | / | / |
| 26:24 | R/W | 0x4 | VCO_GAIN_CTRL. |
| | | | KVCO Gain Control Bits[2:0]. |
| 23 | / | / | / |
| 22:16 | R/W | 0x40 | CNT_INT_CTRL. |
| | | | Counter Initial Control[6:0]. |
| 15 | R/W | 0x0 | C_OD. |
| | | | C-Reg-Od for verify |
| 14:8 | R/W | 0x40 | C_B_IN. |
| | | | C-B-In[6:0] for verify. |
| 7 | R/W | 0x0 | C_OD1. |
| | | | C-Reg-Od1 for verify |
| 6:0 | RO | 0x0 | C_B_OUT. |
| | | | For verify |

### 3.3.5.68. PLL_AUDIO Pattern Control Register0 (Default Value: 0x00000000)

| Offset: 0x0284 | | | Register Name: **PLL_AUDIO_PAT_CTRL_REG0** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x0 | SIG_DELT_PAT_EN. |
| | | | Sigma-delta Pattern Enable. |
| 30:29 | R/W | 0x0 | SPR_FREQ_MODE. |
| | | | Spread Frequency Mode. |
| | | | 00: DC=0 |
| | | | 01: DC=1 |
| | | | 10: Triangular(1 bit pattern) |
| | | | 11: / |
| 28:20 | R/W | 0x0 | WAVE_STEP. |
| | | | Wave Step. |
| 19 | R/W | 0x0 | CLK_SRC_SEL |
| | | | 0: 24MHz |
| | | | 1: 12MHz |
| 18:17 | R/W | 0x0 | FREQ. |
| | | | Frequency. |
| | | | 00: 31.5KHz |
| | | | 01: 32KHz |
| | | | 10: 32.5KHz |
| | | | 11: 33KHz |
| 16:0 | R/W | 0x0 | WAVE_BOT. |
| | | | Wave Bottom. |

### 3.3.5.69. PLL_VIDEO0 Pattern Control Register 0 (Default Value: 0x00000000)

| Offset: 0x0288 | | | Register Name: **PLL_VIDEO0_PAT_CTRL_REG0** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x0 | SIG_DELT_PAT_EN.<br>Sigma-delta Pattern Enable. |
| 30:29 | R/W | 0x0 | SPR_FREQ_MODE.<br>Spread Frequency Mode.<br>00: DC=0<br>01: DC=1<br>10: Triangular(1 bit pattern)<br>11: / |
| 28:20 | R/W | 0x0 | WAVE_STEP.<br>Wave Step. |
| 19 | R/W | 0x0 | CLK_SRC_SEL<br>0: 24MHz<br>1: 12MHz |
| 18:17 | R/W | 0x0 | FREQ.<br>Frequency.<br>00: 31.5KHz<br>01: 32KHz<br>10: 32.5KHz<br>11: 33KHz |
| 16:0 | R/W | 0x0 | WAVE_BOT.<br>Wave Bottom. |

### 3.3.5.70. PLL_DDR Pattern Control Register 0 (Default Value: 0x00000000)

| Offset: 0x0290 | | | Register Name: **PLL_DDR_PAT_CTRL_REG0** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x0 | SIG_DELT_PAT_EN.<br>Sigma-delta Pattern Enable. |
| 30:29 | R/W | 0x0 | SPR_FREQ_MODE.<br>Spread Frequency Mode.<br>00: DC=0<br>01: DC=1<br>10: /<br>11: Triangular(3 bit pattern) |
| 28:20 | R/W | 0x0 | WAVE_STEP.<br>Wave step. |
| 19 | R/W | 0x0 | CLK_SRC_SEL<br>0: 24MHz<br>1: 12MHz |

| 18:17 | R/W | 0x0 | FREQ. |
| | | | Frequency. |
| | | | 00: 31.5KHz |
| | | | 01: 32KHz |
| | | | 10: 32.5KHz |
| | | | 11: 33KHz |
| 16:0 | R/W | 0x0 | WAVE_BOT. |
| | | | Wave Bottom. |

### 3.3.5.71. PLL_AUDIO Pattern Control Register 1 (Default Value: 0x00000000)

| Offset: 0x02A4 | | | Register Name: **PLL_AUDIO_PAT_CTRL_REG1** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:25 | / | / | / |
| 24 | R/W | 0x0 | DITHER_EN. |
| 23:21 | / | / | / |
| 20 | R/W | 0x0 | FRAC_EN. |
| 19:17 | / | / | / |
| 16:0 | R/W | 0x0 | FRAC_IN. |

### 3.3.5.72. PLL_VIDEO0 Pattern Control Register 1 (Default Value: 0x00000000)

| Offset: 0x02A8 | | | Register Name: **PLL_VIDEO0_PAT_CTRL_REG1** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:25 | / | / | / |
| 24 | R/W | 0x0 | DITHER_EN. |
| 23:21 | / | / | / |
| 20 | R/W | 0x0 | FRAC_EN. |
| 19:17 | / | / | / |
| 16:0 | R/W | 0x0 | FRAC_IN. |

### 3.3.5.73. PLL_DDR Pattern Control Register 1 (Default Value: 0x00000000)

| Offset: 0x02B0 | | | Register Name: **PLL_DDR_PAT_CTRL_REG1** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:25 | / | / | / |
| 24 | R/W | 0x0 | DITHER_EN. |
| 23:21 | / | / | / |
| 20 | R/W | 0x0 | FRAC_EN. |
| 19:17 | / | / | / |

| 16:0 | R/W | 0x0 | FRAC_IN. |

### 3.3.5.74. Bus Software Reset Register 0 (Default Value: 0x00000000)

| Offset: 0x02C0 | | | Register Name: **BUS_SOFT_RST_REG0** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:30 | / | / | / |
| 29 | R/W | 0x0 | USB_OHCI0_RST.<br>USB OHCI0 Reset<br>0: Assert<br>1: De-assert. |
| 28 | / | / | / |
| 27 | R/W | 0x0 | USB_EHCI1_RST.<br>USB EHCI1 Reset<br>0: Assert<br>1: De-assert. |
| 26 | R/W | 0x0 | USB_EHCI0_RST.<br>USB EHCI0 Reset<br>0: Assert<br>1: De-assert. |
| 25 | / | / | / |
| 24 | R/W | 0x0 | USB_DRD_RST.<br>USB DRD Reset<br>0: Assert<br>1: De-assert. |
| 23:22 | / | / | / |
| 21 | R/W | 0x0 | SPI1_RST.<br>SPI1 Reset.<br>0: Assert<br>1: De-assert. |
| 20 | R/W | 0x0 | SPI0_RST.<br>SPI0 Reset.<br>0: Assert<br>1: De-assert. |
| 19 | R/W | 0x0 | HSTIMR_RST.<br>HSTIMR Reset.<br>0: Assert<br>1: De-assert. |
| 18 | / | / | / |
| 17 | R/W | 0x0 | EMAC_RST.<br>EMAC Reset.<br>0: Assert<br>1: De-assert. |
| 16:15 | / | / | / |

| 14 | R/W | 0x0 | DRAM_RST. |
| | | | DRAM AHB Reset. |
| | | | 0: Assert |
| | | | 1: De-assert. |
| 13 | R/W | 0x0 | NAND_RST. |
| | | | NAND Reset. |
| | | | 0: Assert |
| | | | 1: De-assert. |
| 12:11 | / | / | / |
| 10 | R/W | 0x0 | MMC2_RST. |
| | | | MMC2 Reset. |
| | | | 0: Assert |
| | | | 1: De-assert. |
| 9 | R/W | 0x0 | MMC1_RST. |
| | | | MMC1 Reset. |
| | | | 0: Assert |
| | | | 1: De-assert. |
| 8 | R/W | 0x0 | MMC0_RST. |
| | | | MMC0 Reset. |
| | | | 0: Assert |
| | | | 1: De-assert. |
| 7 | / | / | / |
| 6 | R/W | 0x0 | DMA_RST. |
| | | | DMA Reset. |
| | | | 0: Assert |
| | | | 1: De-assert. |
| 5 | R/W | 0x0 | SS_RST. |
| | | | SS Reset. |
| | | | 0: Assert |
| | | | 1:De-assert. |
| 4:2 | / | / | / |
| 1 | R/W | 0x0 | MIPI_DSI_RST. |
| | | | MIPI DSI Reset. |
| | | | 0: Assert |
| | | | 1: De-assert. |
| 0 | / | / | / |

### 3.3.5.75.  Bus Software Reset Register 1 (Default Value: 0x00000000)

| Offset: 0x02C4 | | | Register Name: **BUS_SOFT_RST_REG1** |
| --- | --- | --- | --- |
| Bit | R/W | Default/Hex | Description |
| 31:23 | / | / | / |
| 22 | R/W | 0x0 | SPINLOCK_RST. |
| | | | SPINLOCK Reset. |

| | | | |
|---|---|---|---|
| | | | 0: Assert |
| | | | 1: De-assert. |
| 21 | R/W | 0x0 | MSGBOX_RST. |
| | | | MSGBOX Reset. |
| | | | 0: Assert |
| | | | 1: De-assert. |
| 20 | R/W | 0x0 | GPU_RST. |
| | | | GPU Reset. |
| | | | 0: Assert |
| | | | 1: De-assert. |
| 19:13 | / | / | / |
| 12 | R/W | 0x0 | DE_RST. |
| | | | DE Reset. |
| | | | 0: Assert |
| | | | 1: De-assert. |
| 11 | R/W | 0x0 | HDMI1_RST. |
| | | | HDMI1 Reset. |
| | | | 0: Assert |
| | | | 1: De-assert. |
| 10 | R/W | 0x0 | HDMI0_RST. |
| | | | HDMI0 Reset. |
| | | | 0: Assert |
| | | | 1: De-assert. |
| 9 | / | / | / |
| 8 | R/W | 0x0 | CSI_RST. |
| | | | CSI Reset. |
| | | | 0: Assert |
| | | | 1: De-assert. |
| 7:6 | / | / | |
| 5 | R/W | 0x0 | TCON1_RST. |
| | | | TCON1 Reset. |
| | | | 0: Assert |
| | | | 1: De-assert. |
| 4 | R/W | 0x0 | TCON0_RST. |
| | | | TCON0 Reset. |
| | | | 0: Assert |
| | | | 1: De-assert. |
| 3:1 | / | / | / |
| 0 | R/W | 0x0 | VE_RST. |
| | | | VE Reset. |
| | | | 0: Assert |
| | | | 1: De-assert. |

### 3.3.5.76. Bus Software Reset Register 2 (Default Value: 0x00000000)

| Offset: 0x02C8 | | | Register Name: **BUS_SOFT_RST_REG2** |
| --- | --- | --- | --- |
| Bit | R/W | Default/Hex | Description |
| 31:1 | / | / | / |
| 0 | R/W | 0x0 | LVDS_RST.<br>LVDS Reset.<br>0: Assert<br>1: De-assert. |

### 3.3.5.77. Bus Software Reset Register 3 (Default Value: 0x00000000)

| Offset: 0x02D0 | | | Register Name: **BUS_SOFT_RST_REG3** |
| --- | --- | --- | --- |
| Bit | R/W | Default/Hex | Description |
| 31:16 | / | / | / |
| 15 | R/W | 0x0 | TDM_RST.<br>TDM Reset.<br>0: Assert<br>1: De-assert. |
| 14 | R/W | 0x0 | DAUDIO2_RST.<br>DAUDIO 2 Reset.<br>0: Assert<br>1: De-assert. |
| 13 | R/W | 0x0 | DAUDIO1_RST.<br>DAUDIO 1 Reset.<br>0: Assert<br>1: De-assert. |
| 12 | R/W | 0x0 | DAUDIO0_RST.<br>DAUDIO 0 Reset.<br>0: Assert<br>1: De-assert. |
| 11:2 | / | / | / |
| 1 | R/W | 0x0 | OWA_RST.<br>OWA Reset.<br>0: Assert<br>1: De-assert. |
| 0 | / | / | / |

### 3.3.5.78. Bus Software Reset Register 4 (Default Value: 0x00000000)

| Offset: 0x02D8 | | | Register Name: **BUS_SOFT_RST_REG4** |
| --- | --- | --- | --- |
| Bit | R/W | Default/Hex | Description |

| 31:21 | / | / | / |
|-------|-----|-----|-----|
| 20 | R/W | 0x0 | UART4_RST.<br>UART4 Reset.<br>0: Assert<br>1: De-assert. |
| 19 | R/W | 0x0 | UART3_RST.<br>UART3 Reset.<br>0: Assert<br>1: De-assert. |
| 18 | R/W | 0x0 | UART2_RST.<br>UART2 Reset.<br>0: Assert<br>1: De-assert. |
| 17 | R/W | 0x0 | UART1_RST.<br>UART1 Reset.<br>0: Assert<br>1: De-assert. |
| 16 | R/W | 0x0 | UART0_RST.<br>UART0 Reset.<br>0: Assert<br>1: De-assert. |
| 15:3 | / | / | / |
| 2 | R/W | 0x0 | TWI2_RST.<br>TWI2 Reset.<br>0: Assert<br>1: De-assert. |
| 1 | R/W | 0x0 | TWI1_RST.<br>TWI1 Reset.<br>0: Assert<br>1: De-assert. |
| 0 | R/W | 0x0 | TWI0_RST.<br>TWI0 Reset.<br>0: Assert<br>1: De-assert. |

## 3.3.6.    Programming Guidelines

### 3.3.6.1.    PLL

1)    In practical application, other PLLs doesn't support dynamic frequency scaling except for PLL_CPUX and PLL_DDR;
2)    After the PLL_DDR frequency changes, the 30-bit of PLL_DDR Control Register should be written 1 to make it valid;

### 3.3.6.2.    BUS

1)    When setting the BUS clock , you should set the division factor first, and after the division factor becomes valid, switch the clock source. The clock source will be switched after at least three clock cycles;
2)    The BUS clock should not be dynamically changed in most applications.

### 3.3.6.3.    Clock Switch

Make sure that the clock source output is valid before the clock source switch, and then set a proper divide ratio; after the division factor becomes valid, switch the clock source.

### 3.3.6.4.    Gating and reset

Make sure that the reset signal has been released before the release of module clock gating;

## 3.4. CPU Configuration

### 3.4.1. Overview

CPUCFG module is used to configure related CPU control of the two clusters, including power, reset, cache, debug, CPU status, etc; It will be used when you want to disable/enable the CPU, cluster switch, CPU status check, and debug, etc.

It features:

- Capable of CPU reset, including core reset, debug circuit rest, etc
- Capable of Cache control, including cache reset, idle control, etc
- Capable of other CPU-related control, including interface control, CP15 control, and power control, etc
- Capable of checking CPU status, including idle status, SMP status, and interrupt status, etc

### 3.4.2. Functionalities Description

#### 3.4.2.1. L2 Idle Mode

When the L2 of Cluster needs to enter WFI mode, firstly make sure the CPU0/1/2/3 of Cluster all enter WFI mode, which can be checked in Cluster CPU status Register, and then pull the ACINACTM of Cluster high by writing related register bit to 1, and then check whether L2 enters idle status by checking whether the STANDBYWFIL2 is high. Remember to set the ACINACTM to low when exiting the L2 idle mode;

#### 3.4.2.2. CPU Reset System

The CPU reset includes core reset, power on reset and H_reset. And their scopes rank: Core Reset < Power-On Reset < H_Reset.

### 3.4.3. Block Diagram
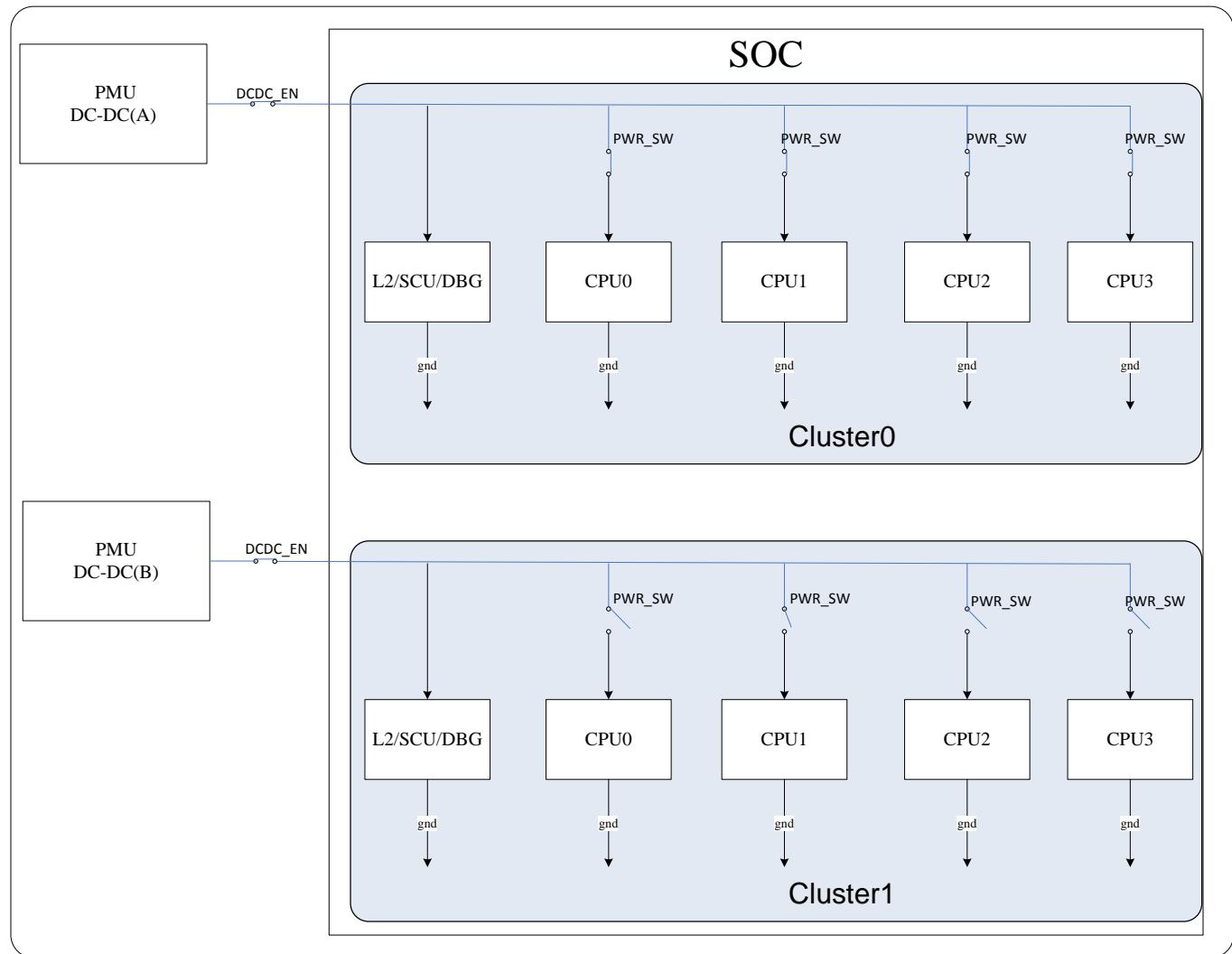


Figure 3-3. CPU Power Domain Diagram(Default State)

The figure above lists the CPU reset power domain. Since each CPU and its appended circuits have the same power domain, the processor and related L1 cache, neon, and vfp should be taken as a whole when it comes to the CPU core enable/disable.

### 3.4.4. Operation Principle

CPU-related operation needs proper configuration of CPUCFG related register, as well as related system control resource including BUS, clock ,reset and power control;

## 3.4.5. Register List

| Module Name | Base Address |
|---|---|
| CPUCFG | 0x01700000 |

| Register Name | Offset | Description |
|---|---|---|
| C0_CTRL_REG0 | 0x0000 | Cluster 0 Control Register0 |
| C0_CTRL_REG1 | 0x0004 | Cluster 0 Control Register1 |
| C0_ADB400_PWRDNREQN_REG | 0x0008 | Cluster 0 adb400 pwrdnreqn Register |
| C1_CTRL_REG0 | 0x0010 | Cluster 1 Control Register0 |
| C1_CTRL_REG1 | 0x0014 | Cluster 1 Control Register1 |
| C1_ADB400_PWRDNREQN_REG | 0x0018 | Cluster 1 adb400 pwrdnreqn Register |
| GENER_CTRL_REG0 | 0x0028 | General Control Register0 |
| GENER_CTRL_REG1 | 0x002C | General Control Register1 |
| C0_CPU_STATUS | 0x0030 | Cluster0 CPU Status Register |
| C1_CPU_STATUS | 0x0034 | Cluster1 CPU Status Register |
| IRQ_FIQ_ STATUS | 0x003C | Cluster CPU Irq and Fiq Status Register |
| IRQ_FIQ_MASK | 0x0040 | Cluster CPU Irq and Fiq Mask Register to CPUS |
| C0_RST_CTRL | 0x0080 | Cluster 0 Reset Control Register |
| C1_RST_CTRL | 0x0084 | Cluster 1 Reset Control Register |
| GIC_JTAG_RST_CTRL | 0x0088 | GIC and Jtag reset control Register |

## 3.4.6. Register Description

### 3.4.6.1. Cluster 0 Control Register0(Default Value:0x60000000)

| Offset: 0x00 | | | Register Name: **C0_CTRL_REG0** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x0 | SYSBAR_DISABLE. Disable broadcasting of barriers onto system bus: 0:Barriers are broadcast onto system bus,this requires an AMBA4 interconnect. 1: Barriers are not broadcast onto the system bus.This is compatible with an AXI3 interconnect. |
| 30 | R/W | 0x1 | BROADCAST_INNER. Enable broadcasting of Inner Shareable transactions: 0: Inner shareable transactions are not broadcasted externally. 1: Inner shareable transactions are broadcasted externally. |

| 29 | R/W | 0x1 | BROADCAST_OUTER.<br>Enable broadcasting of outer shareable transactions:<br>0: Outer Shareable transactions are not broadcasted externally.<br>0: Outer Shareable transactions are broadcasted externally. |
|---|---|---|---|
| 28 | R/W | 0x0 | BROADCAST_CACHE_MAINT<br>Enable broadcasting of cache maintenance operations to downstream caches:<br>0: Cache maintenance operations are not broadcasted to downstream caches.<br>1: Cache maintenance operations are broadcasted to downstream caches. |
| 27:12 | / | / | / |
| 11:8 | R/W | 0x0 | CP15S_DISABLE.<br>Disable write access to some secure CP15 register. |
| 7:6 | / | / | / |
| 5 | R/W | 0x0 | CFGS_DISABLE.<br>Disable write access to some secure GIC register. |
| 4 | R/W | 0x0 | L2_RST_DISABLE.<br>Disable automatic L2 cache invalidate at reset:<br>0: L2 cache is reset by hardware.<br>1: L2 cache is not reset by hardware. |
| 3:0 | R/W | 0x0 | L1_RST_DISABLE.<br>Disable automatic Cluster0 CPU0/1/2/3 L1 cache invalidate at reset:<br>0: L1 cache is reset by hardware.<br>1: L1 cache is not reset by hardware. |

### 3.4.6.2. Cluster 0 Control Register1(Default Value:0x00000000)

| Offset: 0x04 | | | Register Name: **C0_CTRL_REG1** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:24 | / | / | / |
| 23:20 | R/W | 0x0 | SLEEP.<br>Cluster 0 CPU L1 cache Sleep Mode Control<br>0: Normal<br>1: Sleeping. |
| 19:17 | / | / | / |
| 16 | R/W | 0x0 | SLEEP_L2.<br>Cluster 0 L2 Sleep Mode Control<br>0: Normal<br>1: Sleeping. |
| 15:12 | R/W | 0x0 | SHUT_DOWN.<br>Cluster 0 CPU L1 cache Shut Down Control (The data in cache will be lost)<br>0:Normal<br>1:Shut Down. |
| 11:9 | / | / | / |

| 8 | R/W | 0x0 | SHUT_DOWN_L2. <br> Cluster 0 L2 cache Shut Down Control (The data in cache will be lost) <br> 0:Normal <br> 1:Shut Down. |
|---|-----|-----|------|
| 7:1 | / | / | / |
| 0 | R/W | 0x0 | ACINACTM. <br> Snoop interface is inactive and no longer accepting requests. |

### 3.4.6.3. Cluster 0 adb400 pwrdnreqn Register(Default Value:0x00000001)

| Offset: 0x08 | | | Register Name: **C0_ADB400_PWRDNREQN_REG** |
|--------------|-----|-------------|------------------------------------------|
| Bit | R/W | Default/Hex | Description |
| 31:1 | / | / | / |
| 0 | R/W | 0x1 | ADB400_PWRDNREQN. |

### 3.4.6.4. Cluster 1 Control Register0(Default Value:0x60000000)

| Offset: 0x10 | | | Register Name: **C1_CTRL_REG0** |
|--------------|-----|-------------|------------------------------------------|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x0 | SYSBAR_DISABLE. <br> Disable broadcasting of barriers onto system bus: <br> 0: Barriers are broadcast onto system bus,this requires an AMBA4 interconnect. <br> 1: Barriers are not broadcast onto the system bus.This is compatible with an AXI3 interconnect. |
| 30 | R/W | 0x1 | BROADCAST_INNER. <br> Enable broadcasting of Inner Shareable transactions: <br> 0: Inner shareable transactions are not broadcasted externally. <br> 1: Inner shareable transactions are broadcasted externally. |
| 29 | R/W | 0x1 | BROADCAST_OUTER. <br> Enable broadcasting of outer shareable transactions: <br> 0: Outer Shareable transactions are not broadcasted externally. <br> 0: Outer Shareable transactions are broadcasted externally. |
| 28 | R/W | 0x0 | BROADCAST_CACHE_MAINT <br> Enable broadcasting of cache maintenance operations to downstream caches: <br> 0: Cache maintenance operations are not broadcasted to downstream caches. <br> 1: Cache maintenance operations are broadcasted to downstream caches. |
| 27:12 | / | / | / |
| 11:8 | R/W | 0x0 | CP15S_DISABLE. <br> Disable write access to some secure CP15 register. |

| 7:6 | / | / | / |
|---|---|---|---|
| 5 | R/W | 0x0 | CFGS_DISABLE.<br>Disable write access to some secure GIC register. |
| 4 | R/W | 0x0 | L2_RST_DISABLE.<br>Disable automatic L2 cache invalidate at reset:<br>0: L2 cache is reset by hardware.<br>1: L2 cache is not reset by hardware. |
| 3:0 | R/W | 0x0 | L1_RST_DISABLE.<br>Disable automatic Cluster1 CPU0/1/2/3 L1 cache invalidate at reset:<br>0: L1 cache is reset by hardware.<br>1: L1 cache is not reset by hardware. |

### 3.4.6.5. Cluster 1 Control Register1(Default Value:0x00000000)

| Offset: 0x14 | | | Register Name: **C1_CTRL_REG1** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:24 | / | / | / |
| 23:20 | R/W | 0x0 | SLEEP.<br>Cluster 1 CPU L1 cache Sleep Mode Control<br>0: Normal<br>1: Sleeping. |
| 19:17 | / | / | / |
| 16 | R/W | 0x0 | SLEEP_L2.<br>Cluster 1 L2 Sleep Mode Control<br>0: Normal<br>1: Sleeping. |
| 15:12 | R/W | 0x0 | SHUT_DOWN.<br>Cluster 1 CPU L1 cache Shut Down Control (The data in cache will be lost)<br>0:Normal<br>1:Shut Down. |
| 11:9 | / | / | / |
| 8 | R/W | 0x0 | SHUT_DOWN_L2.<br>Cluster 1 L2 cache Shut Down Control (The data in cache will be lost)<br>0:Normal<br>1:Shut Down. |
| 7:1 | / | / | / |
| 0 | R/W | 0x0 | ACINACTM.<br>Snoop interface is inactive and no longer accepting requests. |

### 3.4.6.6. Cluster 1 adb400 pwrdnreqn Register(Default Value:0x00000001)

| Offset: 0x18 | Register Name: **C1_ADB400_PWRDNREQN_REG** |
|---|---|

| Bit | R/W | Default/Hex | Description |
|------|------|------|------|
| 31:1 | / | / | / |
| 0 | R/W | 0x1 | ADB400_PWRDNREQN. |

### 3.4.6.7.  Debug Control Register0(Default Value:0x00000F0F)

| Offset: 0x20 | | | Register Name: **DBG_REG0** |
|------|------|------|------|
| Bit | R/W | Default/Hex | Description |
| 31:20 | / | / | / |
| 19:12 | / | / | / |
| 11:8 | R/W | 0xF | C1_DBGPWRDUP. Cluster Powered-up |
| 7:4 | / | / | / |
| 3:0 | R/W | 0xF | C0_DBGPWRDUP. Cluster Powered-up |

### 3.4.6.8.  General Control Register0(Default Value:0x00000000)

| Offset: 0x28 | | | Register Name: **GENER_CTRL_REG0** |
|------|------|------|------|
| Bit | R/W | Default/Hex | Description |
| 31:1 | / | / | / |
| 0 | R/W | 0x0 | GIC_CFGSDISABLE. Disables write access to some secure GIC registers. |

### 3.4.6.9.  General Control Register1(Default Value:0x0000000E)

| Offset: 0x2C | | | Register Name: **GENER_CTRL_REG1** |
|------|------|------|------|
| Bit | R/W | Default/Hex | Description |
| 31:6 | / | / | / |
| 5:4 | R/W | 0x0 | QOSOVERRIDE. If HIGH, internally generated values override the ARQOS and AWQOS inputs. One bit exists for each slave interface. |
| 3:2 | R/W | 0x3 | ACCHANNELEN. If LOW, then AC requests are never issued on the corresponding slave interface. One bit exists for each slave interface. |
| 1 | R/W | 0x1 | CCI_CLK_ON. Set 1 means CCI Clock always on. |
| 0 | R/W | 0x0 | CCI_CLK_OFF. Set 1 means CCI Clock always off. Note: When CCI_CLK_OFF and CCI_CLK_ON are active simultaneously，the |

| | | | priority of CCI_CLK_OFF is higher than CCI_CLK_ON. |
|---|---|---|---|

### 3.4.6.10. Cluster0 CPU Status Register(Default Value: 0x00000000)

| Offset: 0x30 | | | Register Name: **C0_CPU_STATUS** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:28 | / | / | / |
| 27:24 | R | 0x0 | SMP_AMP<br>A CPU is in Symmetric Multiprocessing mode or Asymmetric Multiprocessing mode.<br>0: AMP mode<br>1: SMP mode |
| 23:20 | / | / | / |
| 19:16 | R | 0x0 | STANDBYWFI.<br>Indicates if Cluster0 CPU0/1/2/3 is in WFI standby mode:<br>0: Processor not in WFI standby mode.<br>1: Processor in WFI standby mode |
| 15:12 | / | / | / |
| 11:8 | R | 0x0 | STANDBYWFE.<br>Indicates if Cluster0 CPU0/1/2/3 is in the WFE standby mode:<br>0: Processor not in WFE standby mode<br>1: Processor in WFE standby mode |
| 7:1 | / | / | / |
| 0 | R | 0x0 | STANDBYWFIL2.<br>Indicates if the Cluster0 L2 memory system is in WFI standby mode.<br>0:active<br>1:idle |

### 3.4.6.11. Cluster1 CPU Status Register(Default Value: 0x00000000)

| Offset: 0x34 | | | Register Name: **C1_CPU_STATUS** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:28 | / | / | / |
| 27:24 | R | 0x0 | SMP_AMP<br>A CPU is in Symmetric Multiprocessing mode or Asymmetric Multiprocessing mode.<br>0: AMP mode<br>1: SMP mode |
| 23:20 | / | / | / |
| 19:16 | R | 0x0 | STANDBYWFI.<br>Indicates if Cluster1 CPU0/1/2/3 is in WFI standby mode:<br>0: Processor not in WFI standby mode. |

| | | | |
|---|---|---|---|
| | | | 1: Processor in WFI standby mode |
| 15:12 | / | / | / |
| 11:8 | R | 0x0 | STANDBYWFE.<br>Indicates if Cluster1 CPU0/1/2/3 is in the WFE standby mode:<br>0: Processor not in WFE standby mode<br>1: Processor in WFE standby mode |
| 7:1 | / | / | / |
| 0 | R | 0x0 | STANDBYWFIL2.<br>Indicates if the Cluster1 L2 memory system is in WFI standby mode.<br>0:active<br>1:idle |

### 3.4.6.12. Cluster CPU IRQ and FIQ Status Register(Default Value: 0x001F0000)

| Offset: 0x3C | | | Register Name: **IRQ_FIQ_ STATUS** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:21 | / | / | / |
| 20:16 | R | 0x1F | EVENT_CNT_OF<br>Event counter over flag[4:0].The CCI-400 counters: The CCNT overflow is bit 4,event counters on bits[3:0].<br>0: Counter has overflowed<br>1: Counter has not overflowed |
| 15:12 | R | 0x0 | C1_FIQ_OUT[3:0].<br>Cluster1 CPU FIQ wakeup output<br>0: Normal<br>1: FIQ happen |
| 11:8 | R | 0x0 | C0_FIQ_OUT[3:0].<br>Cluster0 CPU FIQ wakeup output<br>0: Normal<br>1: FIQ happen |
| 7:4 | R | 0x0 | C1_IRQ_OUT[3:0].<br>Cluster1 CPU IRQ wakeup output<br>0: Normal<br>1: IRQ happen |
| 3:0 | R | 0x0 | C0_IRQ_OUT[3:0].<br>Cluster0 CPU IRQ wakeup output<br>0: Normal<br>1: IRQ happen |

### 3.4.6.13. Cluster CPU IRQ and FIQ Mask Register (Default Value: 0x00000000)

| Offset: 0x40 | Register Name: **IRQ_FIQ_MASK** |
|---|---|

| Bit | R/W | Default/Hex | Description |
|---|---|---|---|
| 31:16 | / | / | / |
| 15:12 | R/W | 0x0 | C1_FIQ_MASK[3:0].<br>Cluster1 CPU FIQ signals Mask to GIC_OUT<br>0: masks<br>1: unmasks. |
| 11:8 | R/W | 0x0 | C0_FIQ_MASK[3:0].<br>Cluster0 CPU FIQ signals Mask to GIC_OUT<br>0: masks<br>1: unmasks. |
| 7:4 | R/W | 0x0 | C1_IRQ_MASK[3:0].<br>Cluster1 CPU IRQ signals Mask to GIC_OUT<br>0: masks<br>1: unmasks. |
| 3:0 | R/W | 0x0 | C0_IRQ_MASK[3:0].<br>Cluster0 CPU IRQ signals Mask to GIC_OUT<br>0: masks<br>1: unmasks. |

### 3.4.6.14.  Cluster 0 Reset Control Register(Default Value: 0x01FF1101)

| Offset: 0x80 | | | Register Name: **C0_RST_CTRL** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:25 | / | / | / |
| 24 | R/W | 0x1 | SOC_DBG_RST.<br>Cluster0 SOC Debug Reset<br>0: assert<br>1: de-assert. |
| 23:20 | R/W | 0xF | ETM_RST.<br>Cluster0 ETM Reset Assert.<br>0: assert<br>1: de-assert. |
| 19:16 | R/W | 0xF | DBG_RST.<br>Cluster0 Debug Reset Assert.<br>0: assert<br>1: de-assert. |
| 15:13 | / | / | / |
| 12 | R/W | 0x1 | HRESET.<br>Cluster0 H_Reset. Reset all the Cluster0 Logic and Cluster0 Interface Logic.<br>0: assert<br>1: de-assert. |
| 11:9 | / | / | / |
| 8 | R/W | 0x1 | L2_RST.<br>Cluster0 L2 Cache Reset |

| | | | 0: assert |
| | | | 1: de-assert. |
| 7:4 | / | / | / |
| 3:0 | R/W | 0x1 | CORE_RESET. |
| | | | Cluster0 CPU0/1/2/3 Reset Assert. |
| | | | 0: assert |
| | | | 1: de-assert. |

### 3.4.6.15. Cluster 1 Reset Control Register(Default Value: 0x01FF1100)

| Offset: 0x84 | | | Register Name: **C1_RST_CTRL** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:25 | / | / | / |
| 24 | R/W | 0x1 | SOC_DBG_RST. |
| | | | Cluster1 SOC Debug Reset |
| | | | 0: assert |
| | | | 1: de-assert. |
| 23:20 | R/W | 0xF | ETM_RST. |
| | | | Cluster1 ETM Reset Assert. |
| | | | 0: assert |
| | | | 1: de-assert. |
| 19:16 | R/W | 0xF | DBG_RST. |
| | | | Cluster1 Debug Reset Assert. |
| | | | 0: assert |
| | | | 1: de-assert. |
| 15:13 | / | / | / |
| 12 | R/W | 0x1 | HRESET. |
| | | | Cluster1 H_Reset. Reset all the Cluster1 Logic and Cluster1 Interface Logic. |
| | | | 0: assert |
| | | | 1: de-assert. |
| 11:9 | / | / | / |
| 8 | R/W | 0x1 | L2_RST. |
| | | | Cluster1 L2 Cache Reset |
| | | | 0: assert |
| | | | 1: de-assert. |
| 7:4 | / | / | / |
| 3:0 | R/W | 0x0 | CORE_RESET. |
| | | | Cluster1 CPU0/1/2/3 Reset Assert. |
| | | | 0: assert |
| | | | 1: de-assert. |

### 3.4.6.16. GIC and Jtag Reset Control Register(Default Value: 0x00000F01)

| Offset: 0x88 | | | Register Name: **GIC_JTAG_RST_CTRL** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:12 | / | / | / |
| 11 | R/W | 0x1 | CS_RST<br>CoreSight Reset.<br>0: assert<br>1: de-assert. |
| 10 | R/W | 0x1 | DAP_RST<br>DAP Reset.<br>0: assert<br>1: de-assert. |
| 9 | R/W | 0x1 | PORTRST<br>Jtag portrst.<br>0: assert<br>1: de-assert. |
| 8 | R/W | 0x1 | TRST.<br>Jtag trst.<br>0: assert<br>1: de-assert. |
| 7:1 | / | / | / |
| 0 | R/W | 0x1 | GIC_RESET.<br>Gic_reset_cpu_reg<br>0: assert<br>1: de-assert. |

## 3.5. TimeStamp

### 3.5.1. Overview

The timestamp module generates and distributes a consistent timestamp value for multiple processors and other SOC IP .

It features:
- The Timestamp uses a timing reference with a fixed frequency clock OSC24M.
- The Timestamp functions as the clock source of the CPU local timer only.

### 3.5.2. Register List

| Module Name | Base Address |
|---|---|
| TIMESTAMP_STA | 0x01710000 |

Note: Timestamp status Register

| Register Name | Offset | Description |
|---|---|---|
| CNT_LOW_REG | 0x0000 | Counter low register[31:0] value |
| CNT_HI_REG | 0x0004 | Counter high register[63:32] value |

| Module Name | Base Address |
|---|---|
| TIMESTAMP_CTRL | 0x01720000 |

| Register Name | Offset | Description |
|---|---|---|
| TSTAMP_CTRL_REG | 0x0000 | Timestamp control register |
| CNT_LOW_REG | 0x0008 | Counter low register[31:0] value |
| CNT_HI_REG | 0x000C | Counter high register[63:32] value |
| CNT_FREQID_REG | 0x0020 | The Counter Base Frequency ID Register |

### 3.5.3. Timestamp Status Register Description

#### 3.5.3.1. Counter Low Register (Default Value: 0x00000000)

| Offset: 0x0000 | | | Register Name: **CNT_LOW_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R | 0x0 | CNT_LOW_REG. Current value of counter[31:0] |

#### 3.5.3.2. Counter High Register (Default Value: 0x00000000)

| Offset: 0x0004 | | | Register Name: **CNT_HI_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R | 0x0 | CNT_HI_REG. Current value of counter[64:32] |

### 3.5.4. Timestamp Counter Control Register Description

#### 3.5.4.1. Timestamp Control Register (Default Value: 0x00000000)

| **Offset: 0x0000** | | | **Register Name: TSTAMP_CTRL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:1 | / | / | /. |
| 0 | R/W | 0x0 | EN. Timestamp Counter Enable. 0: Disable 1: Enable |

#### 3.5.4.2. Counter Low Register (Default Value: 0x00000000)

| Offset: 0x0008 | | | Register Name: **CNT_LOW_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R | 0x0 | CNT_LOW_REG. Current value of counter[31:0] |

### 3.5.4.3. Counter High Register (Default Value: 0x00000000)

| Offset: 0x000C | | | Register Name: **CNT_HI_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R | 0x0 | CNT_HI_REG.<br>Current value of counter[64:32] |

### 3.5.4.4. Counter Base Frequency ID Register (Default : 0x00000000)

| Offset: 0x0020 | | | Register Name: **CNT_BASE_FREQID_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R/W | 0x0 | FREQ.<br>This register must be programmed to match the clock frequency of the timestamp generator, in ticks per second. For example, in our SOC for a 24 MHz clock,it must program 0x16E3600. |

## 3.5.5. Programming Guidelines

1) Timestamp Counter is used to provide time baseline for the local timer of processor, so the counter should be enabled before the OS runs.

2) Timestamp Counter Status Registers(TIMESTAMP STA) are readable only, indicating the current 64-bit counter value;

3) TSTAMP_CTRL_REG of Timestamp Counter Control Registers(TIMESTAMP CTRL) is used to enable or disable the counter; Base Frequency ID Register should define the timestamp generator frequency so that other software interface can access the timestap clock source information through this register.

4) CNT_LOW/HI_REG of Timestamp Counter Control Registers(TIMESTAMP CTRL) is writeable and readable, indicating the current 64-bit counter value. It is not recommended to write this register, however, if writing is required, you must follow steps below:

- Disable the counter;
- The first write access must be to the lower 32bits;(Counter Low Register)
- The second write access can be the high 32bits;(Counter High Register)
- The counter initial value then will be updated.

## 3.6. System Control

### 3.6.1. Overview

| Area | Address | Size(Bytes) |
|---|---|---|
| A1 | 0x00000000--0x00007FFF | 32K |
| A2 | 0x00044000--0x00053FFF | 64K |
| B(Secure RAM) | 0x00020000--0x00037FFF | 96K |
| C0_CPUX I-Cache | | 32K (X=0,1,2,3) |
| C0_CPUX D-Cache | | 32K (X=0,1,2,3) |
| Cluster 0 L2 Cache | | 512K |
| C1_CPUX I-Cache | | 32K (X=0,1,2,3) |
| C1_CPUX D-Cache | | 32K (X=0,1,2,3) |
| Cluster 1 L2 Cache | | 512K |
| Total | | 1728K |

### 3.6.2. System Control Register List

| Module Name | Base Address |
|---|---|
| System Control | 0x01C00000 |

| Register Name | Offset | Description |
|---|---|---|
| VER_REG | 0x24 | Version Register |
| EMAC_CLK_REG | 0x30 | EMAC Clock Register |

## 3.6.3.    System Control Register Description

### 3.6.3.1.   Version Register

| Offset:0x24 | | | Register Name: **VER_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:16 | / | / | / |
| 15 | / | / | / |
| 14:9 | / | / | / |
| 8 | R | x | UBOOT_SEL_PAD_STA. U_boot Select Pin Status. 0: U_Boot; 1: Normal Boot. |
| 7:0 | R | 0x0 | VER_BITS. This read-only bit field always reads back the mask revision level of the chip. |

**Note：**

The UBOOT_SEL_PAD_STA pin is indicated whether system should jump to USB boot, and the pin is pull up by internal 50K resistor in R_state.

### 3.6.3.2.   EMAC Clock Register (Default Value: 0x00000000)

| Offset:0x30 | | | Register Name: **EMAC_CLK_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:16 | / | / | / |
| 15 | R/W | 0x0 | TXC_DIV_CFG. Clock pre-divide radio. External transmit clock (125MHz) is pre-divided by as follows for RGMII. 0: /1, generate 125MHz 1: /5, generate 25MHz |
| 14:13 | / | / | / |
| 12:10 | R/W | 0x0 | ETXDC. Configure EMAC Transmit Clock Delay Chain. |
| 9:5 | R/W | 0x0 | ERXDC. Configure EMAC Receive Clock Delay Chain. |
| 4 | R/W | 0x0 | ERXIE Enable EMAC Receive Clock Invertor. 0: Disable; 1: Enable; |
| 3 | R/W | 0x0 | ETXIE Enable EMAC Transmit Clock Invertor. |

| | | | 0: Disable;<br>1: Enable; |
|---|---|---|---|
| 2 | R/W | 0x0 | EPIT<br>EMAC PHY Interface Type<br>0: GMII/MII;<br>1: RGMII; |
| 1:0 | R/W | 0x0 | ETCS.<br>EMAC Transmit Clock Source<br>00: Transmit clock source for MII;<br>01: External transmit clock source for GMII and RGMII;<br>10: Internal transmit clock source for GMII and RGMII;<br>11: Reserved; |

## 3.7. Timer

### 3.7.1. Overview

Timer 0/1 can take their inputs from Internal OSC or OSC24M. They provide the operating system's scheduler interrupt. It is designed to offer maximum accuracy and efficient management, even for systems with long or short response time. They provide 24-bit programmable overflow counter and work in auto-reload mode or no-reload mode. When the current value in *Current Value Register* is counting down to zero, the timer will generate interrupt if set interrupt enable bit.

The watchdog is used to resume the controller operation when it had been disturbed by malfunctions such as noise and system errors. It features a down counter that allows a watchdog period of up to 16 seconds (512000 cycles). It can generate a general reset or interrupt request.

### 3.7.2. Block Diagram
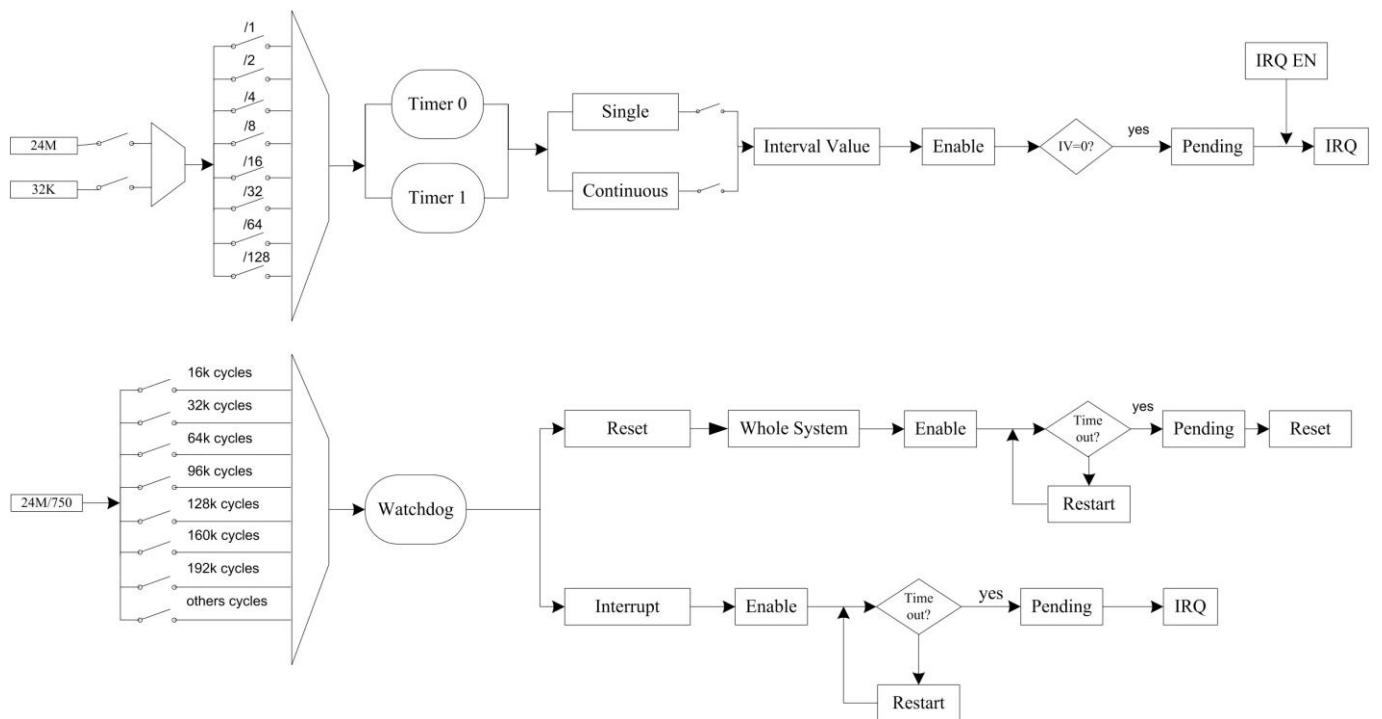


Figure 3-4. Timer Block Diagram

### 3.7.3. Timer Register List

| Module Name | Base Address |
|---|---|
| TIMER | 0x01C20C00 |

| Register Name | Offset | Description |
|---|---|---|
| TMR_IRQ_EN_REG | 0x0 | Timer IRQ Enable Register |
| TMR_IRQ_STA_REG | 0x4 | Timer Status Register |
| TMR0_CTRL_REG | 0x10 | Timer 0 Control Register |
| TMR0_INTV_VALUE_REG | 0x14 | Timer 0 Interval Value Register |
| TMR0_CUR_VALUE_REG | 0x18 | Timer 0 Current Value Register |
| TMR1_CTRL_REG | 0x20 | Timer 1 Control Register |
| TMR1_INTV_VALUE_REG | 0x24 | Timer 1 Interval Value Register |
| TMR1_CUR_VALUE_REG | 0x28 | Timer 1 Current Value Register |
| AVS_CNT_CTL_REG | 0x80 | AVS Control Register |
| AVS_CNT0_REG | 0x84 | AVS Counter 0 Register |
| AVS_CNT1_REG | 0x88 | AVS Counter 1 Register |
| AVS_CNT_DIV_REG | 0x8C | AVS Divisor Register |
| WDOG0_IRQ_EN_REG | 0xA0 | Watchdog 0 IRQ Enable Register |
| WDOG0_IRQ_STA_REG | 0xA4 | Watchdog 0 Status Register |
| WDOG0_CTRL_REG | 0xB0 | Watchdog 0 Control Register |
| WDOG0_CFG_REG | 0xB4 | Watchdog 0 Configuration Register |
| WDOG0_MODE_REG | 0xB8 | Watchdog 0 Mode Register |

### 3.7.4. Timer Register Description

#### 3.7.4.1. Timer IRQ Enable Register (Default Value: 0x00000000)

| Offset:0x0 | | | Register Name: **TMR_IRQ_EN_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:2 | / | / | / |
| 1 | R/W | 0x0 | TMR1_IRQ_EN. Timer 1 Interrupt Enable. 0: No effect; 1: Timer 1 Interval Value reached interrupt enable. |
| 0 | R/W | 0x0 | TMR0_IRQ_EN. Timer 0 Interrupt Enable. 0: No effect; |

| | | | 1: Timer 0 Interval Value reached interrupt enable. |
|---|---|---|---|

### 3.7.4.2. Timer IRQ Status Register (Default Value: 0x00000000)

| Offset:0x04 | | | Register Name: **TMR_IRQ_STA_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:2 | / | / | / |
| 1 | R/W | 0x0 | TMR1_IRQ_PEND. <br> Timer 1 IRQ Pending. Set 1 to the bit will clear it. <br> 0: No effect; <br> 1: Pending, timer 1 interval value is reached. |
| 0 | R/W | 0x0 | TMR0_IRQ_PEND. <br> Timer 0 IRQ Pending. Set 1 to the bit will clear it. <br> 0: No effect; <br> 1: Pending, timer 0 interval value is reached. |

### 3.7.4.3. Timer 0 Control Register (Default Value: 0x00000004)

| Offset:0x10 | | | Register Name: **TMR0_CTRL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:8 | / | / | / |
| 7 | R/W | 0x0 | TMR0_MODE. <br> Timer 0 mode. <br> 0: Continuous mode. When interval value reached, the timer will not disable automatically. <br> 1: Single mode. When interval value reached, the timer will disable automatically. |
| 6:4 | R/W | 0x0 | TMR0_CLK_PRES. <br> Select the pre-scale of timer 0 clock source. <br> 000: /1 <br> 001: /2 <br> 010: /4 <br> 011: /8 <br> 100: /16 <br> 101: /32 <br> 110: /64 <br> 111: /128 |
| 3:2 | R/W | 0x1 | TMR0_CLK_SRC. <br> Timer 0 Clock Source. <br> 00: 32K <br> 01: OSC24M. <br> 10: / |

| | | | 11: / |
|---|---|---|---|
| 1 | R/W | 0x0 | TMR0_RELOAD.<br>Timer 0 Reload.<br>0: No effect,<br>1: Reload timer 0 Interval value.<br>After the bit is set, it can not be written again before it's cleared automatically. |
| 0 | R/W | 0x0 | TMR0_EN.<br>Timer 0 Enable.<br>0: Stop/Pause,<br>1: Start.<br>When the timer is started, it will reload the interval value to internal register, and the current counter will count from interval value to 0. If the current counter does not reach the zero, the timer enable bit is set to "0", the current value counter will pause. At least wait for 2 cycles, the start bit can be set to 1.In timer pause state; the interval value register can be modified. If the timer is started again, and the Software hope the current value register to down-count from the new interval value, the reload bit and the enable bit should be set to 1 at the same time. |

### 3.7.4.4. Timer 0 Interval Value Register(Default Value: 0x00000000)

| Offset:0x14 | | | Register Name: **TMR0_INTV_VALUE_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R/W | 0x0 | TMR0_INTV_VALUE.<br>Timer 0 Interval Value.<br>The value setting should consider the system clock and the timer clock source. |

### 3.7.4.5. Timer 0 Current Value Register(Default Value: 0x00000000)

| Offset:0x18 | | | Register Name: **TMR0_CUR_VALUE_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R/W | 0x0 | TMR0_CUR_VALUE.<br>Timer 0 Current Value.<br>It is a 32-bit down-counter (from interval value to 0). |

### 3.7.4.6. Timer 1 Control Register (Default Value: 0x00000004)

| Offset:0x20 | | | Register Name: **TMR1_CTRL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |

| 31:8 | / | / | / |
|------|------|------|------|
| 7 | R/W | 0x0 | TMR1_MODE. <br> Timer 1 mode. <br> 0: Continuous mode. When interval value reached, the timer will not disable automatically. <br> 1: Single mode. When interval value reached, the timer will disable automatically. |
| 6:4 | R/W | 0x0 | TMR1_CLK_PRES. <br> Select the pre-scale of timer 1 clock source. <br> 000: /1 <br> 001: /2 <br> 010: /4 <br> 011: /8 <br> 100: /16 <br> 101: /32 <br> 110: /64 <br> 111: /128 |
| 3:2 | R/W | 0x1 | TMR1_CLK_SRC. <br> 00: 32K <br> 01: OSC24M. <br> 10: / <br> 11: /. |
| 1 | R/W | 0x0 | TMR1_RELOAD. <br> Timer 1 Reload. <br> 0: No effect, 1: Reload timer 1 Interval value. <br> After the bit is set, it can not be written again before it's cleared automatically. |
| 0 | R/W | 0x0 | TMR1_EN. <br> Timer 1 Enable. <br> 0: Stop/Pause, 1: Start. <br> If the timer is started, it will reload the interval value to internal register, and the current counter will count from interval value to 0. <br> If the current counter does not reach the zero, the timer enable bit is set to "0", the current value counter will pause. At least wait for 2 cycles, the start bit can be set to 1. <br> In timer pause state, the interval value register can be modified. If the timer is started again, and the Software hope the current value register to down-count from the new interval value, the reload bit and the enable bit should be set to 1 at the same time. |

### 3.7.4.7. Timer 1 Interval Value Register(Default Value: 0x00000000)

| Offset:0x24 | | | Register Name: **TMR1_INTV_VALUE_REG** |
|------|------|------|------|
| Bit | R/W | Default/Hex | Description |

| 31:0 | R/W | 0x0 | TMR1_INTV_VALUE. Timer 1 Interval Value. The value setting should consider the system clock and the timer clock source. |

### 3.7.4.8. Timer 1 Current Value Register(Default Value: 0x00000000)

| Offset:0x28 | | | Register Name: **TMR1_CUR_VALUE_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R/W | 0x0 | TMR1_CUR_VALUE. Timer 1 Current Value. It is a 32-bit down-counter (from interval value to 0). |

### 3.7.4.9. AVS Counter Control Register (Default Value: 0x00000000)

| Offset:0x80 | | | Register Name: **AVS_CNT_CTL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:10 | / | / | / |
| 9 | R/W | 0x0 | AVS_CNT1_PS. Audio/Video Sync Counter 1 Pause Control 0: Not pause, 1: Pause Counter 1. |
| 8 | R/W | 0x0 | AVS_CNT0_PS. Audio/Video Sync Counter 0 Pause Control 0: Not pause, 1: Pause Counter 0. |
| 7:2 | / | / | / |
| 1 | R/W | 0x0 | AVS_CNT1_EN. Audio/Video Sync Counter 1 Enable/ Disable. The counter source is OSC24M. 0: Disable, 1: Enable. |
| 0 | R/W | 0x0 | AVS_CNT0_EN. Audio/Video Sync Counter 1 Enable/ Disable. The counter source is OSC24M. 0: Disable, 1: Enable. |

### 3.7.4.10. AVS Counter 0 Register (Default Value: 0x00000000)

| Offset:0x84 | | | Register Name: **AVS_CNT0_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R/W | 0x0 | AVS_CNT0. |

| | | | Counter 0 for Audio/ Video Sync Application |
|---|---|---|---|
| | | | The high 32 bits of the internal 33-bits counter register. The initial value of the internal 33-bits counter register can be set by software. The LSB bit of the 33-bits counter register should be zero when the initial value is updated. It will count from the initial value. The initial value can be updated at any time. It can also be paused by setting AVS_CNT0_PS to '1'. When it is paused, the counter won't increase. |

### 3.7.4.11. AVS Counter 1 Register (Default Value: 0x00000000)

| Offset:0x88 | | | Register Name: **AVS_CNT1_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R/W | 0x0 | AVS_CNT1. Counter 1 for Audio/ Video Sync Application The high 32 bits of the internal 33-bits counter register. The initial value of the internal 33-bits counter register can be set by software. The LSB bit of the 33-bits counter register should be zero when the initial value is updated. It will count from the initial value. The initial value can be updated at any time. It can also be paused by setting AVS_CNT1_PS to '1'. When it is paused, the counter won't increase. |

### 3.7.4.12. AVS Counter Divisor Register (Default Value: 0x05DB05DB)

| Offset:0x8C | | | Register Name: **AVS_CNT_DIV_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:28 | / | / | / |
| 27:16 | R/W | 0x5DB | AVS_CNT1_D. Divisor N for AVS Counter 1 AVS CN1 CLK=24MHz/Divisor_N1. Divisor N1 = Bit [27:16] + 1. The number N is from 1 to 0x7ff. The zero value is reserved. The internal 33-bits counter engine will maintain another 12-bits counter. The 12-bits counter is used for counting the cycle number of one 24Mhz clock. When the 12-bits counter reaches (>= N) the divisor value, the internal 33-bits counter register will increase 1 and the 12-bits counter will reset to zero and restart again. |
| 15:12 | / | / | / |
| 11:0 | R/W | 0x5DB | AVS_CNT0_D. Divisor N for AVS Counter 0 AVS CN0 CLK=24MHz/Divisor_N0. Divisor N0 = Bit [11:0] + 1 The number N is from 1 to 0x7ff. The zero value is reserved. |

| | | | The internal 33-bits counter engine will maintain another 12-bits counter. The 12-bits counter is used for counting the cycle number of one 24Mhz clock. When the 12-bits counter reaches (>= N) the divisor value, the internal 33-bits counter register will increase 1 and the 12-bits counter will reset to zero and restart again. |

### 3.7.4.13. Watchdog0 IRQ Enable Register (Default Value: 0x00000000)

| Offset:0xA0 | | | Register Name: **WDOG0_IRQ_EN_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:1 | / | / | / |
| 0 | R/W | 0x0 | WDOG0_IRQ_EN.<br>Watchdog0 Interrupt Enable.<br>0: No effect,<br>1: Watchdog0 interrupt enable. |

### 3.7.4.14. Watchdog0 Status Register (Default Value: 0x00000000)

| Offset:0xA4 | | | Register Name: **WDOG0_IRQ_STA_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:1 | / | / | / |
| 0 | R/W | 0x0 | WDOG0_IRQ _PEND.<br>Watchdog0 n IRQ Pending. Set 1 to the bit will clear it.<br>0: No effect,<br>1: Pending, watchdog0 interval value is reached. |

### 3.7.4.15. Watchdog0 Control Register (Default Value: 0x00000000)

| Offset:0xB0 | | | Register Name: **WDOG0_CTRL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:13 | / | / | / |
| 12:1 | R/W | 0x0 | WDOG0_KEY_FIELD.<br>Watchdog0 Key Field.<br>Should be written at value 0xA57. Writing any other value in this field aborts the write operation. |
| 0 | R/W | 0x0 | WDOG0_RSTART.<br>Watchdog0 Restart.<br>0: No effect,<br>1: Restart watchdog0. |

**3.7.4.16.** **Watchdog0 Configuration Register (Default Value: 0x00000001)**

| Offset:0xB4 | | | Register Name: **WDOG0_CFG_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:2 | / | / | / |
| 1:0 | R/W | 0x1 | WDOG0_CONFIG.<br>Watchdog0 generates a reset signal<br>00: /<br>01: To whole system<br>10: Only interrupt<br>11: / |

**3.7.4.17.** **Watchdog0 Mode Register (Default Value: 0x00000000)**

| Offset:0xB8 | | | Register Name: **WDOG0_MODE_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:8 | / | / | / |
| 7:4 | R/W | 0x0 | WDOG0_INTV_VALUE.<br>Watchdog0 Interval Value<br>Watchdog0 clock source is *OSC24M / 750*. If the clock source is turned off, Watchdog 0 will not work.<br>0000: 16000 cycles (0.5s)<br>0001: 32000 cycles (1s)<br>0010: 64000 cycles (2s)<br>0011: 96000 cycles (3s)<br>0100: 128000 cycles (4s)<br>0101: 160000 cycles (5s)<br>0110: 192000 cycles (6s)<br>0111: 256000 cycles (8s)<br>1000: 320000 cycles (10s)<br>1001: 384000 cycles (12s)<br>1010: 448000 cycles (14s)<br>1011: 512000 cycles (16s)<br>others: / |
| 3:1 | / | / | / |
| 0 | R/W | 0x0 | WDOG0_EN.<br>Watchdog0 Enable.<br>0: No effect;<br>1: Enable watchdog0. |

## 3.7.5.    Programming Guidelines

### 3.7.5.1.  Timer

Take making a Timer0 1ms delay for an example, 24M clock source, single mode and 2 pre-scale will be selected in the instance.

```
writel(0x2EE0,TMR_0_INTV);                          //Set interval value
writel(0x94, TMR_0_CTRL);                           //Select Single mode,24MHz clock source,2 pre-scale
writel(readl(TMR_0_CTRL)|(1<<1), TMR_0_CTRL);       //Set Reload bit
while((readl(TMR_0_CTRL)>>1)&1);                     //Waiting Reload bit turns to 0
writel(readl(TMR_0_CTRL)|(1<<0), TMR_0_CTRL);       //Enable Timer0
```

### 3.7.5.2.  Watchdog Reset

In the following instance making configurations for Watchdog: configurate clock source as 24M/750, configurate Interval Value as 1s and configurate Watchdog Configuration as To whole system. This instance indicates that reset system after 1s.

```
writel(0x1, WDOG_CONFIG);                           //To whole system
writel(0x10, WDOG_MODE);                            //Interval Value set 1s
writel(readl(WDOG_MODE)|(1<<0), WDOG_MODE);         //Enable Watchdog
```

### 3.7.5.3.  Watchdog Restart

In the following instance making configurations for Watchdog: configurate clock source as 24M/750, configurate Interval Value as 1s and configurate Watchdog Configuration as To whole system. In the following instance, if the time of other codes is larger than 1s, watchdog will reset the whole system. If the sentence of restart watchdog is implemented inside 1s, watchdog will be restarted.

```
writel(0x1, WDOG_CONFIG);                           //To whole system
writel(0x10, WDOG_MODE);                            //Interval Value set 1s
writel(readl(WDOG_MODE)|(1<<0), WDOG_MODE);         //Enable Watchdog
----other codes---
writel(readl(WDOG_CTRL)|(0xA57<<1)|(1<<0)，WDOG_CTRL); //Writel 0xA57 at Key Field and Restart Watchdog
```

## 3.8. Trusted Watchdog

### 3.8.1. Overview

The trusted watchdog is primarily used to protect the trusted world operations from denial of service when secure services are dependent to the RichOS scheduler. For example, if the trusted world is not entered after a defined time limit the SoC is re-started to perform an authentication of the system.

The trusted watchdog can also be used to mask the real cause of a security error thanks to the delayed warm reset it generates.

### 3.8.2. Block Diagram
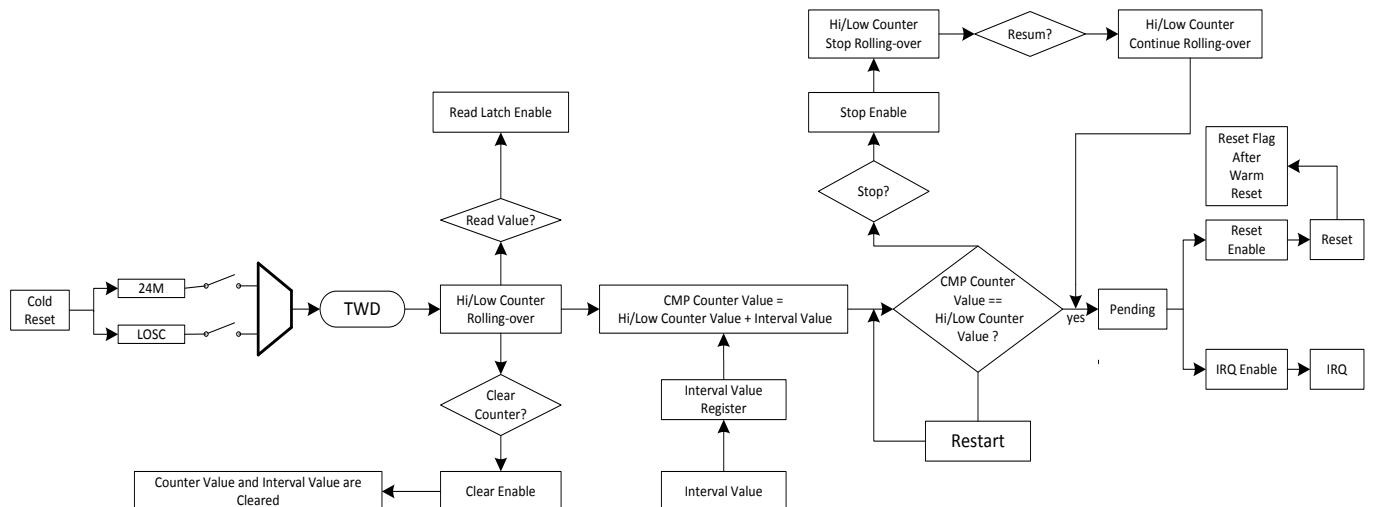


Figure 3-5. TWD Block Diagram

The trusted watchdog must always be running when the SoC wakes up from cold reset and can be refreshed, suspended, or reset only by secure accesses. And a clock of at least 32 kHz is used when the device is not a power saving cycles.

The trusted watchdog timer is able to generate a SoC warm reset after a duration programmed into the timer or set by default in hardware. And the flag indicating the occurrence of a watchdog triggered warm reset has occurred since the last cold reset.

Clock sources driving the watchdog timer must be controlled or managed by a trusted entity. This means that non-trusted world accesses are not permitted to turn on, turn off or modify the characteristics of clock source. The **Clear Enable** will reset relevant bits in the watchdog registers, except the reset flag.

### 3.8.3. TWD Register List

| Module Name | Base Address |
|---|---|
| TWD | 0x01F01800 |

| Register Name | Offset | Description |
|---|---|---|
| TWD_STATUS_REG | 0x0000 | TWD Status Register |
| TWD_CTRL_REG | 0x0010 | TWD Control Register |
| TWD_RESTART_REG | 0x0014 | TWD Restart Register |
| TWD_LOW_CNT_REG | 0x0020 | TWD Low Counter Register |
| TWD_HIGH_CNT_REG | 0x0024 | TWD High Counter Register |
| TWD_INTV_VAL_REG | 0x0030 | TWD Interval Value Register |
| TWD_LOW_CNT_CMP_REG | 0x0040 | TWD Low Counter Compare Register |
| TWD_HIGH_CNT_CMP_REG | 0x0044 | TWD High Counter Compare Register |
| SST_NV_CNT_REG | 0x0100 | Secure Storage NV-Counter Register |
| SYN_DATA_CNT_REG0 | 0x0110 | Synchronize Data Counter Register 0 |
| SYN_DATA_CNT_REG1 | 0x0114 | Synchronize Data Counter Register 1 |
| SYN_DATA_CNT_REG2 | 0x0118 | Synchronize Data Counter Register 2 |
| SYN_DATA_CNT_REG3 | 0x011C | Synchronize Data Counter Register 3 |

### 3.8.4. TWD Register Description

#### 3.8.4.1. TWD Status Register (Default Value: 0x00000000)

| Offset: 0x0000 | | | Register Name: **TWD_STATUS_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:1 | / | / | / |
| 0 | R/W | 0x0 | TWD_PEND_FLAG.<br>Interrupt pending. Set 1 to the bit will clear it.<br>0: No effect.<br>1: Pending. |

#### 3.8.4.2. TWD Control Register (Default Value: 0x00000000)

| Offset: 0x0010 | | | Register Name: **TWD_CTRL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |

| 31 | R/W | 0x0 | CNT64_CLK_SRC_SEL.<br>64-bit counter clock source select.<br>0: LOSC.<br>1: OSC24M. |
|---|---|---|---|
| 30:10 | / | / | / |
| 9 | R/W | 0x0 | TWD_RESET_EN.<br>TWD reset enable.<br>0: Reset disable.<br>1: Reset enable. |
| 8 | R/W | 0x0 | TWD_INT_EN.<br>TWD Interrupt Enable.<br>0: Interrupt disable.<br>1: Interrupt enable. |
| 7:2 | / | / | / |
| 1 | R/W | 0x0 | TWD_STOP_EN.<br>TWD stop enable.<br>0: Resume rolling-over.<br>1: Stop rolling-over. |
| 0 | R/W | 0x0 | TWD_CLR_EN.<br>TWD clear enable.<br>0: No effect.<br>1: To clear relevant registers and it will change to zero after the registers are cleared. |

### 3.8.4.3. TWD Restart Register (Default Value: 0x00000000)

| Offset: 0x0014 | | | Register Name: **TWD_RESTART_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:28 | / | / | / |
| 27:16 | WO | 0x0 | TWD_RESTART_KEYFILED.<br>Should be written at value 0xD14. Writing any other value in this field aborts the write operation. |
| 15:1 | / | / | / |
| 0 | WO | 0x0 | TWD_RESTART_EN.<br>If writing '1' in this bit, the value of *Counter Compare Registers* would change.<br>0: No effect.<br>1: Restart enable. |

### 3.8.4.4. TWD Low Counter Register (Default Value: 0x00000000)

| Offset: 0x0020 | Register Name: **TWD_LOW_CNT_REG** |
|---|---|

| Bit | R/W | Default/Hex | Description |
|---|---|---|---|
| 31:0 | RO | 0x0 | TWD_LOW_CNT. <br> The TWD low 32-bit counter. |

### 3.8.4.5.  TWD High Counter Register (Default Value: 0x00000000)

| Offset: 0x0024 | | | Register Name: **TWD_HIGH_CNT_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | RO | 0x0 | TWD_HIGH_CNT. <br> The TWD high 32-bit counter. |

### 3.8.4.6.  TWD Interval Value Register (Default Value: 0x00000000)

| Offset: 0x0030 | | | Register Name: **TWD_INTV_VAL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R/W | 0x0 | TWD_INTV_VAL. <br> The TWD interval value. |

### 3.8.4.7.  TWD Low Counter Compare Register (Default Value: 0x00000000)

| Offset: 0x0040 | | | Register Name: **TWD_LOW_CNT_CMP_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | RO | 0x0 | TWD_LOW_CMP. <br> The TWD low 32-bit compare counter. |

### 3.8.4.8.  TWD High Counter Compare Register (Default Value: 0x00000000)

| Offset: 0x0044 | | | Register Name: **TWD_HIGH_CNT_CMP_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | RO | 0x0 | TWD_HIGH_CMP. <br> The TWD high 32-bit compare counter. |

### 3.8.4.9.  Secure Storage NV-Counter Register (Default Value: 0x00000000)

| Offset: 0x0100 | | | Register Name: **SST_NV_CNT_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R/W | 0x0 | SST_NV_CNT. <br> This counter protects the trusted world Secure Storage file from replay |

| | | | attacks. |
|---|---|---|---|

### 3.8.4.10. Synchronize Data Counter Register 0 (Default Value: 0x00000000)

| Offset: 0x0110 | | | Register Name: **SYN_DATA_CNT_REG0** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R/W | 0x0 | SYN_DATA_CNT0. This counter is used for synchronizing data stores against replay attacks. |

### 3.8.4.11. Synchronize Data Counter Register 1 (Default Value: 0x00000000)

| Offset: 0x0114 | | | Register Name: **SYN_DATA_CNT_REG1** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R/W | 0x0 | SYN_DATA_CNT1. This counter is used for synchronizing data stores against replay attacks. |

### 3.8.4.12. Synchronize Data Counter Register 2 (Default Value: 0x00000000)

| Offset: 0x0118 | | | Register Name: **SYN_DATA_CNT_REG2** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R/W | 0x0 | SYN_DATA_CNT2. This counter is used for synchronizing data stores against replay attacks. |

### 3.8.4.13. Synchronize Data Counter Register 3 (Default Value: 0x00000000)

| Offset: 0x011C | | | Register Name: **SYN_DATA_CNT_REG3** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R/W | 0x0 | SYN_DATA_CNT3. This counter is used for synchronizing data stores against replay attacks. |

## 3.9. High-speed Timer

### 3.9.1. Overview

High Speed Timer Clock Source are fixed to AHBCLK, which is much higher than OSC24M. Compared with other timers, High Speed Timer clock source is synchronized with AHB clock. When the current value in both LO and HI Current Value Register are counting down to zero, the timer will generate interrupt if set interrupt enable bit.

The HSTimer includes the following features:

* 56-bit counter
* Clock source is synchronized with AHB clock, which means calculating much more accurate than other timers

### 3.9.2. Operation Principle

#### 3.9.2.1. HSTimer clock gating and software reset

By default the HSTimer clock gating is mask. When it is necessary to use HSTimer, it's clock gating should be open in **BUS Clock Gating Register0** and then de-assert the software reset in **BUS Software Reset Register0** on CCU module. If it is no need to use HSTimer, both the gating bit and software reset bit should be set 0.

#### 3.9.2.2. HSTimer reload bit

Differing from the reload of Timer, when interval value is reloaded into current value register, the reload bit would not turn to 0 automatically until you clear it. If software hopes the current value register to down-count from the new interval value in pause status, the reload bit and the enable bit should be written 1 at the same time.

### 3.9.3. HSTimer Register List

| Module Name | Base Address |
|---|---|

| High Speed Timer | 0x01C60000 |
|---|---|

| Register Name | Offset | Description |
|---|---|---|
| HS_TMR_IRQ_EN_REG | 0x00 | HS Timer IRQ Enable Register |
| HS_TMR_IRQ_STAS_REG | 0x04 | HS Timer Status Register |
| HS_TMR_CTRL_REG | 0x10 | HS Timer Control Register |
| HS_TMR_INTV_LO_REG | 0x14 | HS Timer Interval Value Low Register |
| HS_TMR_INTV_HI_REG | 0x18 | HS Timer Interval Value High Register |
| HS_TMR_CURNT_LO_REG | 0x1C | HS Timer Current Value Low Register |
| HS_TMR_CURNT_HI_REG | 0x20 | HS Timer Current Value High Register |

## 3.9.4.    HSTimer Register Description

### 3.9.4.1.   HS Timer IRQ Enable Register (Default Value: 0x00000000)

| Offset:0x0 | | | Register Name: **HS_TMR_IRQ_EN_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:1 | / | / | / |
| 0 | R/W | 0x0 | HS_TMR_INT_EN.<br>High Speed Timer Interrupt Enable.<br>0: No effect;<br>1: High Speed Timer Interval Value reached interrupt enable. |

### 3.9.4.2.   HS Timer IRQ Status Register (Default Value: 0x00000000)

| Offset:0x4 | | | Register Name: **HS_TMR_IRQ_STAS_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:1 | / | / | / |
| 0 | R/W | 0x0 | HS_TMR_IRQ_PEND.<br>High Speed Timer IRQ Pending. Set 1 to the bit will clear it.<br>0: No effect;<br>1: Pending, High speed timer interval value is reached. |

### 3.9.4.3.   HS Timer Control Register (Default Value: 0x00000000)

| Offset:0x10 | Register Name: **HS_TMR_CTRL_REG** |
|---|---|

| Bit | R/W | Default/Hex | Description |
|---|---|---|---|
| 31:8 | / | / | / |
| 7 | R/W | 0x0 | HS_TMR_MODE. High Speed Timer mode. 0: Continuous mode. When interval value reached, the timer will not disable automatically. 1: Single mode. When interval value reached, the timer will disable automatically. |
| 6:4 | R/W | 0x0 | HS_TMR_CLK Select the pre-scale of the high speed timer clock sources. 000: /1 001: /2 010: /4 011: /8 100: /16 101: / 110: / 111: / |
| 3:2 | / | / | / |
| 1 | R/W | 0x0 | HS_TMR_RELOAD. High Speed Timer Reload. 0: No effect, 1: Reload High Speed Timer Interval Value. |
| 0 | R/W | 0x0 | HS_TMR_EN. High Speed Timer Enable. 0: Stop/Pause, 1: Start. If the timer is started, it will reload the interval value to internal register, and the current counter will count from interval value to 0. If the current counter does not reach the zero, the timer enable bit is set to "0", the current value counter will pause. At least wait for 2 cycles, the start bit can be set to 1. In timer pause state, the interval value register can be modified. If the timer is started again, and the Software hope the current value register to down-count from the new interval value, the reload bit and the enable bit should be set to 1 at the same time. |

### 3.9.4.4. HS Timer Interval Value Lo Register

| Offset:0x14 | | | Register Name: **HS_TMR_INTV_LO_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R/W | x | HS_TMR_INTV_VALUE_LO. High Speed Timer Interval Value [31:0]. |

### 3.9.4.5. HS Timer Interval Value Hi Register

| Offset:0x18 | | | Register Name: **HS_TMR_INTV_HI_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:24 | / | / | / |
| 23:0 | R/W | x | HS_TMR_INTV_VALUE_HI.<br>High Speed Timer Interval Value [55:32]. |

**Note:**

The interval value register is a 56-bit register. When read or write the interval value, the Lo register should be read or write first. And the Hi register should be written after the Lo register.

### 3.9.4.6. HS Timer Current Value Lo Register

| Offset:0x1C | | | Register Name: **HS_TMR_CURNT_LO_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R/W | x | HS_TMR_CUR_VALUE_LO.<br>High Speed Timer Current Value [31:0]. |

### 3.9.4.7. HS Timer Current Value Hi Register

| Offset:0x20 | | | Register Name: **HS_TMR_CURNT_HI_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:24 | / | / | / |
| 23:0 | R/W | x | HS_TMR_CUR_VALUE_HI.<br>High Speed Timer Current Value [55:32]. |

**Note:**

- HSTimer current value is a 56-bit down-counter (from interval value to 0).
- The current value register is a 56-bit register. When read or write the current value, the Lo register should be read or write first.

## 3.9.5. Programming Guidelines

Take making a 1us delay using HSTimer for an instance as follow, AHB1CLK will be configurated as 100MHz and n_mode, Single mode and 2 pre-scale will be selected in this instance.

```
writel(0x0, HS_TMR_INTV_HI);          //Set interval value Hi 0x0
writel(0x32, HS_TMR_INTV_LO);         //Set interval value Lo 0x32
writel(0x90, HS_TMR_CTRL);            //Select n_mode,2 pre-scale,single mode
```

```
writel(readl(HS_TMR_CTRL)|(1<<1), HS_TMR_CTRL);        //Set Reload bit
writel(readl(HS_TMR_CTRL)|(1<<0), HS_TMR_CTRL);        //Enable HSTimer
While(!(readl(HS_TMR_IRQ_STAT)&1));                     //Wait for HSTimer to generate pending
Writel(1,HS_TMR_IRQ_STAT);                             //Clear HSTimer pending
```

## 3.10.  PWM

### 3.10.1.  Overview

PWM is suitable for display device such as LVDS.The output of the PWM is a toggling signal whose frequency and duty cycle can be modulated by its programmable registers. Each channel has a dedicated internal 16-bit up counter. If the counter reaches the value stored in the channel period register, it resets. At the beginning of a count period cycle, the PWMOUT is set to active state and count from 0x0000.

The PWM divider divides the clock(24MHz) by 1~4096 according to the pre-scalar bits in the PWM control register.

In PWM cycle mode, the output will be a square waveform, the frequency is set to the period register. In PWM pulse mode, the output will be a positive pulse or a negative pulse.

### 3.10.2.  Block Diagram
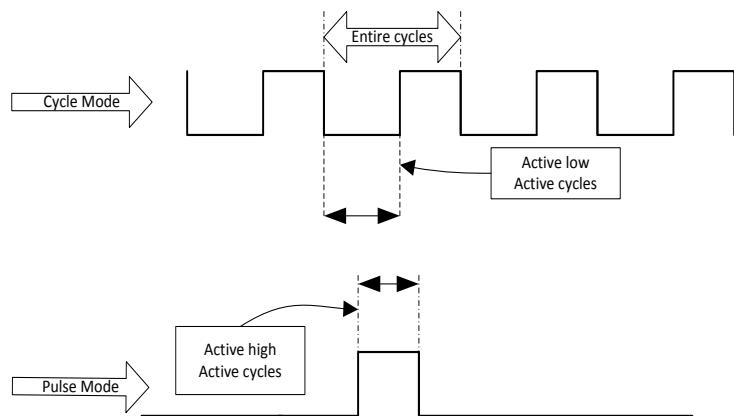


Figure 3-6. PWM Block Diagram

The PWM divider divides the clock (24MHz) by 1-64 according to the pre-scalar bits in the PWM control register. The PWM output Frequency can be divided by 65536 at most. In PWM cycle mode, the output will be a square waveform; the frequency is set to the period register. In PWM pulse mode, the output will be a positive pulse or a negative pulse.

PWM Outputs continuous waveform in Continuous Mode and a pulse waveform in Pulse Mode.Each PWM channel has two 16bit counters exiting in corresponding Period Register, whose bit[31:16] indicate one 16bits counter for counting Entire Cycle and bit[15:0] indicate the other 16bits counter for counting Active Cycle.

## 3.10.3.  PWM Register List

| Module Name | Base Address |
|---|---|
| PWM | 0x01C21400 |

| Register Name | Offset | Description |
|---|---|---|
| PWM_CH_CTRL | 0x00 | PWM Control Register |
| PWM_CH0_PERIOD | 0x04 | PWM Channel 0 Period Register |

## 3.10.4.  PWM Register Description

### 3.10.4.1.  PWM Control Register(Default Value: 0x00000000)

| Offset:0x0 | | | Register Name: **PWM_CTRL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:29 | / | / | /. |
| 28 | RO | 0x0 | PWM0_RDY.<br>PWM0 period register ready.<br>0: PWM0 period register is ready to write,<br>1: PWM0 period register is busy. |
| 27:10 | / | / | / |
| 9 | R/W | 0x0 | PWM0_BYPASS.<br>PWM CH0 bypass enable.<br>If the bit is set to 1, PWM0's output is OSC24MHz.<br>0: disable,<br>1: enable. |
| 8 | R/W | 0x0 | PWM_CH0_PUL_START.<br>PWM Channel 0 pulse output start.<br>0: no effect, 1: output 1 pulse.<br>The pulse width should be according to the period 0 register[15:0],and the pulse state should be according to the active state.<br>After the pulse is finished,the bit will be cleared automatically. |
| 7 | R/W | 0x0 | PWM_CHANNEL0_MODE.<br>0: cycle mode, 1: pulse mode. |

| 6 | R/W | 0x0 | SCLK_CH0_GATING. |
|---|-----|-----|---|
| | | | Gating the Special Clock for PWM0(0: mask, 1: pass). |
| 5 | R/W | 0x0 | PWM_CH0_ACT_STA. |
| | | | PWM Channel 0 Active State. |
| | | | 0: Low Level, 1: High Level. |
| 4 | R/W | 0x0 | PWM_CH0_EN. |
| | | | PWM Channel 0 Enable. |
| | | | 0: Disable, 1: Enable. |
| 3:0 | R/W | 0x0 | PWM_CH0_PRESCAL. |
| | | | PWM Channel 0 Prescalar. |
| | | | These bits should be setting before the PWM Channel 0 clock gate on. |
| | | | 0000: /120 |
| | | | 0001: /180 |
| | | | 0010: /240 |
| | | | 0011: /360 |
| | | | 0100: /480 |
| | | | 0101: / |
| | | | 0110: / |
| | | | 0111: / |
| | | | 1000: /12k |
| | | | 1001: /24k |
| | | | 1010: /36k |
| | | | 1011: /48k |
| | | | 1100: /72k |
| | | | 1101: / |
| | | | 1110: / |
| | | | 1111: /1 |

### 3.10.4.2.  PWM Channel 0 Period Register(Default Value: 0x00000000)

| Offset:0x4 | | | Register Name: **PWM_CH0_PERIOD** |
|---|-----|-----|---|
| Bit | R/W | Default/Hex | Description |
| 31:16 | R/W | x | PWM_CH0_ENTIRE_CYS |
| | | | Number of the entire cycles in the PWM clock. |
| | | | 0 = 1 cycle |
| | | | 1 = 2 cycles |
| | | | …… |
| | | | N = N+1 cycles |
| | | | If the register need to be modified dynamically, the PCLK should be faster than the PWM CLK (PWM CLK = 24MHz/pre-scale). |
| 15:0 | R/W | x | PWM_CH0_ENTIRE_ACT_CYS |
| | | | Number of the active cycles in the PWM clock. |
| | | | 0 = 0 cycle |
| | | | 1 = 1 cycles |

| | | | ...... |
|---|---|---|---|
| | | | N = N cycles |

**Note:**

The active cycles should be no larger than the period cycles.

## 3.11. DMA

### 3.11.1. Overview

There are 8 DMA channels in the chip. Each DMA channel can generate interrupts. According to different pending status, the referenced DMA channel generates corresponding interrupt. And, the configuration information of every DMA channel are storing in the DDR or SRAM. When start a DMA transferring, the **DMA Channel Descriptor Address Register** contains the address information in the DDR or SRAM, where has the relevance configuration information of the DMA transferring.

### 3.11.2. Functionalities Description

### 3.11.2.1. Block Diagram
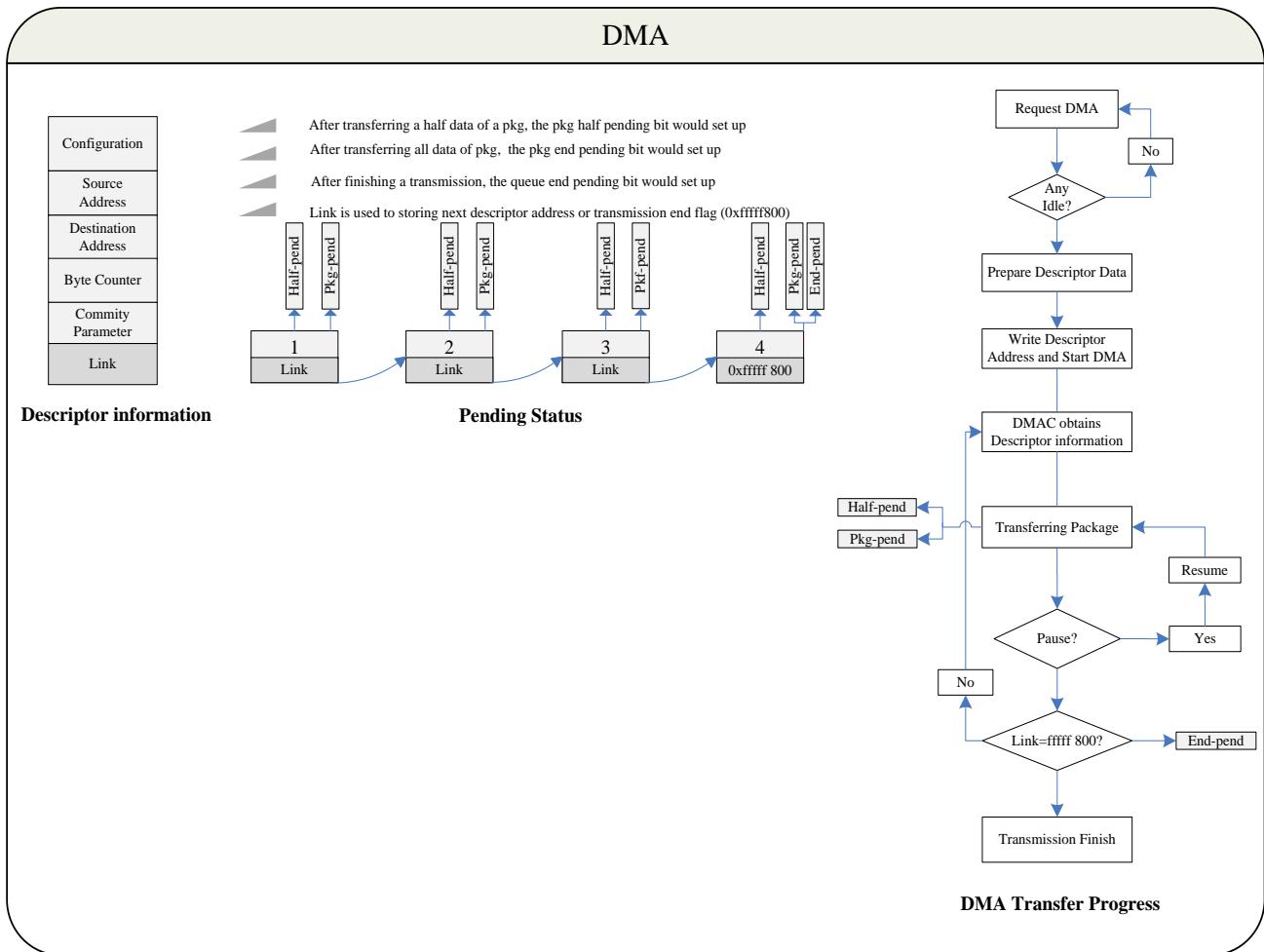


Figure 3-7. DMA Block Diagram

### 3.11.2.2. DRQ Type and Corresponding Relation

Table 3-1. DMA DRQ Table

| Source DRQ Type | | Destination DRQ Type | |
| --- | --- | --- | --- |
| **Port NO.** | **Module Name** | **Port NO.** | **Module Name** |
| Port 0 | SRAM | Port 0 | SRAM |
| Port 1 | SDRAM | Port 1 | SDRAM |
| Port 2 | OWA_RX | Port 2 | OWA_TX |
| Port 3 | DAUDIO0_RX | Port 3 | DAUDIO0_TX |
| Port 4 | DAUDIO1_RX | Port 4 | DAUDIO1_TX |
| Port 5 | NAND | Port 5 | NAND |
| Port 6 | UART0_RX | Port 6 | UART0_TX |
| Port 7 | UART1_RX | Port 7 | UART1_TX |
| Port 8 | UART2_RX | Port 8 | UART2_TX |

| Port 9 | UART3_RX | Port 9 | UART3_TX |
|--------|----------|--------|----------|
| Port 10 | UART4_RX | Port 10 | UART4_TX |
| Port 11 | / | Port 11 | / |
| Port 12 | / | Port 12 | / |
| Port 13 | / | Port 13 | / |
| Port 14 | / | Port 14 | / |
| Port 15 | / | Port 15 | / |
| Port 16 | / | Port 16 | / |
| Port 17 | USB DRD_EP1 | Port 17 | USB DRD_EP1 |
| Port 18 | USB DRD_EP2 | Port 18 | USB DRD_EP2 |
| Port 19 | USB DRD_EP3 | Port 19 | USB DRD_EP3 |
| Port20 | USB DRD_EP4 | Port 20 | USB DRD_EP4 |
| Port 21 | USB DRD_EP5 | Port 21 | USB DRD_EP5 |
| Port 22 | / | Port 22 | / |
| Port 23 | SPI0_RX | Port 23 | SPI0_TX |
| Port 24 | SPI1_RX | Port 24 | SPI1_TX |
| Port 25 | | Port 25 | |
| Port 26 | | Port 26 | |
| Port 27 | | Port 27 | DAUDIO2_TX |
| Port 28 | TDM-RX | Port 28 | TDM-TX |
| Port 29 | | Port 29 | |
| Port 30 | | Port 30 | |

**Note:**

SRAM or DRAM DRQ signal is always high.


### 3.11.2.3. DMA Descriptor

In this section, the DMA descriptor registers will be introduced in detail.

When starting a DMA transmission, the module data are transferred as packages, which have the link data information. And, by reading the DMA Status Register, the status of a DMA channel could be known. Reading back the descriptor address register, the value is the link data in the transferring package. If only the value is equal to 0xffffff800, then it can be regarded as NULL, which means the package is the last package in this DMA transmission. Otherwise, the value means the start address of the next package. And, the Descriptor Address Register can be changed during a package transferring.

When transferring the half of a package, the relevant pending bit will be set up automatically, and if the corresponding interrupt is enabled, DMA generates an interrupt to the system. The similar thing would occur when transferring a package completely. Meanwhile, if DMA have transferred the last package in the data, the relevant pending bit would be set up, and generates an interrupt if the corresponding interrupt is enabled. The flow-process diagram is showed in Block Diagram section.

During a DMA transmission, the configuration could be obtained via the Configuration Register. And, behind the address of the config register in DDR or SRAM, there are some registers including other information of a DMA transmission. The

structure chart is showed in Block Diagram section. Also, other information of a transferring data can be obtained by reading the Current Source Address Register, Current Destination Address Register and Byte Counter Left Register. The configuration must be word-aligning.

The transferring data would be paused when setting up the relevant Pause Register, if coming up emergency. And the pausing data could be presumable when set 0 to the same bit in Pause Register.

## 3.11.3. DMA Register List

| Module Name | Base Address |
|---|---|
| DMA | 0x01C02000 |

| Register Name | Offset | Description |
|---|---|---|
| DMA_IRQ_EN_REG | 0x0 | DMA IRQ Enable Register |
| DMA_IRQ_PEND_REG | 0x10 | DMA IRQ Pending Register |
| DMA_AUTO_GATE_REG | 0x20 | DMA Auto Gating Register |
| DMA_SECURE_REG | 0x28 | DMA Security Register |
| DMA_STA_REG | 0x30 | DMA Status Register |
| DMA_EN_REG | 0x100+N*0x40 | DMA Channel Enable Register (N=0~7) |
| DMA_PAU_REG | 0x100+N*0x40+0x4 | DMA Channel Pause Register (N=0~7) |
| DMA_DESC_ADDR_REG | 0x100+N*0x40+0x8 | DMA Channel Start Address Register (N=0~7) |
| DMA_CFG_REG | 0x100+N*0x40+0xC | DMA Channel Configuration     Register (N=0~7) |
| DMA_CUR_SRC_REG | 0x100+N*0x40+0x10 | DMA Channel Current Source Register (N=0~7) |
| DMA_CUR_DEST_REG | 0x100+N*0x40+0x14 | DMA Channel Current Destination Register (N=0~7) |
| DMA_BCNT_LEFT_REG | 0x100+N*0x40+0x18 | DMA Channel Byte Counter Left Register (N=0~7) |
| DMA_PARA_REG | 0x100+N*0x40+0x1C | DMA Channel Parameter Register (N=0~7) |

## 3.11.4.  DMA Register Description

### 3.11.4.1.  DMA IRQ Enable Register (Default Value: 0x00000000)

| Offset: 0x0000 | | | Register Name: **DMA_IRQ_EN_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | / | / | / |
| 30 | R/W | 0x0 | DMA7_QUEUE_IRQ_EN<br>DMA 7 Queue End Transfer Interrupt Enable.<br>0: Disable, 1: Enable. |
| 29 | R/W | 0x0 | DMA7_PKG_IRQ_EN<br>DMA 7 Package End Transfer Interrupt Enable.<br>0: Disable, 1: Enable. |
| 28 | R/W | 0x0 | DMA7_HLAF_IRQ_EN<br>DMA 7 Half Package Transfer Interrupt Enable.<br>0: Disable, 1: Enable. |
| 27 | / | / | / |
| 26 | R/W | 0x0 | DMA6_QUEUE_IRQ_EN<br>DMA 6 Queue End Transfer Interrupt Enable.<br>0: Disable, 1: Enable. |
| 25 | R/W | 0x0 | DMA6_PKG_IRQ_EN<br>DMA 6 Package End Transfer Interrupt Enable.<br>0: Disable, 1: Enable. |
| 24 | R/W | 0x0 | DMA6_HLAF_IRQ_EN<br>DMA 6 Half Package Transfer Interrupt Enable.<br>0: Disable, 1: Enable. |
| 23 | / | / | / |
| 22 | R/W | 0x0 | DMA5_QUEUE_IRQ_EN<br>DMA 5 Queue End Transfer Interrupt Enable.<br>0: Disable, 1: Enable. |
| 21 | R/W | 0x0 | DMA5_PKG_IRQ_EN<br>DMA 5 Package End Transfer Interrupt Enable.<br>0: Disable, 1: Enable. |
| 20 | R/W | 0x0 | DMA5_HLAF_IRQ_EN<br>DMA 5 Half package Transfer Interrupt Enable.<br>0: Disable, 1: Enable. |
| 19 | / | / | / |
| 18 | R/W | 0x0 | DMA4_QUEUE_IRQ_EN<br>DMA 4 Queue End Transfer Interrupt Enable.<br>0: Disable, 1: Enable. |
| 17 | R/W | 0x0 | DMA4_PKG_IRQ_EN<br>DMA 4 Package End Transfer Interrupt Enable. |

| | | | 0: Disable, 1: Enable. |
|---|---|---|---|
| 16 | R/W | 0x0 | DMA4_HLAF_IRQ_EN |
| | | | DMA 4 Half Package Transfer Interrupt Enable. |
| | | | 0: Disable, 1: Enable. |
| 15 | / | / | / |
| 14 | R/W | 0x0 | DMA3_QUEUE_IRQ_EN |
| | | | DMA 3 Queue End Transfer Interrupt Enable. |
| | | | 0: Disable, 1: Enable. |
| 13 | R/W | 0x0 | DMA3_PKG_IRQ_EN |
| | | | DMA 3 Package End Transfer Interrupt Enable. |
| | | | 0: Disable, 1: Enable. |
| 12 | R/W | 0x0 | DMA3_HLAF_IRQ_EN |
| | | | DMA 3 Half Package Transfer Interrupt Enable. |
| | | | 0: Disable, 1: Enable. |
| 11 | / | / | / |
| 10 | R/W | 0x0 | DMA2_QUEUE_IRQ_EN |
| | | | DMA 2 Queue End Transfer Interrupt Enable. |
| | | | 0: Disable, 1: Enable. |
| 9 | R/W | 0x0 | DMA2_PKG_IRQ_EN |
| | | | DMA 2 Package End Transfer Interrupt Enable. |
| | | | 0: Disable, 1: Enable. |
| 8 | R/W | 0x0 | DMA2_HLAF_IRQ_EN |
| | | | DMA 2 Half Package Transfer Interrupt Enable. |
| | | | 0: Disable, 1: Enable. |
| 7 | / | / | / |
| 6 | R/W | 0x0 | DMA1_QUEUE_IRQ_EN |
| | | | DMA 1 Queue End Transfer Interrupt Enable. |
| | | | 0: Disable, 1: Enable. |
| 5 | R/W | 0x0 | DMA1_PKG_IRQ_EN |
| | | | DMA 1 Package End Transfer Interrupt Enable. |
| | | | 0: Disable, 1: Enable. |
| 4 | R/W | 0x0 | DMA1_HLAF_IRQ_EN |
| | | | DMA 1 Half Package Transfer Interrupt Enable. |
| | | | 0: Disable, 1: Enable. |
| 3 | / | / | / |
| 2 | R/W | 0x0 | DMA0_QUEUE_IRQ_EN |
| | | | DMA 0 Queue End Transfer Interrupt Enable. |
| | | | 0: Disable, 1: Enable. |
| 1 | R/W | 0x0 | DMA0_PKG_IRQ_EN |
| | | | DMA 0 Package End Transfer Interrupt Enable. |
| | | | 0: Disable, 1: Enable. |
| 0 | R/W | 0x0 | DMA0_HLAF_IRQ_EN |
| | | | DMA 0 Half Package Transfer Interrupt Enable. |
| | | | 0: Disable, 1: Enable |

### 3.11.4.2. DMA IRQ Pending Status Register (Default Value: 0x00000000)

| Offset:0x10 | | | Register Name: **DMA_IRQ_PEND_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | / | / | / |
| 30 | R/W | 0x0 | DMA7_QUEUE_IRQ_PEND. DMA 7 Queue End Transfer Interrupt Pending. Set 1 to the bit will clear it. 0: No effect, 1: Pending. |
| 29 | R/W | 0x0 | DMA7_PKG_IRQ_ PEND DMA 7 Package End Transfer Interrupt Pending. Set 1 to the bit will clear it. 0: No effect, 1: Pending. |
| 28 | R/W | 0x0 | DMA7_HLAF_IRQ_PEND. DMA 7 Half Package Transfer Interrupt Pending. Set 1 to the bit will clear it. 0: No effect, 1: Pending. |
| 27 | / | / | / |
| 26 | R/W | 0x0 | DMA6_QUEUE_IRQ_PEND. DMA 6 Queue End Transfer Interrupt Pending. Set 1 to the bit will clear it. 0: No effect, 1: Pending. |
| 25 | R/W | 0x0 | DMA6_PKG_IRQ_ PEND DMA 6 Package End Transfer Interrupt Pending. Set 1 to the bit will clear it. 0: No effect, 1: Pending. |
| 24 | R/W | 0x0 | DMA6_HLAF_IRQ_PEND. DMA 6 Half Package Transfer Interrupt Pending. Set 1 to the bit will clear it. 0: No effect, 1: Pending. |
| 23 | / | / | / |
| 22 | R/W | 0x0 | DMA5_QUEUE_IRQ_PEND. DMA 5 Queue End Transfer Interrupt Pending. Set 1 to the bit will clear it. 0: No effect, 1: Pending. |
| 21 | R/W | 0x0 | DMA5_PKG_IRQ_ PEND DMA 5 Package End Transfer Interrupt Pending. Set 1 to the bit will clear it. 0: No effect, 1: Pending. |
| 20 | R/W | 0x0 | DMA5_HLAF_IRQ_PEND. DMA 5 Half Package Transfer Interrupt Pending. Set 1 to the bit will clear it. 0: No effect, 1: Pending. |
| 19 | / | / | / |
| 18 | R/W | 0x0 | DMA4_QUEUE_IRQ_PEND. DMA 4 Queue End Transfer Interrupt Pending. Set 1 to the bit will clear it. 0: No effect, 1: Pending. |
| 17 | R/W | 0x0 | DMA4_PKG_IRQ_ PEND DMA 4 Package End Transfer Interrupt Pending. Set 1 to the bit will clear it. 0: No effect, 1: Pending. |
| 16 | R/W | 0x0 | DMA4_HLAF_IRQ_PEND. DMA 4 Half Package Transfer Interrupt Pending. Set 1 to the bit will clear it. 0: No effect, 1: Pending. |

| 15 | / | / | / |
|---|---|---|---|
| 14 | R/W | 0x0 | DMA3_QUEUE_IRQ_PEND.<br>DMA 3 Queue End Transfer Interrupt Pending. Set 1 to the bit will clear it.<br>0: No effect, 1: Pending. |
| 13 | R/W | 0x0 | DMA3_PKG_IRQ_ PEND<br>DMA 3 Package End Transfer Interrupt Pending. Set 1 to the bit will clear it.<br>0: No effect, 1: Pending. |
| 12 | R/W | 0x0 | DMA3_HLAF_IRQ_PEND.<br>DMA 3 Half Package Transfer Interrupt Pending. Set 1 to the bit will clear it.<br>0: No effect, 1: Pending. |
| 11 | / | / | / |
| 10 | R/W | 0x0 | DMA2_QUEUE_IRQ_PEND.<br>DMA 2 Queue End Transfer Interrupt Pending. Set 1 to the bit will clear it.<br>0: No effect, 1: Pending. |
| 9 | R/W | 0x0 | DMA2_PKG_IRQ_ PEND<br>DMA 2 Package End Transfer Interrupt Pending. Set 1 to the bit will clear it.<br>0: No effect, 1: Pending. |
| 8 | R/W | 0x0 | DMA2_HLAF_IRQ_PEND.<br>DMA 2 Half Package Transfer Interrupt Pending. Set 1 to the bit will clear it.<br>0: No effect, 1: Pending. |
| 7 | / | / | / |
| 6 | R/W | 0x0 | DMA1_QUEUE_IRQ_PEND.<br>DMA 1 Queue End Transfer Interrupt Pending. Set 1 to the bit will clear it.<br>0: No effect, 1: Pending. |
| 5 | R/W | 0x0 | DMA1_PKG_IRQ_ PEND<br>DMA 1 Package End Transfer Interrupt Pending. Set 1 to the bit will clear it.<br>0: No effect, 1: Pending. |
| 4 | R/W | 0x0 | DMA1_HLAF_IRQ_PEND.<br>DMA 1 Half Package Transfer Interrupt Pending. Set 1 to the bit will clear it.<br>0: No effect, 1: Pending. |
| 3 | / | / | / |
| 2 | R/W | 0x0 | DMA0_QUEUE_IRQ_PEND.<br>DMA 0 Queue End Transfer Interrupt Pending. Set 1 to the bit will clear it.<br>0: No effect, 1: Pending. |
| 1 | R/W | 0x0 | DMA0_PKG_IRQ_ PEND<br>DMA 0 Package End Transfer Interrupt Pending. Set 1 to the bit will clear it.<br>0: No effect, 1: Pending. |
| 0 | R/W | 0x0 | DMA0_HLAF_IRQ_PEND.<br>DMA 0 Half Package Transfer Interrupt Pending. Set 1 to the bit will clear it.<br>0: No effect, 1: Pending. |

### 3.11.4.3. DMA Auto Gating Register (Default Value: 0x00000000)

| Offset:0x20 | Register Name: **DMA_AUTO_GATE_REG** |
|---|---|

| Bit | R/W | Default/Hex | Description |
|---|---|---|---|
| 31:3 | / | / | / |
| 2 | R/W | 0x0 | DMA_MCLK_CIRCUIT. DMA MCLK interface circuit auto gating bit. 0: Auto gating enable 1: Auto gating disable. |
| 1 | R/W | 0x0 | DMA_COMMON_CIRCUIT. DMA common circuit auto gating bit. 0: Auto gating enable 1: Auto gating disable. |
| 0 | R/W | 0x0 | DMA_CHAN_CIRCUIT. DMA channel circuit auto gating bit. 0: Auto gating enable 1: Auto gating disable. |

**Note:**

When initializing DMA Controller,bit-2 should be set up.

### 3.11.4.4. DMA Security Register (Default Value: 0x00000000)

| Offset:0x28 | | | Register Name: **DMA_SECURE_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:8 | / | / | / |
| 7 | R/W | 0x0 | DMA7_SECURE. Indicating DMA 7 security. 0: Secure, 1: Non-secure. |
| 6 | R/W | 0x0 | DMA6_SECURE. Indicating DMA 6 security. 0: Secure, 1: Non-secure. |
| 5 | R/W | 0x0 | DMA5_SECURE. Indicating DMA 5 security. 0: Secure, 1: Non-secure. |
| 4 | R/W | 0x0 | DMA4_SECURE. Indicating DMA 4 security. 0: Secure, 1: Non-secure. |
| 3 | R/W | 0x0 | DMA3_SECURE. Indicating DMA 3 security. 0: Secure, 1: Non-secure. |
| 2 | R/W | 0x0 | DMA2_SECURE. Indicating DMA 2 security. 0: Secure, |

| | | | 1: Non-secure. |
|---|---|---|---|
| 1 | R/W | 0x0 | DMA1_SECURE. Indicating DMA 1 security. 0: Secure, 1: Non-secure. |
| 0 | R/W | 0x0 | DMA0_SECURE. Indicating DMA 0 security. 0: Secure, 1: Non-secure. |

### 3.11.4.5. DMA Status Register (Default Value: 0x00000000)

| Offset:0x30 | | | Register Name: **DMA_STA_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:8 | / | / | / |
| 7 | RO | 0x0 | DMA7_STATUS DMA Channel 7 Status. 0: Idle, 1: Busy. |
| 6 | RO | 0x0 | DMA6_STATUS DMA Channel 6 Status. 0: Idle, 1: Busy. |
| 5 | RO | 0x0 | DMA5_STATUS DMA Channel 5 Status. 0: Idle, 1: Busy. |
| 4 | RO | 0x0 | DMA4_STATUS DMA Channel 4 Status. 0: Idle, 1: Busy. |
| 3 | RO | 0x0 | DMA3_STATUS DMA Channel 3 Status. 0: Idle, 1: Busy. |
| 2 | RO | 0x0 | DMA2_STATUS DMA Channel 2 Status. 0: Idle, 1: Busy. |
| 1 | RO | 0x0 | DMA1_STATUS DMA Channel 1 Status. 0: Idle, 1: Busy. |
| 0 | RO | 0x0 | DMA0_STATUS DMA Channel 0 Status. 0: Idle, 1: Busy. |

### 3.11.4.6. DMA Channel Enable Register (Default Value: 0x00000000)

| Offset: 0x100+N*0x40+0x0(N=0~7) | | | Register Name: **DMA_EN_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:1 | / | / | / |
| 0 | R/W | 0x0 | DMA_EN. DMA Channel Enable 0: Disable, 1: Enable. |

### 3.11.4.7. DMA Channel Pause Register (Default Value: 0x00000000)

| Offset: 0x100+N*0x40+0x4(N=0~7) | | | Register Name: **DMA_ PAU_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:1 | / | / | / |
| 0 | R/W | 0x0 | DMA_PAUSE. Pausing DMA Channel Transfer Data. 0: Resume Transferring, 1: Pause Transferring. |

### 3.11.4.8. DMA Channel Descriptor Address Register

| Offset: :0x100+N*0x40+0x8(N=0~7) | | | Register Name: **DMA_DESC_ADDR_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R/W | x | DMA_DESC_ADDR DMA Channel Descriptor Address. |

### 3.11.4.9. DMA Channel Configuration Register (Default Value: 0x00000000)

| Offset: 0x100+N*0x40+0xC(N=0~7) | | | Register Name: **DMA_CFG_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:27 | / | / | / |
| 26:25 | RO | 0x0 | DMA_DEST_DATA_WIDTH. DMA Destination Data Width. 00: 8-bit 01: 16-bit 10: 32-bit 11: / |
| 24:23 | RO | 0x0 | DMA_DEST_BST_LEN. DMA Destination Burst Length. 00: 1 01: / |

| 10: 8 |
| --- |
| 11: / |

| 22:21 | RO | 0x0 | DMA_ADDR_MODE. |
| --- | --- | --- | --- |
| | | | DMA Destination Address Mode |
| | | | 0x0: Linear Mode |
| | | | 0x1: IO Mode |
| | | | 0x2: / |
| | | | 0x3: / |
| 20:16 | RO | 0x0 | DMA_DEST_DRQ_TYPE. |
| | | | DMA Destination DRQ Type |
| | | | The details in DRQ Type and Port Corresponding Relation. |
| 15:11 | / | / | / |
| 10:9 | RO | 0x0 | DMA_SRC_DATA_WIDTH. |
| | | | DMA Source Data Width. |
| | | | 00: 8-bit |
| | | | 01: 16-bit |
| | | | 10: 32-bit |
| | | | 11: / |
| 8:7 | RO | 0x0 | DMA_SRC_BST_LEN. |
| | | | DMA Source Burst Length. |
| | | | 00: 1 |
| | | | 01: / |
| | | | 10: 8 |
| | | | 11: / |
| 6:5 | RO | 0x0 | DMA_SRC_ADDR_MODE. |
| | | | DMA Source Address Mode |
| | | | 0x0: Linear Mode |
| | | | 0x1: IO Mode |
| | | | 0x2: / |
| | | | 0x3: / |
| 4:0 | RO | 0x0 | DMA_SRC_DRQ_TYPE. |
| | | | DMA Source DRQ Type |
| | | | The details in DRQ Type and Port Corresponding Relation. |

**Note1:**

If the DRQ type is dram, then, the corresponding burst length will be fixed, and the options will be invalid.

**Note2:**

The address of the *DMA Channel Configuration Register* must be word-aligned.

**Note3:**

If the DRQ type is SRAM, then source address of destination address must be word-aligned.

### 3.11.4.10.  DMA Channel Current Source Address Register (Default Value: 0x00000000)

| Offset: 0x100+N*0x40+0x10 (N=0~7) | Register Name: **DMA_CUR_SRC_REG** |
| --- | --- |

| Bit | R/W | Default/Hex | Description |
|---|---|---|---|
| 31:0 | RO | 0x0 | DMA_CUR_SRC. <br> DMA Channel Current Source Address, read only. |

**Note:**

The address of the *DMA Channel Current Source Address Register* must be word-aligned.

### 3.11.4.11.  DMA Channel Current Destination Address Register (Default Value: 0x00000000)

| Offset: 0x100+N*0x40+0x14(N=0~7) | | | Register Name: **DMA_CUR_DEST_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R/W | 0 | DMA_CUR_DEST. <br> DMA Channel Current Destination Address, read only. |

**Note:**

The address of the **DMA Channel Current Destination Address Register** must be word-aligned.

### 3.11.4.12.  DMA Channel Byte Counter Left Register (Default Value: 0x00000000)

| Offset: 0x100+N*0x40+0x18(N=0~7) | | | Register Name: **DMA_BCNT_LEFT_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:25 | / | / | / |
| 24:0 | RO | 0x0 | DMA_BCNT_LEFT. <br> DMA Channel Byte Counter Left, read only. |

**Note:**

The address of the **DMA Channel Byte Counter Left Register** must be word-aligned.

### 3.11.4.13.  DMA Channel Parameter Register (Default Value: 0x00000000)

| Offset: 0x100+N*0x40+0x1C(N=0~7) | | | Register Name: **DMA_PARA_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:16 | / | / | / |
| 15:8 | RO | 0x0 | DATA_BLK_SIZE. <br> Data Block Size N. |
| 7:0 | RO | 0x0 | WAIT_CYC. <br> Wait Clock Cycles n. |

**Note1:**

The number of data block size usually depends on the capacity of the device's FIFO in the practical application.

**Note2:**

The data block size must be multiple of burst*width (byte). For example: if burst is 4 and the width is 32-bit, so the data block size must be m*16(byte), i.e. N = m * 16.

**Note3:**

When DMA controller has completed transferring N bytes data, and waiting n clock cycles to check the DRQ signal.

**Note4:**

This register is only effective to devices, and the Data Block Size N should be 0 if it is less than 32.

## 3.12. GIC

### 3.12.1. Interrupt Source

| Interruptnumber | Interrupt Source | Description |
|---|---|---|
| 0 | SGI 0 | SGI 0 interrupt |
| 1 | SGI 1 | SGI 1 interrupt |
| 2 | SGI 2 | SGI 2 interrupt |
| 3 | SGI 3 | SGI 3 interrupt |
| 4 | SGI 4 | SGI 4 interrupt |
| 5 | SGI 5 | SGI 5 interrupt |
| 6 | SGI 6 | SGI 6 interrupt |
| 7 | SGI 7 | SGI 7 interrupt |
| 8 | SGI 8 | SGI 8 interrupt |
| 9 | SGI 9 | SGI 9 interrupt |
| 10 | SGI 10 | SGI 10 interrupt |
| 11 | SGI 11 | SGI 11 interrupt |
| 12 | SGI 12 | SGI 12 interrupt |
| 13 | SGI 13 | SGI 13 interrupt |
| 14 | SGI 14 | SGI 14 interrupt |
| 15 | SGI 15 | SGI 15 interrupt |
| 16 | PPI 0 | PPI 0 interrupt |
| 17 | PPI 1 | PPI 1 interrupt |
| 18 | PPI 2 | PPI 2 interrupt |
| 19 | PPI 3 | PPI 3 interrupt |
| 20 | PPI 4 | PPI 4 interrupt |
| 21 | PPI 5 | PPI 5 interrupt |
| 22 | PPI 6 | PPI 6 interrupt |
| 23 | PPI 7 | PPI 7 interrupt |
| 24 | PPI 8 | PPI 8 interrupt |
| 25 | PPI 9 | PPI 9 interrupt |
| 26 | PPI 10 | PPI 10 interrupt |
| 27 | PPI 11 | PPI 11 interrupt |
| 28 | PPI 12 | PPI 12 interrupt |
| 29 | PPI 13 | PPI 13 interrupt |
| 30 | PPI 14 | PPI 14 interrupt |
| 31 | PPI 15 | PPI 15 interrupt |
| 32 | UART 0 | UART 0 interrupt |
| 33 | UART 1 | UART 1 interrupt |

| 34 | UART 2 | UART 2 interrupt |
|---|---|---|
| 35 | UART 3 | UART 3 interrupt |
| 36 | UART 4 | UART 4 interrupt |
| 37 | | |
| 38 | TWI 0 | TWI 0 interrupt |
| 39 | TWI 1 | TWI 1 interrupt |
| 40 | TWI 2 | TWI 2 interrupt |
| 41 | | |
| 42 | | |
| 43 | | |
| 44 | OWA | OWA interrupt |
| 45 | DAUDIO-0 | DAUDIO-0 interrupt |
| 46 | DAUDIO-1 | DAUDIO-1 interrupt |
| 47 | PB_EINT | PB_EINT interrupt |
| 48 | | |
| 49 | PG_EINT | PG_EINT interrupt |
| 50 | Timer 0 | Timer 0 interrupt |
| 51 | Timer 1 | Timer 1 interrupt |
| 52 | | |
| 53 | | |
| 54 | | |
| 55 | | |
| 56 | | |
| 57 | Watchdog | Watchdog interrupt |
| 58 | | |
| 59 | | |
| 60 | | |
| 61 | | |
| 62 | | |
| 63 | | |
| 64 | External NMI | External Non-Mask Interrupt |
| 65 | R_timer 0 | R_timer 0 interrupt |
| 66 | R_timer 1 | R_timer 1 interrupt |
| 67 | | |
| 68 | R_watchdog | R_watchdog interrupt |
| 69 | R_CIR-RX | R_CIR-RX interrupt |
| 70 | R_UART | R_UART interrupt |
| 71 | R_RSB | R_RSB interrupt |
| 72 | | |
| 73 | R_TH | R_TH interrupt |
| 74 | R_LRADC | R_LRADC interrupt |
| 75 | | |
| 76 | R_TWI | R_TWI interrupt |
| 77 | R_PL_EINT | R_PL_EINT interrupt |

| 78 | R_TWD | R_TWD interrupt |
|-----|-------|-----------------|
| 79 | | |
| 80 | | |
| 81 | M-box | M-box interrupt |
| 82 | DMA | DMA channel interrupt |
| 83 | HS Timer | HS Timer interrupt |
| 84 | | |
| 85 | | |
| 86 | | |
| 87 | | |
| 88 | SMC | SMC interrupt |
| 89 | | |
| 90 | VE | VE interrupt |
| 91 | | |
| 92 | SD/MMC 0 | SD/MMC Host Controller 0 interrupt |
| 93 | SD/MMC 1 | SD/MMC Host Controller 1 interrupt |
| 94 | SD/MMC 2 | SD/MMC Host Controller 2 interrupt |
| 95 | | |
| 96 | | |
| 97 | SPI 0 | SPI 0 interrupt |
| 98 | SPI 1 | SPI 1 interrupt |
| 99 | | |
| 100 | | |
| 101 | | |
| 102 | NAND | NAND Flash Controller interrupt |
| 103 | USB-DRD | USB-DRD interrupt |
| 104 | USB-EHCI0 | USB-EHCI0 interrupt |
| 105 | USB-OHCI0 | USB-OHCI0 interrupt |
| 106 | USB-EHCI1 | USB-EHCI1 interrupt |
| 107 | | |
| 108 | | |
| 109 | | |
| 110 | | |
| 111 | | |
| 112 | SS | SS interrupt |
| 113 | | |
| 114 | EMAC | EMAC interrupt |
| 115 | MIPI_CSI | MIPI_CSI interrupt |
| 116 | CSI | CSI interrupt |
| 117 | CSI_CCI | CSI_CCI interrupt |
| 118 | LCD0 | LCD0 Controller interrupt |
| 119 | LCD1 | LCD1 Controller interrupt |
| 120 | HDMI | HDMI interrupt |
| 121 | MIPI DSI | MIPI DSI interrupt |

| 122 |  |  |
| --- | --- | --- |
| 123 | DE_IRQ0 | RT-mixer and RTWB interrupt |
| 124 | DE_IRQ1 | Rotation interrupt |
| 125 |  |  |
| 126 |  |  |
| 127 |  |  |
| 128 |  |  |
| 129 | GPU | GPU interrupt |
| 130 | TDM | TDM interrupt |
| 131 | DAUDIO-2 | DAUDIO-2 interrupt |
| 132 | PH_EINT | PH_EINT interrupt |
| 133 |  |  |
| 134 |  |  |
| 135 |  |  |
| 136 |  |  |
| 137 |  |  |
| 138 |  |  |
| 139 |  |  |
| 140 |  |  |
| 141 |  |  |
| 142 |  |  |
| 143 |  |  |
| 144 |  |  |
| 145 |  |  |
| 146 |  |  |
| 147 |  |  |
| 148 |  |  |
| 159 |  |  |
| 150 |  |  |
| 151 |  |  |
| 152 |  |  |
| 153 |  |  |
| 154 |  |  |
| 155 |  |  |
| 156 |  |  |
| 157 |  |  |
| 158 |  |  |
| 159 |  |  |
| 160 | Cluster0_CTI0 | Cluster0_CTI0 interrupt |
| 161 | Cluster0_CTI1 | Cluster0_CTI1 interrupt |
| 162 | Cluster0_CTI2 | Cluster0_CTI2 interrupt |
| 163 | Cluster0_CTI3 | Cluster0_CTI3 interrupt |
| 164 | Cluster0_PMU0 | Cluster0_PMU0 interrupt |
| 165 | Cluster0_PMU1 | Cluster0_PMU1 interrupt |

| 166 | Cluster0_PMU2 | Cluster0_PMU2 interrupt |
|-----|---------------|-------------------------|
| 167 | Cluster0_PMU3 | Cluster0_PMU3 interrupt |
| 168 | Cluster0_AXIERR | Cluster0_AXIERR interrupt |
| 169 | Cluster0_COMMTX0 | Cluster0_COMMTX0 interrupt |
| 170 | Cluster0_COMMTX1 | Cluster0_COMMTX1 interrupt |
| 171 | Cluster0_COMMTX2 | Cluster0_COMMTX2 interrupt |
| 172 | Cluster0_COMMTX3 | Cluster0_COMMTX3 interrupt |
| 173 | Cluster0_COMMRX0 | Cluster0_COMMRX0 interrupt |
| 174 | Cluster0_COMMRX1 | Cluster0_COMMRX1 interrupt |
| 175 | Cluster0_COMMRX2 | Cluster0_COMMRX2 interrupt |
| 176 | Cluster0_COMMRX3 | Cluster0_COMMRX3 interrupt |
| 177 | Cluster0_AXI_WR | Cluster0_AXI_WR interrupt |
| 178 | Cluster0_AXI_RD | Cluster0_AXI_RD interrupt |
| 179 | Cluster1_CTI0 | Cluster1_CTI0 interrupt |
| 180 | Cluster1_CTI1 | Cluster1_CTI1 interrupt |
| 181 | Cluster1_CTI2 | Cluster1_CTI2 interrupt |
| 182 | Cluster1_CTI3 | Cluster1_CTI3 interrupt |
| 183 | Cluster1_PMU0 | Cluster1_PMU0 interrupt |
| 184 | Cluster1_PMU1 | Cluster1_PMU1 interrupt |
| 185 | Cluster1_PMU2 | Cluster1_PMU2 interrupt |
| 186 | Cluster1_PMU3 | Cluster1_PMU3 interrupt |
| 187 | Cluster1_AXIERR | Cluster1_AXIERR interrupt |
| 188 | Cluster1_COMMTX0 | Cluster1_COMMTX0 interrupt |
| 189 | Cluster1_COMMTX1 | Cluster1_COMMTX1 interrupt |
| 190 | Cluster1_COMMTX2 | Cluster1_COMMTX2 interrupt |
| 191 | Cluster1_COMMTX3 | Cluster1_COMMTX3 interrupt |
| 192 | Cluster1_COMMRX0 | Cluster1_COMMRX0 interrupt |
| 193 | Cluster1_COMMRX1 | Cluster1_COMMRX1 interrupt |
| 194 | Cluster1_COMMRX2 | Cluster1_COMMRX2 interrupt |
| 195 | Cluster1_COMMRX3 | Cluster1_COMMRX3 interrupt |
| 196 | Cluster1_AXI_WR | Cluster1_AXI_WR interrupt |
| 197 | Cluster1_AXI_RD | Cluster1_AXI_RD interrupt |
| 198 | CCI400_ERR | CCI400_ERR interrupt |
| 199 | CCI400_EVENTOF | CCI400_EVENTOF interrupt |

## 3.13. Message Box

### 3.13.1. Overview

Message Box provides an MSGBox-interrupt mechanism for on-chip processors intercommunication.

The MSGBox-interrupt mechanism allows the software to establish a communication channel between the two users through a set of registers and associated interrupt signals by sending or receiving messages.

The Message Box includes the following features:

- Two users for Message Box instance(User0 for CPUS and User1 for C0-CPUX/C1-CPUX)
- Eight Message Queues for the MSGBox instance
- Each of Queues could be configured as transmitter or receiver for user
- Two interrupts (one per user ) for the MSGBox instance
- Register polling for the MSGBox instance
- 32-bit message width
- Each message queue has four-message FIFO depth

### 3.13.2. Functionalities Description

#### 3.13.2.1. Typical Applications

Message Box is typically designed for making the on-chip processors interconnection be true. It could establish an interconnection channel between processors by configuring a set of Message Box registers. Each of Message Queues is bidirectional for users, that means, while a message queue is configured as a receiver for a user, it is a transmitter for the interconnectible user beside. If a processor would like to interconnect with the other processor, it should configure one or more Message Queues firstly. Although Message Box provides two interrupt mechanism to notice user to transmit or receive messages, the way check out queue FIFO full status is usually adopted before transmitting a message and receiving a message is still depended on the reception notification. The Message Box is usually applied as the below flow chart:
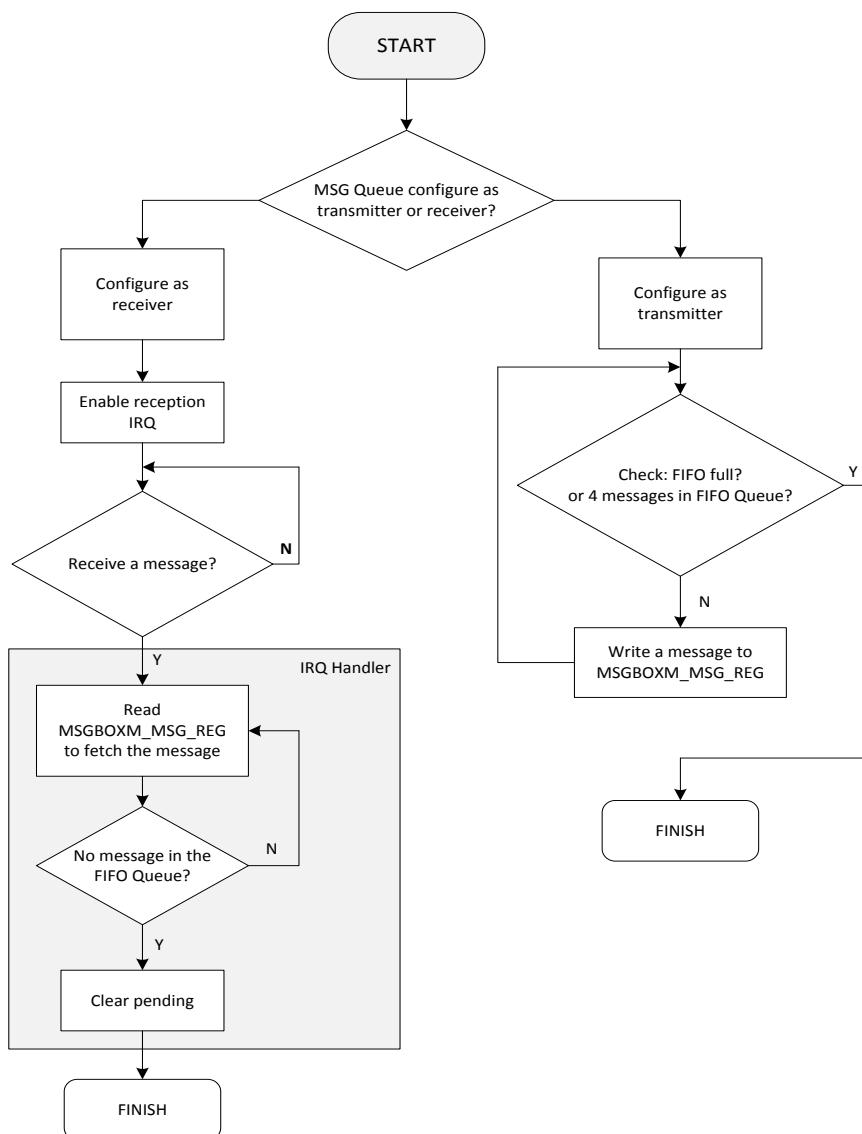
Figure 3-8. Message Box Typical Application Chart

### 3.13.2.2. Functional Block Diagram

# MSGBox
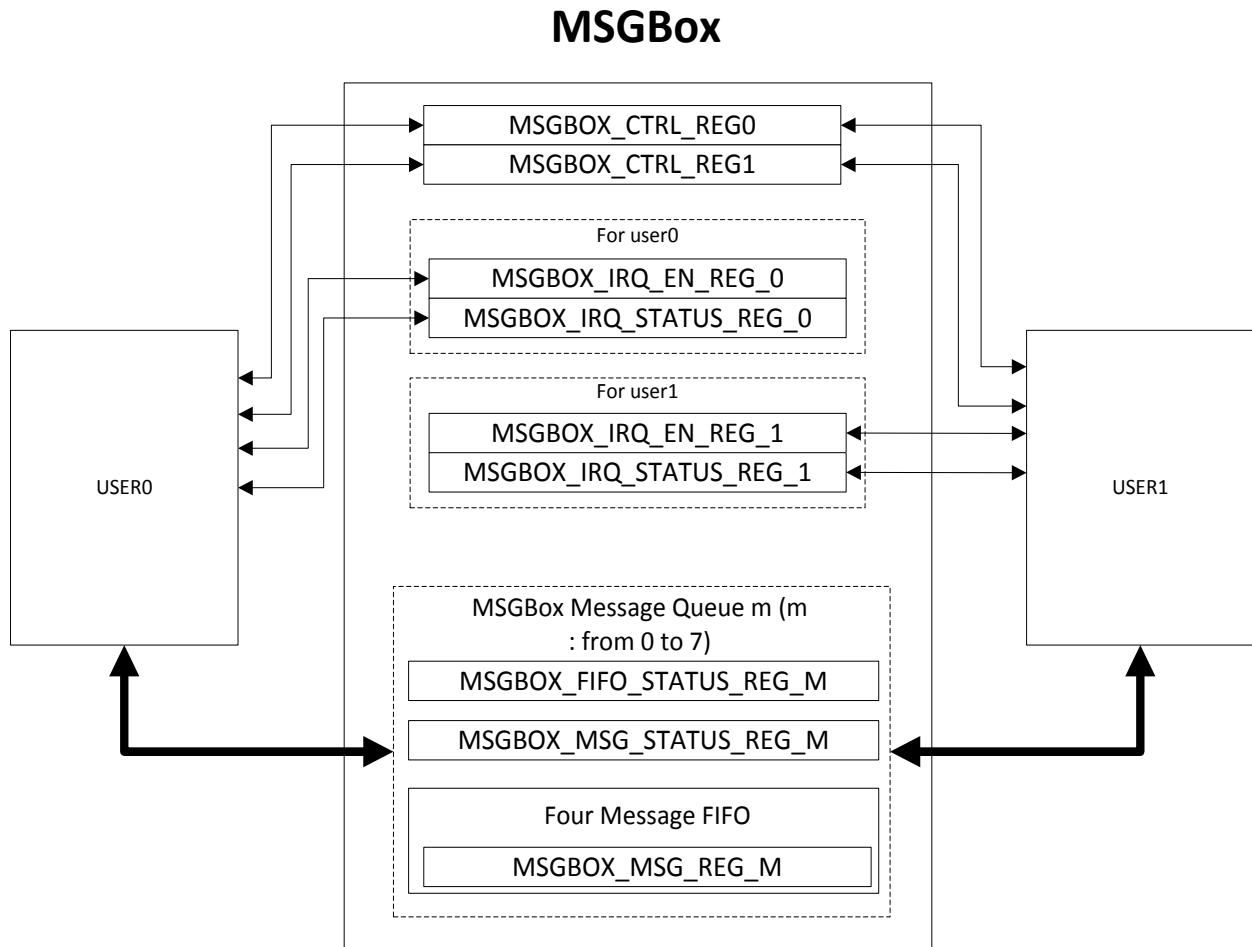


Figure 3-9. Message Box Functional Block Diagram

Message Box supports a set of registers for a processor to establish an interconnection channel with the others. The processor determines message queue numbers for interconnection and the used queues to be transmitter or receiver for itself and the interconnectible one. Every queue has a *MSGBox FIFO Status Register* for processor to check out queue FIFO full status and a *MSGBox Message Status Register* for processor to check out message numbers in queue FIFO. Otherwise, every queue has a corresponding IRQ status bit and a corresponding IRQ enable bit, which used for requesting an interrupt.

### 3.13.3. Operation Principle

#### 3.13.3.1. Message Box clock gating and software reset

By default the Message Box clock gating is mask. When it is necessary to use Message Box, it's clock gating should be open in *Bus Clock Gating Register1* and then de-assert the software reset in *Bus Software Reset Register1* on CCU module. If it is no need to use Message Box, both the gating bit and software reset bit should set 0.

#### 3.13.3.2. Message Queue Assignment

When a processor needs to transmit or receive a message from the other one, it should configure the Message Queue assignment for the other one and itself. *MSGBOX_CTRL_REG0* and *MSGBOX_CTRL_REG1* hold the eight Message Queues assignment. For an instance, RECEPTION_MQ0 bit is set to 0 and TRANSMIT_MQ0 bit is set to 1, which means, user1 transmits messages and user0 receives them. Or RECEPTION_MQ0 bit and TRANSMIT_MQ0 bit are both set to 0, which means user0 transmits messages to itself.

#### 3.13.3.3. Interrupt request

Message Box provides Message reception and queue-not-full notification interrupt mechanism. For a Message Queue configured as transmitter for a user, this queue transmit pending bit will always be set to 1 of this user if it is not full. For a Message Queue configured as receiver for a user, this queue reception pending bit will be set to 1 for this user only if it receives a new message. For example, Message Queue0 is configured as a transmitter for user0 and a receiver for user1. The thing Message Queue0 is not full always makes TRANSMIT_MQ0_IRQ_PEND bit set to 1. If TRANSMIT_MQ0_IRQ_EN bit is set to 1, user0 will request a queue-not-full interrupt. When Message Queue0 has received a new message, RECEPTION_MQ0_IRQ_PEND bit would be set to 1 and user1 will request a new message reception interrupt if RECEPTION_MQ0_IRQ_EN bit is set to 1. *MSGBox IRQ Status Register u (u=0, 1)* hold the IRQ status for user0 and user1. *MSGBox IRQ Enable Register u (u=0, 1)* determine whether the user could request the interrupt or not.

#### 3.13.3.4. Transmit and receive messages

Every Message Queue has a couple of private registers for query: *MSGBox Message Status Register* and *MSGBox FIFO Status Register* and a store register bridged to Message Queue FIFO: *MSGBox Message Queue Register.* MSGBox Message Status Register records present message number in the Message Queue. MSGBox FIFO Status Register indicates whether the Message Queue is full obviously. MSGBox Message Queue Register stores the next to be read message of the message FIFO queue or the message to be written into the queue FIFO. The thing that queue is not full

usually indicates that you could write messages into the queue FIFO and that there is one or more message in the queue FIFO indicates that you could read messages from the queue FIFO.

Writing a message into the queue FIFO realizes a transmission and reading a message makes a reception. You could transmit messages by writing messages to MSGBox Message Queue Register continuously or receive messages by reading MSGBox Message Queue Register continuously. The wiring or reading operation could be continuous means it's no need to make a delay between operations.

## 3.13.4. Message Box Register List

| Module Name | Base Address |
|---|---|
| MSGBOX | 0x01C17000 |

| Register Name | Offset | Description |
|---|---|---|
| MSGBOX_CTRL_REG0 | 0x0000 | Message Queue Attribute Control Register 0 |
| MSGBOX_CTRL_REG1 | 0x0004 | Message Queue Attribute Control Register 1 |
| MSGBOXU_IRQ_EN_REG | 0x0040+n*0x20 | IRQ Enable For User N(N=0,1) |
| MSGBOXU_IRQ_STATUS_REG | 0x0050+n*0x20 | IRQ Status For User N(N=0,1) |
| MSGBOXM_FIFO_STATUS_REG | 0x0100+N*0x4 | FIFO Status For Message Queue N(N = 0~7) |
| MSGBOXM_MSG_STATUS_REG | 0x0140+N*0x4 | Message Status For Message Queue N(N=0~7) |
| MSGBOXM_MSG_REG | 0x0180+N*0x4 | Message Register For Message Queue N(N=0~7) |

## 3.13.5. Message Box Register Description

### 3.13.5.1. MSGBox Control Register 0(Default Value: 0x10101010)

| Offset: 0x00 | | | Register Name: MSGBOX_CTRL_REG0 |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:29 | / | / | / |
| 28 | R/W | 0x1 | TRANSMIT_MQ3. Message Queue 3 is a Transmitter of user u. 0: user0 1: user1 |
| 27:25 | / | / | / |

| 24 | R/W | 0x0 | RECEPTION_MQ3.<br>Message Queue 3 is a Receiver of user u.<br>0: user0<br>1: user1 |
|---|---|---|---|
| 23:21 | / | / | / |
| 20 | R/W | 0x1 | TRANSMIT_MQ2.<br>Message Queue 2 is a Transmitter of user u.<br>0: user0<br>1: user1 |
| 19:17 | / | / | / |
| 16 | R/W | 0x0 | RECEPTION_MQ2.<br>Message Queue 2 is a Receiver of user u.<br>0: user0<br>1: user1 |
| 15:13 | / | / | / |
| 12 | R/W | 0x1 | TRANSMIT_MQ1<br>Message Queue 1 is a Transmitter of user u.<br>0: user0<br>1: user1 |
| 11:9 | / | / | / |
| 8 | R/W | 0x0 | RECEPTION_MQ1.<br>Message Queue 1 is a Receiver of user u.<br>0: user0<br>1: user1 |
| 7:5 | / | / | / |
| 4 | R/W | 0x1 | TRANSMIT_MQ0.<br>Message Queue 0 is a Transmitter of user u.<br>0: user0<br>1: user1 |
| 3:1 | / | / | / |
| 0 | R/W | 0x0 | RECEPTION_MQ0.<br>Message Queue 0 is a Receiver of user u.<br>0: user0<br>1: user1 |

### 3.13.5.2. MSGBox Control Register 1(Default Value: 0x10101010)

| Offset: 0x04 | | | Register Name: **MSGBOX_CTRL_REG1** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:29 | / | / | / |
| 28 | R/W | 0x1 | TRANSMIT_MQ7.<br>Message Queue 7 is a Transmitter of user u.<br>0: user0<br>1: user1 |

| 27:25 | / | / | / |
|---|---|---|---|
| 24 | R/W | 0x0 | RECEPTION_MQ7.<br>Message Queue 7 is a Receiver of user u.<br>0: user0<br>1: user1 |
| 23:21 | / | / | / |
| 20 | R/W | 0x1 | TRANSMIT_MQ6.<br>Message Queue 6 is a Transmitter of user u.<br>0: user0<br>1: user1 |
| 19:17 | / | / | / |
| 16 | R/W | 0x0 | RECEPTION_MQ6.<br>Message Queue 6 is a Receiver of user u.<br>0: user0<br>1: user1 |
| 15:13 | / | / | / |
| 12 | R/W | 0x1 | TRANSMIT_MQ5<br>Message Queue 5 is a Transmitter of user u.<br>0: user0<br>1: user1 |
| 11:9 | / | / | / |
| 8 | R/W | 0x0 | RECEPTION_MQ5.<br>Message Queue 5 is a Receiver of user u.<br>0: user0<br>1: user1 |
| 7:5 | / | / | / |
| 4 | R/W | 0x1 | TRANSMIT_MQ4.<br>Message Queue 4 is a Transmitter of user u.<br>0: user0<br>1: user1 |
| 3:1 | / | / | / |
| 0 | R/W | 0x0 | RECEPTION_MQ4.<br>Message Queue 4 is a Receiver of user u.<br>0: user0<br>1: user1 |

### 3.13.5.3. MSGBox IRQ Enable Register u(u=0,1)( Default Value: 0x00000000)

| Offset:0x40+N*0x20 (N=0,1) | | | Register Name: **MSGBOXU_IRQ_EN_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:16 | / | / | / |
| 15 | R/W | 0x0 | TRANSMIT_MQ7_IRQ_EN.<br>0: Disable |

| | | | 1: Enable (It will Notify user u by interrupt when Message Queue 7 is not full.) |
|---|---|---|---|
| 14 | R/W | 0x0 | RECEPTION_MQ7_IRQ_EN. <br> 0: Disable <br> 1: Enable (It will notify user u by interrupt when Message Queue 7 has received a new message.) |
| 13 | R/W | 0x0 | TRANSMIT_MQ6_IRQ_EN. <br> 0: Disable <br> 1: Enable (It will Notify user u by interrupt when Message Queue 6 is not full.) |
| 12 | R/W | 0x0 | RECEPTION_MQ6_IRQ_EN. <br> 0: Disable <br> 1: Enable (It will notify user u by interrupt when Message Queue 6 has received a new message.) |
| 11 | R/W | 0x0 | TRANSMIT_MQ5_IRQ_EN. <br> 0: Disable <br> 1: Enable (It will Notify user u by interrupt when Message Queue 5 is not full.) |
| 10 | R/W | 0x0 | RECEPTION_MQ5_IRQ_EN. <br> 0: Disable <br> 1: Enable (It will notify user u by interrupt when Message Queue 5 has received a new message.) |
| 9 | R/W | 0x0 | TRANSMIT_MQ4_IRQ_EN. <br> 0: Disable <br> 1: Enable (It will Notify user u by interrupt when Message Queue 4 is not full.) |
| 8 | R/W | 0x0 | RECEPTION_MQ4_IRQ_EN. <br> 0: Disable <br> 1: Enable (It will notify user u by interrupt when Message Queue 4 has received a new message.) |
| 7 | R/W | 0x0 | TRANSMIT_MQ3_IRQ_EN. <br> 0: Disable <br> 1: Enable (It will Notify user u by interrupt when Message Queue 3 is not full.) |
| 6 | R/W | 0x0 | RECEPTION_MQ3_IRQ_EN. <br> 0: Disable <br> 1: Enable (It will notify user u by interrupt when Message Queue 3 has received a new message.) |
| 5 | R/W | 0x0 | TRANSMIT_MQ2_IRQ_EN. <br> 0: Disable <br> 1: Enable (It will Notify user u by interrupt when Message Queue 2 is not full.) |
| 4 | R/W | 0x0 | RECEPTION_MQ2_IRQ_EN. <br> 0: Disable <br> 1: Enable (It will notify user u by interrupt when     Message Queue 2 has |

| | | | received a new message.) |
|---|---|---|---|
| 3 | R/W | 0x0 | TRANSMIT_MQ1_IRQ_EN. <br> 0: Disable <br> 1: Enable (It will Notify user u by interrupt when Message Queue 1 is not full.) |
| 2 | R/W | 0x0 | RECEPTION_MQ1_IRQ_EN. <br> 0: Disable <br> 1: Enable (It will notify user u by interrupt when Message Queue 1 has received a new message.) |
| 1 | R/W | 0x0 | TRANSMIT_MQ0_IRQ_EN. <br> 0: Disable <br> 1: Enable (It will Notify user u by interrupt when Message Queue 0 is not full.) |
| 0 | R/W | 0x0 | RECEPTION_MQ0_IRQ_EN. <br> 0: Disable <br> 1: Enable (It will notify user u by interrupt when Message Queue 0 has received a new message.) |

### 3.13.5.4. MSGBox IRQ Status Register u(Default Value: 0x0000AAAA)

| Offset:0x50+N*0x20 (N=0,1) | | | Register Name: **MSGBOXU_IRQ_STATUS_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:16 | / | / | / |
| 15 | R/W | 0x1 | TRANSMIT_MQ7_IRQ_PEND. <br> 0: No effect, <br> 1: Pending. This bit will be pending for user u when Message Queue 7 is not full. Set one to this bit will clear it. |
| 14 | R/W | 0x0 | RECEPTION_MQ7_IRQ_PEND. <br> 0: No effect, <br> 1: Pending. This bit will be pending for user u when Message Queue 7 has received a new message. Set one to this bit will clear it. |
| 13 | R/W | 0x1 | TRANSMIT_MQ6_IRQ_PEND. <br> 0: No effect, <br> 1: Pending. This bit will be pending for user u when Message Queue 6 is not full. Set one to this bit will clear it. |
| 12 | R/W | 0x0 | RECEPTION_MQ6_IRQ_PEND. <br> 0: No effect, <br> 1: Pending. This bit will be pending for user u when Message Queue 6 has received a new message. Set one to this bit will clear it. |
| 11 | R/W | 0x1 | TRANSMIT_MQ5_IRQ_PEND. <br> 0: No effect, <br> 1: Pending. This bit will be pending for user u when Message Queue 5 is not full. Set one to this bit will clear it. |
| 10 | R/W | 0x0 | RECEPTION_MQ5_IRQ_PEND. |

| | | | |
|---|---|---|---|
| | | | 0: No effect,<br>1: Pending. This bit will be pending for user u when Message Queue 5 has received a new message. Set one to this bit will clear it. |
| 9 | R/W | 0x1 | TRANSMIT_MQ4_IRQ_PEND.<br>0: No effect,<br>1: Pending. This bit will be pending for user u when Message Queue 4 is not full. Set one to this bit will clear it. |
| 8 | R/W | 0x0 | RECEPTION_MQ4_IRQ_PEND.<br>0: No effect,<br>1: Pending. This bit will be pending for user u when Message Queue 4 has received a new message. Set one to this bit will clear it. |
| 7 | R/W | 0x1 | TRANSMIT_MQ3_IRQ_PEND.<br>0: No effect,<br>1: Pending. This bit will be pending for user u when Message Queue 3 is not full. Set one to this bit will clear it. |
| 6 | R/W | 0x0 | RECEPTION_MQ3_IRQ_PEND.<br>0: No effect,<br>1: Pending. This bit will be pending for user u when Message Queue 3 has received a new message. Set one to this bit will clear it. |
| 5 | R/W | 0x1 | TRANSMIT_MQ2_IRQ_PEND.<br>0: No effect,<br>1: Pending. This bit will be pending for user u when Message Queue 2 is not full. Set one to this bit will clear it. |
| 4 | R/W | 0x0 | RECEPTION_MQ2_IRQ_PEND.<br>0: No effect,<br>1: Pending. This bit will be pending for user u when Message Queue 2 has received a new message. Set one to this bit will clear it. |
| 3 | R/W | 0x1 | TRANSMIT_MQ1_IRQ_PEND.<br>0: No effect,<br>1: Pending. This bit will be pending for user u when Message Queue 1 is not full. Set one to this bit will clear it. |
| 2 | R/W | 0x0 | RECEPTION_MQ1_IRQ_PEND.<br>0: No effect,<br>1: Pending. This bit will be pending for user u when Message Queue 1 has received a new message. Set one to this bit will clear it. |
| 1 | R/W | 01 | TRANSMIT_MQ0_IRQ_PEND.<br>0: No effect,<br>1: Pending. This bit will be pending for user u when Message Queue 0 is not full. Set one to this bit will clear it. |
| 0 | R/W | 0x0 | RECEPTION_MQ0_IRQ_PEND.<br>0: No effect,<br>1: Pending. This bit will be pending for user u when Message Queue 0 has received a new message. Set one to this bit will clear it. |

### 3.13.5.5. MSGBox FIFO Status Register m(Default Value: 0x00000000)

| Offset:0x100+N*0x4 (N=0~7) | | | Register Name: **MSGBOXM_FIFO_STATUS_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31: 1 | / | / | / |
| 0 | RO | 0x0 | FIFO_FULL_FLAG. 0: The Message FIFO queue is not full (space is available), 1: The Message FIFO queue is full. This FIFO status register has the status related to the message queue. |

### 3.13.5.6. MSGBox Message Status Register m(Default Value: 0x00000000)

| Offset:0x140+N*0x4 (N=0~7) | | | Register Name: **MSGBOXM_MSG_STATUS_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:3 | / | / | / |
| 2:0 | RO | 0x0 | MSG_NUM. Number of unread messages in the message queue. Here, limited to four messages per message queue. 000: There is no message in the message FIFO queue. 001: There is 1 message in the message FIFO queue. 010: There are 2 messages in the message FIFO queue. 011: There are 3 messages in the message FIFO queue. 100: There are 4 messages in the message FIFO queue. 101~111:/ |

### 3.13.5.7. MSGBox Message Queue Register m(Default Value: 0x00000000)

| Offset:0x180+N*0x4 (N=0~7) | | | Register Name: **MSGBOXM_MSG_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R/W | 0x0 | The message register stores the next to be read message of the message FIFO queue. Reads remove the message from the FIFO queue. |

## 3.13.6. Programming Guidelines

Assuming that C0-CPU0 establishes an interconnected channel with CPUS using message queue0 and message queue1. The example is as follow:

CPU0 of Cluster 0

Step 1：CPU0 establishes an interconnection channel with CPUS

writel(readl(BUS_GATING_REG1)|(1<<21) ,BUS_GATING_REG1);    //open Message Box clock gating

writel (readl(BUS_RST_REG1)|(1<<21) , BUS_RST_REG1);          //software reset Message Box

writel(0x00000110, MSGBOX_CTRL_REG0);                //set CPU0 transmits messages to CPUS through Queue0 and
// set CPUS transmits messages to CPU0 through Queue1

Writel(0x4, MSGBOXU_IRQ_EN_REG1);                //enable queue1 reception irq of user1

Writel(0x1, MSGBOXU_IRQ_EN_REG0);                //enable queue0 reception irq of user0


Step 2: CPU0 transmits 4 messages to CPUS

//before transmitting messages, check out queue FIFO full status or message numbers in queue FIFO.

for(i=0;i<4;i++)

writel(queue0_transmit_buf[i], MSGBOXM_MSG_REG0); //write 4 messages stored in

// queue0_transmit_buf[i](i=0~4) to queue0

step 3: CPU0 waits for CPUS to read all messages in queue0

while(readl(MSGBOXM_MSG_STATUS_REG 0));          //CPU0 waits for CPUS to read all messages in queue0


Step 4: CPU0 waits for CPUS transmits messages for itself and receive them in queue1 irq handler

while(!readl(MSGBOXM_FIFO_STATUS_REG1));          //wait for queue1 turns to full status

queue1 irq handler:

while(readl(MSGBOXM_MSG_STATUS_REG1) != 0)        //read all messages from queue1 and store them in

{                                                // queue1_receive_buf[i](i=0~4)

    rdata=readl(MSGBOXM_MSG_REG 1);

    writel(rdata,queue0_receive_buf[i]);

}

CPUS

┈┈┈┈┈┈┈┈┈┈┈┈┈┈┈┈┈┈┈┈┈┈┈┈┈┈┈┈┈┈┈┈┈┈┈┈┈┈┈┈┈┈┈┈┈┈┈

Step 1: CPUS waits for CPU0 transmits messages for itself and receive them in queue0 irq handler

while(!readl(MSGBOXM_FIFO_STATUS_REG0));              //wait for queue0 turns to full status

queue0 irq handler:

while(readl(MSGBOXM_MSG_STATUS_REG0) != 0)        //read all messages from queue0 and store them in

{                                                // queue0_receive_buf[i](i=0~4)

    rdata=readl(MSGBOXM_MSG_REG 0);

    writel(rdata,queue0_receive_buf[i]);

}

Step 2: CPUS transmits 4 messages to CPU0

for(i=0;i<4;i++)

writel(queue1_transmit_buf[i], MSGBOXM_MSG_REG1); //write 4 messages stored in

// queue1_transmit_buf[i](i=0~4) to queue1


The result of the upper instance is: the messages in the queue0_transmit_buf[i](i=0~4) are transmitted to queue0_receive_buf[i](i=0~4) and the messages in the queue1_transmit_buf[i](i=0~4) are transmitted to queue1_receive_buf[i](i=0~4).。

## 3.14. Spinlock

### 3.14.1. Overview

Spinlock provides hardware assistance for synchronizing the processes running on multiple processors in the device. The SpinLock module implements thirty-two 32-bit spinlocks (or hardware semaphores), which provide an efficient way to perform a lock operation of a device resource using a single read access, thus avoiding the need for a 'read-modify-write' bus transfer that not all the programmable cores are capable of.

Spinlocks are present to solve the need for synchronization and mutual exclusion between heterogeneous processors and those not operating under a single, shared operating system. There is no alternative mechanism to accomplish these operations between processors in separate subsystems. However, Spinlocks do not solve all system synchronization issues. They have limited applicability and should be used with care to implement higher level synchronization protocols.

A spinlock is appropriate for mutual exclusion for access to a shared data structure. It should be used only when:

1)  The time to hold the lock is predictable and small (for example, a maximum hold time of less than 200 CPU cycles may be acceptable).
2)  The locking task cannot be preempted, suspended, or interrupted while holding the lock (this would make the hold time large and unpredictable).
3)  The lock is lightly contended, that is the chance of any other process (or processor) trying to acquire the lock while it is held is small.

If the conditions are not met, then a spinlock is not a good candidate. One alternative is to use a spinlock for critical section control (engineered to meet the conditions) to implement a higher level semaphore that can support preemption, notification, timeout or other higher level properties.

The Spinlock includes the following features:

•  Spinlock module includes 32 spinlocks
•  Two kinds of status of lock register: TAKEN and NOT TAKEN

## 3.14.2.  Functionalities Description

### 3.14.2.1.  Typical Applications
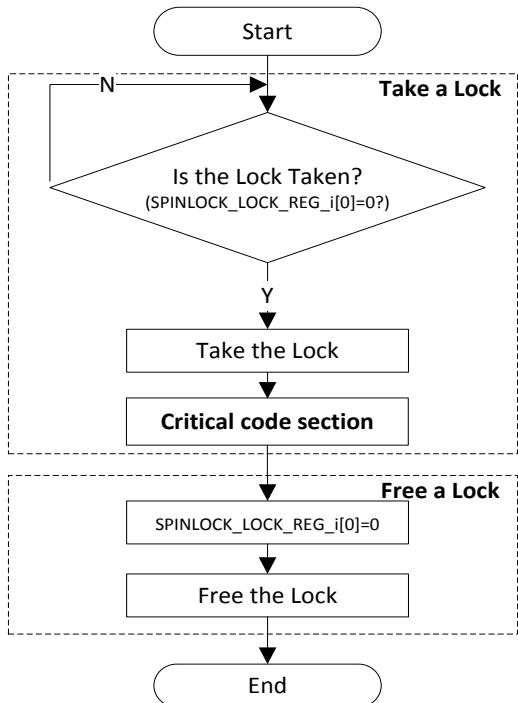


Figure3-10. Spinlock Typical Application Flow Chart

### 3.14.2.2.  Functional Block Diagram
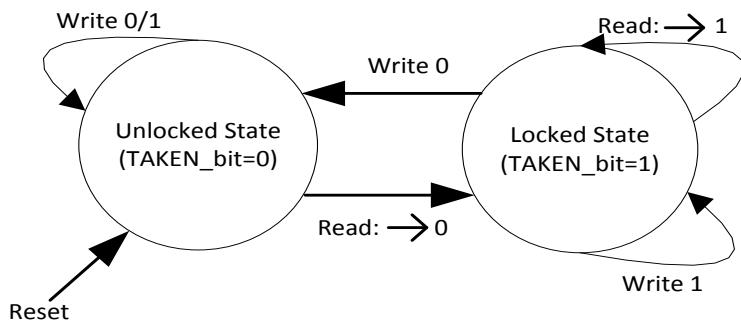


Figure 3-11. Spinlock Lock Register State Diagram

Every lock register has two kinds of states: TAKEN(locked) or NOT TAKEN(Unlocked). Only read-0-access and write-0-access could change lock register' state and the other accesses has no effect. Just 32-bit reads and writes are

supported to access all lock registers.

## 3.14.3. Operation Principle

### 3.14.3.1. Spinlock clock gating and software reset

Spinlock clock gating should be open before using it. Setting *Bus Clock Gating Register1* bit[22] to 1 could activate Spinlock and then de-asserting it's software reset. Setting *Bus Software Reset Register1* bit[22] to 1 could de-assert the software reset of Spinlock. If it is no need to use spinlock, both the gating bit and software reset bit should be set 0.

### 3.14.3.2. Take and free a spinlock

Checking out *SpinLock Register Status* is necessary when a processor would like to take a spinlock. This register stores all 32 lock registers' status: TAKEN or NOT TAKEN(free).

In order to request to take a spinlock, a processor has to do a read-access to the corresponding lock register. If lock register returns 0, the processor takes this spinlock. And if lock register returns 1, the processor must retry.

Writing 0 to a lock register frees the corresponding spinlock. If the lock register is not taken, write-access has no effect. For a taken spinlock, every processor has the privilege to free this spinlock. But it is suggested that the processor which has taken the spinlock free it for strictness.

## 3.14.4. Spinlock Register List

| Module Name | Base Address |
|---|---|
| Spinlock | 0x01C18000 |

| Register Name | Offset | Description |
|---|---|---|
| SPINLOCK_SYSTATUS_REG | 0x0000 | Spinlock System Status Register |
| SPINLOCK_STATUS_REG | 0x0010 | Spinlock Status Register |
| SPINLOCK_LOCK_REGN | 0x100+N*0x4 | Spinlock Register N (N=0~31) |

## 3.14.5. Spinlock Register Description

### 3.14.5.1. Spinlock System Status Register (Default Value: 0x10000000)

| Offset: 0x0 | | | Register Name: **SPINLOCK_SYSTATUS_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:30 | / | / | / |
| 29:28 | RO | 0x1 | LOCKS_NUM. Number of lock registers implemented. 0x1: This instance has 32 lock registers. 0x2: This instance has 64 lock registers. 0x3: This instance has 128 lock registers. 0x4: This instance has 256 lock registers. |
| 27:16 | / | / | / |
| 15:9 | / | / | / |
| 8 | RO | 0x0 | IU0. In-Use flag0, covering lock register0-31. 0: All lock register 0-31 are in the Not Taken state. 1: At least one of the lock register 0-31 is in the Taken state. |
| 7:0 | / | / | / |

### 3.14.5.2. Spinlock Register Status (Default Value: 0x00000000)

| Offset: 0x10 | | | Register Name: **SPINLOCK_STATUS_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| [i] (i=0~31) | RO | 0x0 | LOCK_REG_STATUS. SpinLock[i] status (i=0~31) 0: The Spinlock is free, 1: The Spinlock is taken. |

### 3.14.5.3. Spinlock Register N (N=0 to 31)( Default Value:0x00000000)

| Offset:0x100+N*0x4 (N=0~31) | | | Register Name: **SPINLOCKN_LOCK_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:1 | / | / | / |
| 0 | R/W | 0x0 | TAKEN. Lock State. Read 0x0: The lock was previously Not Taken (free).The requester is granted the lock. |

| | | | Write 0x0: Set the lock to Not Taken (free). |
|---|---|---|---|
| | | | Read 0x1: The lock was previously Taken. The requester is not granted the lock and must retry. |
| | | | Write 0x1: No update to the lock value. |

## 3.14.6. Programming Guidelines

Take C0-CPU0's synchronization with CPUS with Spinlock0 for an example, CPU0 takes the spinlock0 firstly in the instance：

CPU0 of Cluster 0

───────────────────────────────────────────────────────────────────

Step 1: CPU0 initializes Spinlock

writel(readl(BUS_GATING_REG1)|(1<<22) ,BUS_GATING_REG1);   //open Spinlock clock gating

writel (readl(BUS_RST_REG1)|(1<<22) , BUS_RST_REG1);       //software reset Spinlock

Step 2: CPU0 requests to take spinlock0

rdata=readl(SPINLOCK_STATUS_REG0);                        //check lock register0 status, if it is taken, check till

if(rdata != 0)   rdata=readl(SPINLOCK_STATUS_REG0);       // lock register0 is free

    ⋮

rdata=readl(SPINLOCKN_LOCK_REG0);                        //request to take spinlock0, if fail, retry till

if(rdata != 0)   rdata=readl(SPINLOCKN_LOCK_REG0);       // lock register0 is taken

    ⋮

  ---------- CPU0 critical code section ----------

Step 3: CPU0 free spinlock0

writel (0, SPINLOCKN_LOCK_REG0);                         //CPU0 frees spinlock0

Step 4: CPU0 waits for CPUS' freeing spinlock0

writel (readl(SPINLOCK_STATUS_REG0) == 1);              // CPU0 waits for CPUS' freeing spinlock0

CPUS

───────────────────────────────────────────────────────────────────

Step 1: CPU0 has taken spinlock0, CPUS waits for CPU0' freeing spinlock0

while(readl(SPINLOCK_STATUS_REG0) == 1);               // CPUS waits for CPU0' freeing spinlock0

Step 2: CPUS takes spinlock0 and go on

---------- CPUS critical code section ----------

Step 3: CPUS frees spinlock0

writel (0, SPINLOCKN_LOCK_REG0);                        //CPUS frees spinlock0

## 3.15. Security System

### 3.15.1. Overview

The Security System (SS) is one encrypt/ decrypt function accelerator. It is suitable for a variety of applications. It can support both encryption/decryption and signature/verification, calculate the hash value. Several modes are supported by the security system. SS has an internal DMA(IDMA) controller to transfer data between SS and memory.

It includes the following features:

- Support symmetrical algorithm :AES, DES, TDES
- Support Secure Hash algorithm: MD5, SHA-1,SHA-224,SHA-256
- Support asymmetrical algorithm :RSA512/1024/2048/3072-bits
- Support signature and verification based on the RSA algorithm
- Support 160-bits hardware PRNG with 175-bits seed
- Support 256-bits hardware TRNG
- Support ECB, CBC, CTR modes for DES/TDES
- Support ECB, CBC, CTR, CTS modes for AES
- Support 128-bits, 192-bits and 256-bits key size for AES
- Support IDMA mode
- Support serial and parallel mode

## 3.15.2. Block Diagram
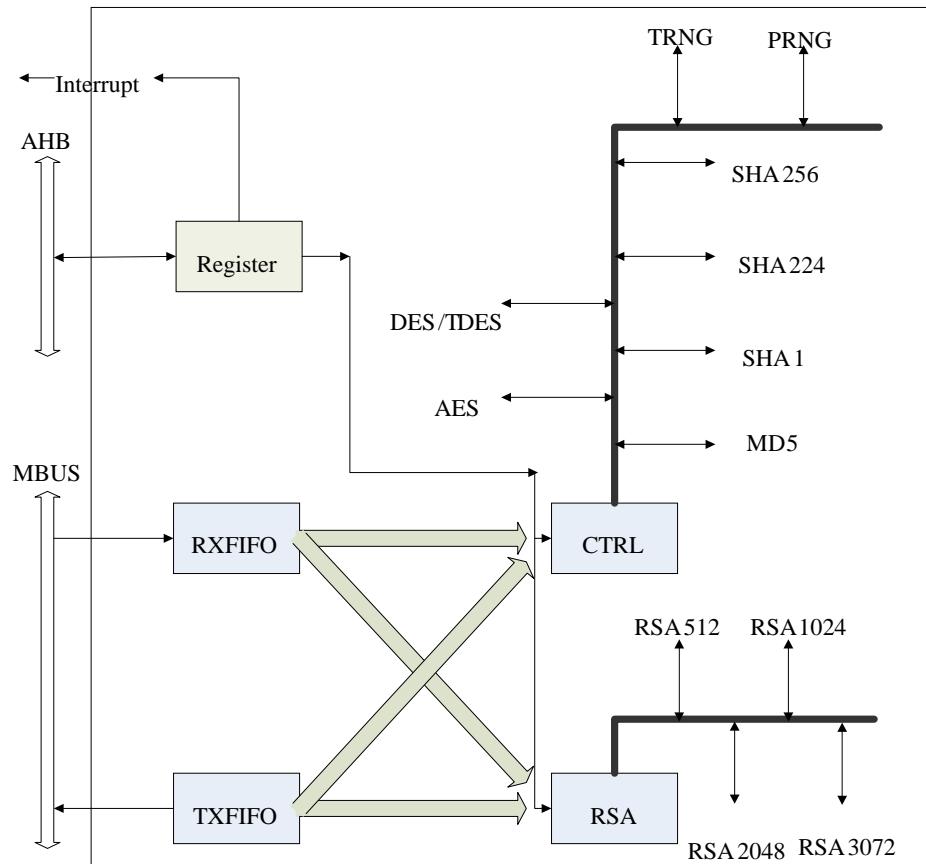


Figure 3-12. SS Block Diagram

## 3.15.3. Security System Register List

| Module Name | Base Address |
|---|---|
| SS | 0x01C15000 |

| Register Name | Offset | Description |
|---|---|---|
| SS_CTL | 0x00 | Security Control Register |
| SS_ICR | 0x04 | Security Interrupt Control Register |
| SS_ISR | 0x08 | Security Interrupt Status Register |
| SS_Key_Address | 0x10 | Security Input Key Address Register |
| SS_IV_Address | 0x18 | Security Initialization Vector/Preload Counter/Public Modulus Address Register |
| SS_DataSrc_Address | 0x20 | Security Source Address Register |

| SS_DataDst_Address | 0x28 | Security Destination Address Register |
|---|---|---|
| SS_Data_Length | 0x30 | Security Data Length Register |
| SS_CTR0 | 0x34 | Security Counter0 Register for Stream0 |
| SS_CTR1 | 0x38 | Security Counter1 Register for Stream0 |
| SS_CTR2 | 0x3C | Security Counter2 Register for Stream0 |
| SS_CTR3 | 0x40 | Security Counter3 Register for Stream0 |
| SS_CTR4 | 0x48 | Security Counter4 Register for Stream1 |
| SS_CTR5 | 0x4C | Security Counter5Register for Stream1 |
| SS_CTR6 | 0x50 | Security Counter6 Register for Stream1 |
| SS_CTR7 | 0x54 | Security Counter7 Register for Stream1 |

## 3.15.4. Security System Register Description

### 3.15.4.1. Security System Control Register(Default Value: 0x20000000)

| Offset: 0x0 | | | Register Name: **SS_CTL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0 | Stream1_SELECT<br>This bit is set by software and cleared by hardware.<br>Software write "1" select stream1, when write "0" invalid. |
| 30 | R/W | 0 | Stream0_SELECT<br>This bit is set by software and cleared by hardware.<br>Write "1" select stream0,,when write "0" invalid.<br>Note:When execute a stream,be sure to write bit31 or bit30 to 1; When both bit31 and bit30 wrote at the same time, the priority of bit30 is higher than bit31; When both bit31 and bit30 are 0, on behalf of no stream in the execution. |
| 29 | R | 1 | SS_IDLE<br>0: the status of SS is busy<br>1: the status of SS is idle |
| 28 | R/W | 0 | FLOW_MODE<br>Mode with the last flow<br>0:non-continue mode<br>1:continue mode |
| 27 | R/W | 0 | DMA Read/Write Consistent<br>0:Send end flag after data write-instruction finished<br>1:Read data when receive response of write-instruction ,if write-operation is non-finished, waiting until write-operation finished. |
| 26:23 | R/W | 0 | SKEY_SELECT |

| | | | AES key select |
|---|---|---|---|
| | | | 0: Select input SS_KEYx (Normal Mode) |
| | | | 1:Select    {SSK} |
| | | | 2:Select    {HUK} |
| | | | 3:Select    {RSSK} |
| | | | 4-7:Reserved |
| | | | 8-15: Select internal Key n (n from 0 to 7) |
| | | | DIE_ID |
| 22:20 | R | x | Die Bonding ID |
| 19 | / | / | / |
| | | | TRNG/PRNG_MODE |
| | | | TRNG/PRNG generator mode |
| | | | 0: One-shot mode |
| 18 | R/W | 0 | 1: Continue mode |
| | | | IV_MODE |
| | | | IV Steady of SHA-1/SHA-224/SHA-256/MD5 constants |
| | | | 0: Constants |
| | | | 1: Arbitrary IV |
| 17 | R/W | 0 | Notes: It is only used for SHA-1/SHA-224/SHA-256/MD5 engine. |
| | | | AES_CTS_LAST_PACKAGE_FLAG |
| | | | When writing "1", it means this is the last package for AES-CTS mode. (the |
| 16 | R/W | 0 | size of the last package >128bit) |
| 15 | / | / | / |
| | | | SS_OP_MODE |
| | | | SS Operation Mode |
| | | | 00: Electronic Code Book (ECB) mode |
| | | | 01: Cipher Block Chaining (CBC) mode |
| | | | 10: Counter (CTR) mode |
| 14:13 | R/W | 0 | 11: AES Ciphertext Stealing (CTS) mode |
| | | | CTR_WIDTH |
| | | | Counter Width for CTR Mode |
| | | | 00: 16-bits Counter |
| | | | 01: 32-bits Counter |
| | | | 10: 64-bits Counter |
| 12:11 | R/W | 0 | 11: 128-bits Counter |
| | | | RSA Pubic Modulus_Width |
| | | | 00:512 bit |
| | | | 01:1024 bit |
| | | | 10:2048 bit |
| 10:9 | R/W | 0 | 11:3072 bit |
| | | | AES_KEY_SIZE |
| | | | Key Size for AES |
| | | | 00: 128-bits |
| | | | 01: 192-bits |
| 8:7 | R/W | 0 | 10: 256-bits |

| | | | 11: Reserved |
|---|---|---|---|
| 6 | R/W | 0 | SS_OP_DIR<br>SS Operation Direction<br>0: Encryption<br>1: Decryption |
| 5:2 | R/W | 0 | SS_METHOD<br>SS Method<br>0000: AES<br>0001: DES<br>0010: Triple DES (3DES)<br>0011: MD5<br>0100: PRNG<br>0101: TRNG<br>0110: SHA-1<br>0111: SHA-224<br>1000: SHA-256<br>1001: RSA<br>Others: Reserved |
| 1 | R/W | 0 | PRNG/TRNG_START<br>PRNG/TRNG start bit<br>In PRNG/TRNG one-shot mode, write '1' to start PRNG/TRNG. After generating one group random data , this bit is clear to '0' by hardware. |
| 0 | R/W | 0 | SS_Start<br>Write '1' to start SS . |

### 3.15.4.2. Security System Interrupt Control Register(Default Value: 0x00000000)

| Offset: 0x4 | | | Register Name: **SS_ INT_CTL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:2 | / | / | / |
| 1 | R/W | 0 | FLOW1_ENCRY/DECRY_END_Enable<br>Flow1 Encry/Decry End Available Interrupt Enable<br>0: Disable<br>1: Enable |
| 0 | R/W | 0 | FLOW0_ENCRY/DECRY_END_Enable<br>Flow0 Encry/Decry End Available Interrupt Enable<br>0: Disable<br>1: Enable |

### 3.15.4.3. Security System Interrupt Status Register(Default Value: 0x00000000)

| Offset: 0x8 | Register Name: **SS_ INT_STATUS_REG** |
|---|---|

| Bit | R/W | Default/Hex | Description |
|---|---|---|---|
| 31:2 | / | / | / |
| 1 | R/W | 0 | FLOW1_ENCRY/DECRY_END_PENDING_BIT<br>Flow1 Encry/Decry End Available Pending bit<br>0: No end pending<br>1: end pending<br>Notes: Write '1' to clear it |
| 0 | R/W | 0 | FLOW0_ENCRY/DECRY_END_PENDING_BIT<br>Flow0 Encry/Decry End Available Pending bit<br>0: No end pending<br>1: end pending<br>Notes: Write '1' to clear it. |

### 3.15.4.4. Security System Key Address Register(Default Value: 0x00000000)

| Offset: 0x10 | | | Register Name: **SS_ KEY_ADR_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R/W | 0 | SS Key Input Address(2 words alignment)write in the register |

### 3.15.4.5. Security System Pubic Modulus/IV/ Counter Address Register(Default Value: 0x00000000)

| Offset: 0x18 | | | Register Name: **SS_PM/IV/CNT_ADR_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R/W | 0 | SS Initialization Vector/Preload Counter/Pubic Modulus Address(2words alignment)write in the register |

### 3.15.4.6. Security System Data Source Address Register(Default Value: 0x00000000)

| Offset: 0x20 | | | Register Name: **SS_DataSrc_ADR_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R/W | 0 | SS Data Source Address<br>Source address (2words alignment)write in the register |

### 3.15.4.7. Security System Data Destination Address Register(Default Value: 0x00000000)

| Offset: 0x28 | | | Register Name: **SS_DataDst_ADR_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R/W | 0 | SS Data Destination Address<br>Destination address (2words alignment)write in the register |

| | | | Note:Read out data from destination address in bytes only for SHA1/ SHA224/SHA256 . |
|---|---|---|---|

### 3.15.4.8.  Security System Data Length Register(Default Value: 0x00000000)

| Offset: 0x30 | | | Register Name: **SS_Data_LEN_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R/W | 0 | SS Data Length<br>Record the length of plaint/cipher and random data access in Word.<br>4 words alignment for AES;<br>2 words alignment for DES/3DES;<br>16 words alignment for SHA1/MD5/SHA224/SHA256;<br>5 words alignment for PRNG random data;<br>8 words alignment for TRNG random data;<br>Note:When it uses in AES-CTS mode , Data Length access in Byte. |

### 3.15.4.9.  Security System Counter[n] Register0(Default Value: 0x00000000)

| Offset: 0x34+4*n | | | Register Name: **SS_ CTR[n]_REG0** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R | 0 | SS_CTR_VALUE<br>Record counter's middle value every block in AES/DES/3DES CTR mode (n=0~3)<br>Note:SS_CTR[0] ~SS_CTR[3] for stream0 only. |

### 3.15.4.10.   Security System Counter[n] Register1(Default Value: 0x00000000)

| Offset: 0x48+4*(n-4) | | | Register Name: **SS_ CTR[n]_REG1** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R | 0 | SS_CTR_VALUE<br>Record counter's middle value every block in AES/DES/3DES CTR mode (n=4~7)<br>Note:SS_CTR[4] ~SS_CTR[7] for stream1 only. |

## 3.15.5.  Security System Clock Requirement

| Clock Name | Description | Requirement |
|---|---|---|
| ahb_clk | AHB bus clock | >=24MHz |
| ss_clk | SS serial clock | <= 300MHz && >=24MHz |

## 3.15.6. Programming Guidelines

(1) Datasrc_address needs align by 2 words.

(2) Datadst_address needs align by 2 words.

(3) After the selected signal of channel-0 or channel-1 is recognized as low level(the bit31 or bit30 of SS_CTL register is 0),channel-0 or channel-1 could be requested to calculate.

(4) The input data sequence of RSA is that the low word store in the low address.

(5) Before the Hash algorithm is operated, the SS controller need reset, which avoid the influence of other algorithms to the Hash algorithm.

(6) For SHA1/SHA224/SHA256,It should be noted the sequence of the initial hash value.

SHA1/SHA224/SHA256 is the big-endian algorithm, within each word,the most significant bit is stored in the left-most bit position.For example,the initial hash value of SHA1 in Fips180-2,$H^{(0)}$ shall consist of the following five 32-bit words,in hex:
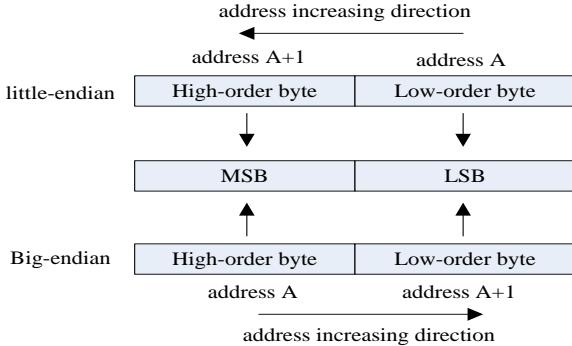
$H_0^{(0)}$ = 67452301

$H_1^{(0)}$ = efcdab89

$H_2^{(0)}$ = 98badcfe

$H_3^{(0)}$ = 10325476

$H_4^{(0)}$ = c3d2e1f0



The default access mode of ARM is litter-endian.So When we write the initial value in the IV descriptor address, according to the following array input sequence:

For SHA1:

unsigned char iv_sha1[20]={

0x67,0x45,0x23,0x01,0xef,0xcd,0xab,0x89,0x98,0xba,0xdc,0xfe,0x10,0x32,0x54,0x76,

0xc3,0xd2,0xe1,0xf0};

Then:

IV descriptor address :　　　　0x01234567

IV descriptor address +0x4:　　0x89abcdef

IV descriptor address +0x8:　　0xfedcba98

IV descriptor address +0xC:　　0x76543210

IV descriptor address +0x10:　0xf0e1d2c3

For SHA224:

unsigned char iv_sha224[32]={

0xc1,0x05,0x9e,0xd8,0x36,0x7c,0xd5,0x07,0x30,0x70,0xdd,0x17,0xf7,0x0e,0x59,0x39,

0xff,0xc0,0x0b,0x31,0x68,0x58,0x15,0x11,0x64,0xf9,0x8f,0xa7,0xbe,0xfa,0x4f,0xa4};

For SHA256:

unsigned char iv_sha256[32]={

0x6a,0x09,0xe6,0x67,0xbb,0x67,0xae,0x85,0x3c,0x6e,0xf3,0x72,0xa5,0x4f,0xf5,0x3a,

0x51,0x0e,0x52,0x7f,0x9b,0x05,0x68,0x8c,0x1f,0x83,0xd9,0xab,0x5b,0xe0,0xcd,0x19};

(7) The generation of the true random numbers uses one 16MHz Oscillator source.So when using TRNG algorithm,the 16MHz oscillator need enable,that the bit[0] of **OSC24M_CTRL_REG** in R_PRCM is wrote to 1.After completed,the 16MHz oscillator need disable.

## 3.16. Security ID

### 3.16.1. Overview

There is one on chip EFUSE, which provides 128-bit, 64-bit and one 32-bit electrical fuses for security application. The users can use them as root key, security JTAG key and other applications.

It includes the following features:

- 128-bit electrical fuses for chip ID
- 64-bit electrical fuses for thermal sensor

## 3.17. Secure Memory Controller

### 3.17.1. Overview

The SMC is an Advanced Microcontroller Bus Architecture compliant System-on-Chip peripheral. It is a high-performance, area-optimized address space controller with on-chip AMBA bus interfaces that conform to the AMBA Advanced extensible Interface protocol and the AMBA Advanced Peripheral Bus protocol.

You can configure the SMC to provide the optimum security address region control functions required for your intended application.

The SMC includes the following features:

- Enables you to program security access permissions each address region.
- Permits the transfer of data between master and slave only if the security status of the AXI transaction matches the security settings of the memory region it addresses.

### 3.17.2. Functionalities Description

By default, the SMC performs read or write speculative that means it forwards an AXI transaction address to a slave, before it verifies that the AXI transaction is permitted to read address or write address respectively.

The SMC only permits the transfer of data between its AXI bus interfaces, after verifying the access that the read or write access is permitted respectively. If the verification fails, then it prevents the transfer of data between the master and slave as Denied AXI transactions.

When the speculative accesses are disabled, the SMC verifies the permissions of the access before it forwards the access to the salve. If the SMC:

- Permits the access, it commences an AXI transaction to the slave, and it adds one clock latency.
- Denies the access, it prevents the transfer of data between the master and slave. In this situation, the slave is unaware when the SMC prevents the master from accessing the slave.

## 3.17.3. Diagram and Table

### 3.17.3.1. DRM Block Diagram

G. NS.M stands for General Non-secure Master

D. NS.M stands for Non-secure Master appointed by DRM

S.M. stands for Secure Mater



G.NS.M only can read data from NSZ and write data into NSZ

D.NS.M can read data from NSZ and DRM, but only can write data into DRM

S.M can read data from the whole DRAM SPACE
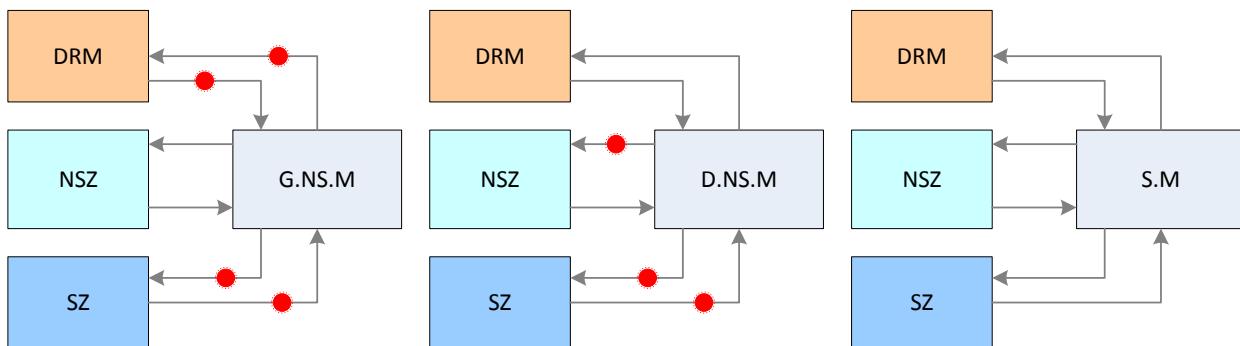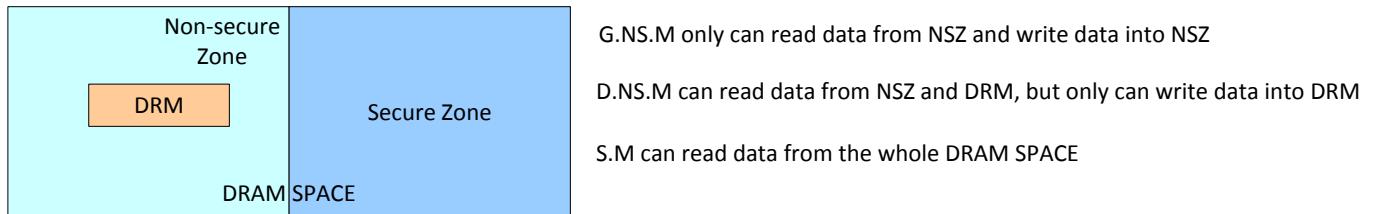


Figure 3-13. DRM Block Diagram

### 3.17.3.2. Master ID Table

| ID | Master | ID | Master |
|----|--------|----|--------|
| 0 | CPU | 11 | DMA |
| 1 | GPU | 12 | VE |
| 2 | CPUS | 13 | CSI |
| 3 | ATH (test interface for AHB) | 14 | NAND |
| 4 | USB-DRD | 15 | SS |
| 5 | MSTG0 (MMC0) | 16 | DE_RT-MIXER0 |
| 6 | MSTG1 (MMC1) | 17 | DE_RT-MIXER1 |
| 7 | MSTG2 (MMC2) | 18 | DE_RT-WB |
| 8 | USB0 | 19 | DE_ROT |
| 9 | USB1 | 20 | |
| 10 | EMAC | 21 | |

### 3.17.3.3. Region Size Table

| Size<n> | Size of region<n> | Base address constraints |
|---|---|---|
| b000000-b001101 | Reserved | - |
| b001110 | 32KB | - |
| b001111 | 64KB | Bit [15] must be zero |
| b010000 | 128KB | Bits [16:15] must be zero |
| b010001 | 256KB | Bits [17:15] must be zero |
| b010010 | 512KB | Bits [18:15] must be zero |
| b010011 | 1MB | Bits [19:15] must be zero |
| b010100 | 2MB | Bits [20:15] must be zero |
| b010101 | 4MB | Bits [21:15] must be zero |
| b010110 | 8MB | Bits [22:15] must be zero |
| b010111 | 16MB | Bits [23:15] must be zero |
| b011000 | 32MB | Bits [24:15] must be zero |
| b011001 | 64MB | Bits [25:15] must be zero |
| b011010 | 128MB | Bits [26:15] must be zero |
| b011011 | 256MB | Bits [27:15] must be zero |
| b011100 | 512MB | Bits [28:15] must be zero |
| b011101 | 1GB | Bits [29:15] must be zero |
| b011110 | 2GB | Bits [30:15] must be zero |
| b011111 | 4GB | Bits [31:15] must be zero |
| B100000 | 8GB | Bits [32:15] must be zero |

## 3.17.4.  Operation Modes

### 3.17.4.1. Security inversion

There are two modes of operation for the region security permissions, with or without security inversion. By default, if you program a region to support non-secure accesses, the SMC ensures that region must also support secure accesses. For example, if you program the region permissions for region 3 to be non-secure read only, the SMC permits access to region 3 for secure reads and non-secure reads. If you require that some regions are not accessible to masters in Secure state, but are accessible in Non-secure state, then you must enable security inversion. See **Region security permissions** section and *Security Inversion Enable Register* for more information.

### 3.17.4.2. Inversion is disabled

**Table 3-2** shows the possible security permissions when security inversion is disabled.

**Table 3-2**

| SPN field | Secure Read | Secure Write | Non-secure Read | Non-secure Write |
|---|---|---|---|---|
| 4b0000 | No | No | No | No |
| 4b0100 | No | Yes | No | No |
| 4b0001, 4b0101 | No | Yes | No | Yes |
| 4b1000 | Yes | No | No | No |
| 4b0010, 4b1010 | Yes | No | Yes | No |
| 4b1100 | Yes | Yes | No | No |
| 4b1001, 4b1101 | Yes | Yes | No | Yes |
| 4b0110, 4b1110 | Yes | Yes | Yes | No |
| 4b0011-4b1111 | Yes | Yes | Yes | Yes |

### 3.17.4.3. Inversion is disabled

If you enable security inversion, the SMC permits you to program any combination of security permissions as **Table 3-3** shows.

**Table 3-3**

| SPN field | Secure Read | Secure Write | Non-secure Read | Non-secure Write |
|---|---|---|---|---|
| 4b0000 | No | No | No | No |
| 4b0001 | No | No | No | Yes |
| 4b0010 | No | No | Yes | No |
| 4b0011 | No | No | Yes | Yes |
| 4b0100 | No | Yes | No | No |
| 4b0101 | No | Yes | No | Yes |
| 4b0110 | No | Yes | Yes | No |
| 4b0111 | No | Yes | Yes | Yes |
| 4b1000 | Yes | No | No | No |
| 4b1001 | Yes | No | No | Yes |
| 4b1010 | Yes | No | Yes | No |
| 4b1011 | Yes | No | Yes | Yes |
| 4b1100 | Yes | Yes | No | No |
| 4b1101 | Yes | Yes | No | Yes |
| 4b1110 | Yes | Yes | Yes | No |
| 4b1111 | Yes | Yes | Yes | Yes |

## 3.17.5. SMC Register List

| Module Name | Base Address |
|---|---|
| SMC | 0x01C1E000 |

| Register Name | Offset | Description |
|---|---|---|
| SMC_CONFIG_REG | 0x0 | SMC Configuration Register |
| SMC_ACTION_REG | 0x4 | SMC Action Register |
| SMC_LD_RANGE_REG | 0x8 | SMC Lock Down Range Register |
| SMC_LD_SELECT_REG | 0xC | SMC Lock Down Select Register |
| SMC_INT_STATUS_REG | 0x10 | SMC Interrupt Status Register |
| SMC_INT_CLEAR_REG | 0x14 | SMC Interrupt Clear Register |
| SMC_MST_BYP_REG | 0x18 | SMC Master Bypass Register |
| SMC_MST_SEC_REG | 0x1C | SMC Master Secure Register |
| SMC_FAIL_ADDR_REG | 0x20 | SMC Fail Address Register |
| SMC_FAIL_CTRL_REG | 0x28 | SMC Fail Control Register |
| SMC_FAIL_ID_REG | 0x2C | SMC Fail ID Register |
| SMC_SPECU_CTRL_REG | 0x30 | SMC Speculation Control Register |
| SMC_SEC_INV_EN_REG | 0x34 | SMC Security Inversion Enable Register |
| SMC_MST_ATTRI_REG | 0x48 | SMC Master Attribute Register |
| DRM_MASTER_EN_REG | 0x50 | DRM Master Enable Register |
| DRM_ILLACCE_REG | 0x58 | DRM Illegal Access Register |
| DRM_STATADDR_REG | 0x60 | DRM Start Address Register |
| DRM_ENDADDR_REG | 0x68 | DRM End Address Register |
| SMC_REGION_SETUP_LO_REG | 0x100+N*0x10 | Region Setup Low Register N<br><br>(N=0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15) |
| SMC_REGION_SETUP_HI_REG | 0x104+N*0x10 | Region Setup High Register N<br><br>(N=0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15) |
| SMC_REGION_ATTR_REG | 0x108+N*0x10 | Region Attribute Register N<br><br>(N=0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15) |

## 3.17.6. SMC Register Description

### 3.17.6.1. SMC Configuration Register(Default Value: 0x00001F0F)

| Offset: 0x0 | | | Register Name: **SMC_CONFIG_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:14 | / | / | / |
| 13:8 | R | 0x1F | ADDR_WIDTH_RTN.<br>Address width. Return the width of the AXI address bus.<br>6'b 000000-6'b011110 reserved.<br>6'b 011111 = 32-bit<br>……<br>6'b 111111 = 64-bit |
| 7:4 | / | / | / |
| 3:0 | R | 0xF | REGIONS_RTN.<br>Returns the number of the regions that the SMC provides.<br>4'b0000 = reserved<br>4'b0001 = 2 regions<br>……<br>4'b1111 = 16 regions. |

### 3.17.6.2. SMC Action Register(Default Value: 0x00000001)

| Offset: 0x4 | | | Register Name: **SMC_ACTION_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:2 | / | / | / |
| 1:0 | R/W | 0x1 | SMC_INT_RESP.<br>Control how the SMC uses the bresps[1:0], rresps[1:0], and smc_int signals when a region permission failure occurs:<br>2'b00 = sets smc_int LOW and issues an OKEY response<br>2'b01 = sets smc_int LOW and issues a DECERR response<br>2'b10 = sets smc_int HIGH and issues an OKEY response<br>2'b11 = sets smc_int HIGH and issues a DECERR response |

**Note：**

This action is only valid for CPU access, not for MBUS and DMA access.

### 3.17.6.3. SMC Lockdown Range Register(Default Value: 0x00000000)

| Offset: 0x8 | | | Register Name: **SMC_LD_RANGE_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x0 | LOCKDOWN_EN.<br>When set to 1, it enables the lockdown_regions field to control the regions that are to be locked. |
| 30:4 | / | / | / |

| 3:0 | R/W | 0x0 | NO_REGIONS_LOCKDOWN. |
| | | | Control the number of regions to lockdown when the enable bit is set to 1. |
| | | | 4'b0000 = region no_of_regions-1 is locked |
| | | | 4'b0001 = region no_of_regions-1 to region no_of_regions-2 are locked |
| | | | …… |
| | | | 4'b1111 = region no_of_regions-1 to region no_of_regions-16 are locked |

**Note1:**

No_of_regions is the value of the no_of_regions field in the configuration register.

**Note2:**

the value programmed in lockdown_range register must not be greater than no_of_regions-1 ,else all regions are locked.

### 3.17.6.4. SMC Lockdown Select Register(Default Value: 0x00000000)

| Offset: 0xC | | | Register Name: **SMC_LD_SELECT_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:3 | / | / | / |
| 2 | R/W | 0x0 | ACCESS_TYPE_SPECU. |
| | | | Modify the access type of the speculation_control register: |
| | | | 0: no effect. The speculation register remains RW. |
| | | | 1: speculation_control register is RO |
| 1 | R/W | 0x0 | ACCESS_TYPE_SEC_INV_EN. |
| | | | Modify the access type of the security_inversion_en register. |
| | | | 0: no effect. Security_inversion_en register remains RW. |
| | | | 1: security_inversion_en register is RO |
| 0 | R/W | 0x0 | ACCESS_TYPE_LOCKDOWN_RANGE. |
| | | | Modify the access type of the lockdown_range register. |
| | | | 0: no effect. Lockdown_range register remains RW |
| | | | 1: lockdown_range register is RO. |

### 3.17.6.5. SMC Interrupt Status Register(Default Value: 0x00000000)

| Offset: 0x10 | | | Register Name: **SMC_INT_STATUS_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:2 | / | / | / |
| 1 | R | 0x0 | INT_OVERRUN. |
| | | | When set to 1, it indicates the occurrence of two or more region permission failure since the interrupt was last cleared. |
| 0 | R | 0x0 | INT_STATUS. |
| | | | Return the status of the interrupt. |
| | | | 0: interrupt is inactive. |
| | | | 1: interrupt is active. |

### 3.17.6.6. SMC Interrupt Clear Register(Default Value: 0x00000000)

| Offset: 0x14 | | | Register Name: **SMC_INT_CLEAR_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R/W | 0x0 | SMC_CLR_REG. Write any value to the int_clear register sets the : Status bit to 0 in the int_status register Overrun bit to 0 in the int_status register. Note: It will be auto clear after the write operation. |

### 3.17.6.7. SMC Master Bypass Register(Default Value: 0xFFFFFFFF)

| Offset: 0x18 | | | Register Name: **SMC_MST_BYP_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R/W | 0xFFFFFFFF | SMC_MASTER_BYPASS_EN. SMC Master n Bypass Enable. (n = 0~31, see the Table 1. MASTER and MASTER ID for detail.) Note: Bit[31:0] stand for Master ID [31:0] If the master n bypass enable is set to 0, the master n access must be through the SMC. 0: Bypass Disable 1: Bypass Enable. |

### 3.17.6.8. SMC Master Secure Register(Default Value: 0x00000000)

| Offset: 0x1C | | | Register Name: **SMC_MST_SEC_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R/W | 0x0 | SMC_MASTER_SEC. SMC Master n Secure Configuration. (n = 0~31，see the Table 1 for detail.) 0: secure 1: non-secure. |

### 3.17.6.9. SMC Fail Address Register(Default Value: 0x00000000)

| Offset: 0x20 | | | Register Name: **SMC_FAIL_ADDR_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R | 0x0 | FIRST_ACCESS_FAIL. Return the address bits [31:0] of the first access to fail a region permission check after the interrupt was cleared. |

| | | | For external 16-bit DDR2, the address [2:0] is fixed to zero. |
|---|---|---|---|
| | | | For external 32-bit DDR2 and 16-bit DDR3, the address [3:0] is fixed to zero. |
| | | | For external 32-bit DDR3, the address [4:0] is fixed to zero. |

**Note：**

if the master ID="SRAM" and the register value is between 0x80000 to 0xBFFFF,

the real address should be divide by 4.

### 3.17.6.10. SMC Fail Control Register(Default Value: 0x00000000)

| Offset: 0x28 | | | Register Name: **SMC_FAIL_CTRL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:25 | / | / | / |
| 24 | R | 0x0 | READ_WRITE.<br>This bit indicates whether the first access to fail a region permission check was a write or read as:<br>0 = read access<br>1 = write access. |
| 23:22 | / | / | / |
| 21 | R | 0x0 | NON_SECURE.<br>After clearing the interrupt status, this bit indicates whether the first access to fail a region permission check was non-secure. Read as:<br>0 = secure access<br>1 = non-secure access |
| 20 | R | 0x0 | PRIVILEGED.<br>After clearing the interrupt status, this bit indicates whether the first access to fail a region permission check was privileged. Read as:<br>0 = unprivileged access.<br>1 = privileged access |
| 19:0 | / | / | / |

### 3.17.6.11. SMC Fail ID Register(Default Value: 0x00001F00)

| Offset: 0x2C | | | Register Name: **SMC_FAIL_ID_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:24 | / | / | / |
| 23:16 | R | 0x0 | FAIL_BST_LEN.<br>Fail burst length.<br>0 = 1 word length<br>……<br>0xf =16 words length |
| 15:8 | / | / | / |
| 7:0 | R | 0x0 | FAIL_MASTER_ID. |

| | | | Fail Master ID. |
|---|---|---|---|
| | | | The value stands for master id, see the Table 1 MASTER and MASTER ID for detail. |

### 3.17.6.12.    SMC Speculation Control Register(Default Value: 0x00000000)

| Offset: 0x30 | | | Register Name: **SMC_SPECU_CTRL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:2 | / | / | / |
| 1 | R/W | 0x0 | WRITE_SPECU. Write_speculation. Control the write access speculation: 0 = write access speculation is enabled 1 = write access speculation is disabled. |
| 0 | R/W | 0x0 | READ_SPECU. Read_speculation. Control the read access speculation: 0 = read access speculation is enabled 1 = read access speculation is disabled. |

### 3.17.6.13.    SMC Security Inversion Enable Register(Default Value: 0x00000000)

| Offset: 0x34 | | | Register Name: **SMC_SEC_INV_EN_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:1 | / | / | / |
| 0 | R/W | 0x0 | SEC_INV_EN. Security_inversion_en. Controls whether the SMC permits security inversion to occur. 0 = security inversion is not permitted. 1 = security inversion is permitted. This enables a region to be accessible to masters in Non-secure state but not accessible to masters in Secure state. See Table 2 and Table 3. |

### 3.17.6.14.    SMC Master Attribute Register(Default Value: 0x00000000)

| Offset: 0x48 | | | Register Name: **SMC_MST_ATTRI_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R/W | 0x0 | MST_ATTRI. 0: The secure attribute of master is up to master security extensions; 1: The secure attribute of master is up to *Master Secure Register*. |

### 3.17.6.15. SMC Master Enable Register(Default Value: 0x00000000)

| Offset: 0x50 | | | Register Name: **DRM_MASTER_EN_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x0 | DRM_EN. <br> DRM enable. |
| 30:12 | / | / | / |
| 13 | R/W | 0x0 | GPU_WRITE_EN <br> GPU write enable. |
| 12 | R/W | 0x0 | GPU_READ_EN <br> GPU read enable. |
| 11:8 | / | / | / |
| 7 | R/W | 0x0 | DE_ROT <br> DE_ROT enable. |
| 6 | R/W | 0x0 | DE_RT-WB <br> DE_RT-WB enable. |
| 5 | R/W | 0x0 | DE_RT-MIXER1 <br> DE_RT-MIXER1 enable. |
| 4 | R/W | 0x0 | DE_RT-MIXER0 <br> DE_RT-MIXER0 enable. |
| 3:1 | / | / | / |
| 0 | R/W | 0x0 | VE_ENCODE_EN <br> VE encode enable. |

### 3.17.6.16. SMC Illegal Access Register(Default Value: 0x00000000)

| Offset: 0x58 | | | Register Name: **DRM_ILLACCE_REG0** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | RO | 0x0 | DRM_ILLACCE_REG. <br> When a master, which is non-secure, accesses the DRM space, then the relevant bit will be set up. See Table 1 for detail. |

### 3.17.6.17. SMC Start Address Register(Default Value: 0x00000000)

| Offset: 0x60 | | | Register Name: **DRM_STATADDR_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:15 | R/W | 0x0 | DRM_STATADDR_REG. |
| 14:0 | / | / | / |

### 3.17.6.18.    SMC End Address Register(Default Value: 0x00000000)

| Offset: 0x68 | | | Register Name: **DRM_ENDADDR_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:15 | R/W | 0x0 | DRM_ENDADDR_REG. |
| 14:0 | / | / | / |

### 3.17.6.19.    SMC Region Setup Low Register(Default Value: 0x00000000)

| Offset: 0x100+N*0x10(N=0~15) | | | Register Name: **SMC_REGION_SETUP_LO_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:15 | R/W | 0x0 | BASE_ADDRESS_LOW. Controls the base address [31:15] of region<n>. The SMC only permits a region to start at address 0x0, or at a multiple of its region size. For example, if the size of a region is 512MB, and it is not at address 0x0, the only valid settings for this field are: 17'b00100000000000000 17'b01000000000000000 17'b01100000000000000 17'b10000000000000000 17'b10100000000000000 17'b11000000000000000 17'b11100000000000000 |
| 14:0 | / | / | / |

**Note1:**

For region 0, this field is Read Only (RO). The SMC sets the base address of region 0 to 0x0.

The base address should be equal to the DRAM absolutely address.

### 3.17.6.20.    SMC Region Setup High Register(Default Value: 0x00000000)

| Offset: 0x104+N*0x10(N=0~15) | | | Register Name: **SMC_REGION_SETUP_HI_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R/W | 0x0 | BASE_ADDRESS_HIGH The SMC only permits a region to start at address 0x0, or at a multiple of its region size. If you program a region size to be 8GB or more, then the SMC might ignore certain bits depending on the region size. |

### 3.17.6.21.    SMC Region Attributes Register(Default Value: 0x00000000)

| Offset: 0x108+N*0x10(N=0~15) | Register Name: **SMC_REGION_ATTR_REG** |
|---|---|

| Bit | R/W | Default/Hex | Description |
|---|---|---|---|
| 31:28 | R/W | 0x0 | REGION_ATTR_SPN.<br>SP<n>. Permission setting for region <n>. if an AXI transaction occurs to region n, the value in the sp<n> field controls whether the SMC permits the transaction to proceed. . See Table 2 and Table 3. |
| 27:16 | / | / | /. |
| 15:8 | R/W | 0x0 | SUB_REGION_DISABLE.<br>Subregion_disable. Regions are split into eight equal-sized sub-regions, and each bit enables the corresponding subregion to be disabled.<br>Bit [15] = 1 subregion 7 is disabled.<br>Bit [14] = 1 subregion 6 is disabled.<br>Bit [13] = 1 subregion 5 is disabled.<br>Bit [12] = 1 subregion 4 is disabled.<br>Bit [11] = 1 subregion 3 is disabled.<br>Bit [10] = 1 subregion 2 is disabled.<br>Bit [9] = 1 subregion 1 is disabled.<br>Bit [8] = 1 subregion 0 is disabled. |
| 7 | / | / | / |
| 6:1 | R/W | 0x0 | REGION_ATTR_SIZE.<br>Size<n>. Size of region<n>, see Table 4 for detail. |
| 0 | R/W | 0x0 | REGION_ATTR_EN.<br>EN<n>. Enable for region<n>.<br>0 = region < n> is disabled.<br>1 = region < n> is enabled. |

# 3.18. Secure Memory Touch Arbiter

## 3.18.1. Overview

Secure Memory Touch Arbiter provides a software interface to the protection bits in a secure system in a TrustZone design. It provides system flexibility that enables to configure different areas of memory as secure or non-secure.

The SMTA includes the following features:
- It has protection bits to enable you to program some areas of memory as secure or non –secure.

## 3.18.2. Functionalities Description

### 3.18.2.1. Typical Applications

The SMTA provides a software interface to set up memory areas as secure or non-secure. It does this in two ways:
- Programmable protection bits that can be allocated to areas of memory as determined by an external decoder
- Programmable region size value for use by an AXI TrustZone Memory Adapter.

### 3.18.2.2. SMTA Configuration Table

| Register | Bit | SMTA0 | SMTA1 | SMTA2 |
|---|---|---|---|---|
| | | Module Name | Module Name | Module Name |
| **SMTA DECPORTx** (x=0,1,2) | [0] | / | NAND | / |
| | [1] | I2C0 | DMA | / |
| | [2] | I2C1 | SS | / |
| | [3] | SPI0 | SRAM A1 | / |
| | [4] | SPI1 | USB_DRD | DE |
| | [5] | GPIO | USB0 | / |
| | [6] | / | DRAMC | / |
| | [7] | SD/eMMC0 | / | / |

### 3.18.3. SMTA Register List

| Module Name | Base Address |
|---|---|
| SMTA | 0x01C23400 |

| Register Name | Offset | Description |
|---|---|---|
| SMTA_R0SIZE_REG | 0x0 | SMTA R0SIZE Register |
| SMTA_DECPORT0_STA_REG | 0x4 | SMTA Decode Port0 Status Register |
| SMTA_DECPORT0_SET_REG | 0x8 | SMTA Decode Port0 Set Register |
| SMTA_DECPORT0_CLR_REG | 0xC | SMTA Decode Port0 Clear Register |
| SMTA_DECPORT1_STA_REG | 0x10 | SMTA Decode Port1 Status Register |
| SMTA_DECPORT1_SET_REG | 0x14 | SMTA Decode Port1 Set Register |
| SMTA_DECPORT1_CLR_REG | 0x18 | SMTA Decode Port1 Clear Register |
| SMTA_DECPORT2_STA_REG | 0x1C | SMTA Decode Port2 Status Register |
| SMTA_DECPORT2_SET_REG | 0x20 | SMTA Decode Port2 Set Register |
| SMTA_DECPORT2_CLR_REG | 0x24 | SMTA Decode Port2 Clear Register |

### 3.18.4. SMTA Register Description

#### 3.18.4.1. SMTA R0SIZE Register(Default Value: 0x00000018)

| Offset: 0x0 | | | Register Name: **SMTA_R0SIZE_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:10 | / | / | /. |
| 9:0 | RO | 0x18 | SEC_RAM_SIZE.<br>Secure RAM region size in 4KB step.<br>0x000: = no secure region<br>0x001: = 4KB secure region<br>0x002: = 8KB secure region<br>0x003: = 12KB secure region<br>0x004: = 16KB secure region<br>.......<br>0x010: = 64KB secure region<br>.........<br>0x1FF: = 2044KB secure region<br>0x200 or above sets the entire RAM to secure regardless of size. |

### 3.18.4.2. SMTA DECPORT0 Status Register(Default Value: 0x00000000)

| Offset: 0x4 | | | Register Name: **SMTA_DECPORT0_STA_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:8 | / | / | / |
| 7:0 | RO | 0x0 | STA_DEC_PROT0_OUT.<br>Show the status of the decode protection output:<br>0: = Decode region corresponding to the bit is secure<br>1: = Decode region corresponding to the bit is non-secure.<br>There is one bit of the register for each protection output (See the SMTA Configuration Table in detail). |

### 3.18.4.3. SMTA DECPORT0 Set Register(Default Value: 0x00000000)

| Offset: 0x8 | | | Register Name: **SMTA_DECPORT0_SET_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:8 | / | / | /. |
| 7:0 | WO | 0x0 | SET_DEC_PORT0_OUT.<br>Sets the corresponding decode protection output:<br>0: = No effect<br>1: = Set decode region to non-secure.<br>There is one bit of the register for each protection output (See the SMTA Configuration Table in detail). |

### 3.18.4.4. SMTA DECPORT0 Clear Register(Default Value: 0x00000000)

| Offset: 0xC | | | Register Name: **SMTA_DECPORT0_CLR_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:8 | / | / | / |
| 7:0 | WO | 0x0 | CLR_DEC_PROT0_OUT.<br>Clears the corresponding decode protection output:<br>0: = No effect<br>1: = Set decode region to secure.<br>There is one bit of the register for each protection output (See the SMTA Configuration Table in detail). |

### 3.18.4.5. SMTA DECPORT1 Status Register(Default Value: 0x00000000)

| Offset: 0x10 | | | Register Name: **SMTA_DECPORT1_STA_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |

| 31:8 | / | / | / |
|------|---|---|---|
| 7:0 | RO | 0x0 | STA_DEC_PROT1_OUT.<br>Show the status of the decode protection output:<br>0: = Decode region corresponding to the bit is secure<br>1: = Decode region corresponding to the bit is non-secure.<br>There is one bit of the register for each protection output (See the SMTA Configuration Table in detail). |

### 3.18.4.6. SMTA DECPORT1 Set Register(Default Value: 0x00000000)

| Offset: 0x14 | | | Register Name: **SMTA_DECPORT1_SET_REG** |
|------|------|------|------|
| Bit | R/W | Default/Hex | Description |
| 31:8 | / | / | / |
| 7:0 | WO | 0x0 | SET_DEC_PORT1_OUT.<br>Sets the corresponding decode protection output:<br>0: = No effect<br>1: = Set decode region to non-secure.<br>There is one bit of the register for each protection output (See the SMTA Configuration Table in detail). |

### 3.18.4.7. SMTA DECPORT1 Clear Register(Default Value: 0x00000000)

| Offset: 0x18 | | | Register Name: **SMTA_DECPORT1_CLR_REG** |
|------|------|------|------|
| Bit | R/W | Default/Hex | Description |
| 31:8 | / | / | / |
| 7:0 | WO | 0x0 | CLR_DEC_PROT1_OUT.<br>Clears the corresponding decode protection output:<br>0: = No effect<br>1: = Set decode region to secure.<br>There is one bit of the register for each protection output (See the SMTA Configuration Table in detail). |

### 3.18.4.8. SMTA DECPORT2 Status Register(Default Value: 0x00000000)

| Offset: 0x1C | | | Register Name: **SMTA_DECPORT2_STA_REG** |
|------|------|------|------|
| Bit | R/W | Default/Hex | Description |
| 31:8 | / | / | / |
| 7:0 | RO | 0x0 | STA_DEC_PROT2_OUT.<br>Show the status of the decode protection output:<br>0: = Decode region corresponding to the bit is secure<br>1: = Decode region corresponding to the bit is non-secure. |

| | | | There is one bit of the register for each protection output (See the SMTA Configuration Table in detail). |

### 3.18.4.9.  SMTA DECPORT2 Set Register(Default Value: 0x00000000)

| Offset: 0x20 | | | Register Name: **SMTA_DECPORT2_SET_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:8 | / | / | / |
| 7:0 | WO | 0x0 | SET_DEC_PORT2_OUT. Sets the corresponding decode protection output: 0: = No effect 1: = Set decode region to non-secure. There is one bit of the register for each protection output (See the SMTA Configuration Table in detail). |

### 3.18.4.10.    SMTA DECPORT2 Clear Register(Default Value: 0x00000000)

| Offset: 0x24 | | | Register Name: **SMTA_DECPORT2_CLR_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:8 | / | / | / |
| 7:0 | WO | 0x0 | CLR_DEC_PROT2_OUT. Clears the corresponding decode protection output: 0: = No effect 1: = Set decode region to secure. There is one bit of the register for each protection output (See the SMTA Configuration Table in detail). |

# 3.19. Thermal Sensor Controller

## 3.19.1. Overview

The thermal sensors have become common elements in wide range of modern system on chip (SOC) platform. Thermal sensors are used to constantly monitor the temperature on the chip.

A83T embeds three thermal sensors in possible hot spots on the die, sensor0 located in the CPU Class0, sensor1 located in the CPU Class1 and the sensor2 located in the GPU. The thermal sensor Generates interrupt to SW to lower temperature via DVFS, on reaching a certain thermal threshold.

The Thermal Sensor Controller includes the following features:

- Supports APB 32-bits bus width
- Power supply voltage:1.8V
- Low power dissipation
- Periodic temperature measurement
- Averaging filter for thermal sensor reading
- Support over-temperature protection interrupt and over-temperature alarm interrupt

## 3.19.2. Clock and Timing Requirements

CLK_IN = 24MHz
Conversion Time = 1/(24MHz/14Cycles) =0.583us
THERMAL_PER (configured by the value of THERMAL_PER) is must be greater than (ACQ1 + ACQ0+Conversion Time)
**THERMAL_PER > ACQ1 + ACQ0+Conversion Time**:
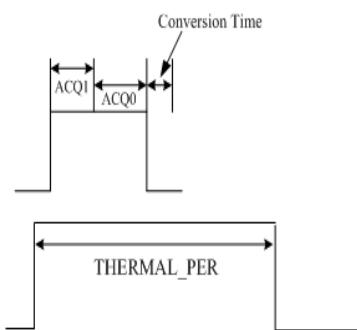


Figure 3-14. Thermal Conversion phase

### 3.19.3.  Thermal Sensor Register List

| Module Name | Base Address |
|---|---|
| Thermal Sensor | 0x01F04000 |

| Register Name | Offset | Description |
|---|---|---|
| THS_CTRL0 | 0x00 | THS Control Register0 |
| THS_CTRL1 | 0x04 | THS Control Register1 |
| ADC_CDAT | 0x14 | ADC calibration data Register |
| THS_CTRL2 | 0x40 | THS Control Register2 |
| THS_INT_CTRL | 0x44 | THS Interrupt Control Register |
| THS_STAT | 0x48 | THS Status Register |
| THS0_ALARM_CTRL | 0x50 | Alarm threshold Control Register0 |
| THS1_ALARM_CTRL | 0x54 | Alarm threshold Control Register1 |
| THS2_ALARM_CTRL | 0x58 | Alarm threshold Control Register2 |
| THS0_SHUTDOWN_CTRL | 0x60 | Shutdown threshold Control Register0 |
| THS1_SHUTDOWN_CTRL | 0x64 | Shutdown threshold Control Register1 |
| THS2_SHUTDOWN_CTRL | 0x68 | Shutdown threshold Control Register2 |
| THS_FILTER | 0x70 | Median filter Control Register |
| THS0_1_CDATA | 0X74 | Thermal Sensor 0 1 Calibration Data |
| THS2_CDATA | 0X78 | Thermal Sensor2 Calibration Data |
| THS0_DATA | 0x80 | THS0 Data Register |
| THS1_DATA | 0x84 | THS1 Data Register |
| THS2_DATA | 0x88 | THS2 Data Register |

### 3.19.4.  Thermal Sensor Register Description

#### 3.19.4.1.  THS Control Register0 (Default Value: 0x00000000)

| Offset: 0x00 | | | Register Name: **THS0_CTRL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:16 | / | / | / |
| 15:0 | R/W | 0x0 | SENSOR_ACQ0<br>ADC acquire time<br>CLK_IN/(N+1) |

### 3.19.4.2.  THS Control Register1 (Default Value: 0x00000000)

| Offset: 0x04 | | | Register Name: **THS1_CTRL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:24 | / | / | / |
| 23 | R/W | 0x0 | Debug control<br>0:Disable the debug mode<br>1: Enable the debug mode |
| 22 | / | / | / |
| 21:20 | R/W | 0x0 | THS_OP_BIAS.<br>THS OP Bias |
| 19:18 | / | / | / |
| 17 | R/W | 0x0 | ADC_CALI_EN.<br>ADC Calibration<br>1: start Calibration, it is clear to 0 after calibration |
| 16:0 | / | / | / |

### 3.19.4.3.  ADC calibration Data Register (Default Value: 0x00000000)

| Offset: 0x14 | | | Register Name: **ADC_CDAT_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:12 | / | / | / |
| 11:0 | R/W | 0x0 | ADC_CDAT.<br>ADC calibration data |

### 3.19.4.4.  THS Control Register2 (Default Value: 0x00040000)

| Offset: 0x40 | | | Register Name: **THS2_CTRL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:16 | R/W | 0x4 | SENSOR_ACQ1.<br>Sensor acquire time<br>CLK_IN/(N+1) |
| 15:3 | / | / | / |
| 2 | R/W | 0x0 | SENSE2_EN.<br>Enable temperature measurement sensor2<br>0:Disable<br>1:Enable |
| 1 | R/W | 0x0 | SENSE1_EN.<br>Enable temperature measurement sensor1<br>0:Disable<br>1:Enable |
| 0 | R/W | 0x0 | SENSE0_EN. |

| | | | Enable temperature measurement sensor0<br>0:Disable<br>1:Enable |

### 3.19.4.5. THS Interrupt Control Register (Default Value: 0x00000000)

| Offset: 0x44 | | | Register Name: **THS_INT_CTRL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:12 | R/W | 0x0 | THERMAL_PER.<br>4096*(n+1)/CLK_IN |
| 11 | / | / | / |
| 10 | R/W | 0x0 | THS2_DATA_IRQ_EN.<br>Selects Temperature measurement data of sensor2<br>0: No select<br>1: Select |
| 9 | R/W | 0x0 | THS1_DATA_IRQ_EN.<br>Selects Temperature measurement data of sensor1<br>0: No select<br>1: Select |
| 8 | R/W | 0x0 | THS0_DATA_IRQ_EN.<br>Selects Temperature measurement data of sensor0<br>0: No select<br>1: Select |
| 7 | / | / | / |
| 6 | R/W | 0x0 | SHUT_INT2_EN.<br>Selects shutdown interrupt for sensor2<br>0: No select<br>1: Select |
| 5 | R/W | 0x0 | SHUT_INT1_EN.<br>Selects shutdown interrupt for sensor1<br>0: No select<br>1: Select |
| 4 | R/W | 0x0 | SHUT_INT0_EN.<br>Selects shutdown interrupt for sensor0<br>0: No select<br>1: Select |
| 3 | / | / | / |
| 2 | R/W | 0x0 | ALARM_INT2_EN.<br>Selects Alert interrupt for sensor2<br>0: No select<br>1: Select |
| 1 | R/W | 0x0 | ALARM_INT1_EN.<br>Selects Alert interrupt for sensor1<br>0: No select |

| | | | 1: Select |
|---|---|---|---|
| 0 | R/W | 0x0 | ALARM_INT0_EN. |
| | | | Selects Alert interrupt for sensor0 |
| | | | 0: No select |
| | | | 1: Select |

### 3.19.4.6. THS status Register (Default Value: 0x00000000)

| Offset: 0x48 | | | Register Name: **THS_STAT_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:11 | / | / | / |
| 10 | R/W | 0x0 | THS2_DATA_IRQ _STS. |
| | | | Data interrupt status for sensor2 |
| | | | Write '1' to clear this interrupt or automatic clear if interrupt condition fails |
| 9 | R/W | 0x0 | THS1_DATA_IRQ _STS. |
| | | | Data interrupt status for sensor1 |
| | | | Write '1' to clear this interrupt or automatic clear if interrupt condition fails |
| 8 | R/W | 0x0 | THS0_DATA_IRQ _STS. |
| | | | Data interrupt status for sensor0 |
| | | | Write '1' to clear this interrupt or automatic clear if interrupt condition fails |
| 7 | / | / | / |
| 6 | R/W | 0x0 | SHUT_INT2_STS. |
| | | | Shutdown interrupt status for sensor2 |
| | | | Write '1' to clear this interrupt or automatic clear if interrupt condition fails |
| 5 | R/W | 0x0 | SHUT_INT1_STS. |
| | | | Shutdown interrupt status for sensor1 |
| | | | Write '1' to clear this interrupt or automatic clear if interrupt condition fails |
| 4 | R/W | 0x0 | SHUT_INT0_STS. |
| | | | Shutdown interrupt status for sensor0 |
| | | | Write '1' to clear this interrupt or automatic clear if interrupt condition fails |
| 3 | / | / | / |
| 2 | R/W | 0x0 | ALARM_INT2_STS. |
| | | | Alarm interrupt pending for sensor2 |
| | | | Write '1' to clear this interrupt or automatic clear if interrupt condition fails |
| 1 | R/W | 0x0 | ALARM_INT1_STS. |
| | | | Alarm interrupt pending for sensor1 |
| | | | Write '1' to clear this interrupt or automatic clear if interrupt condition |

| | | | fails |
|---|---|---|---|
| 0 | R/W | 0x0 | ALARM_INT0_STS. |
| | | | Alarm interrupt pending for sensor0 |
| | | | Write '1' to clear this interrupt or automatic clear if interrupt condition |
| | | | fails |

### 3.19.4.7. Alarm threshold Control Register0 (Default Value: 0x051a0644)

| Offset: 0x50 | | | Register Name: **THS0_ALARM_CTRL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:28 | / | / | / |
| 27:16 | R/W | 0x51A | ALARM0_T_HOT. |
| | | | Thermal sensor0 Alarm Threshold for hot temperature |
| 15:12 | / | / | / |
| 11:0 | R/W | 0x644 | ALARM0_T_HYST |
| | | | Thermal sensor0 Alarm threshold for hysteresis temperature |

### 3.19.4.8. Alarm threshold Control Register1 (Default Value: 0x051a0644)

| Offset: 0x54 | | | Register Name: **THS1_ALARM_CTRL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:28 | / | / | / |
| 27:16 | R/W | 0x51A | ALARM1_T_HOT. |
| | | | Thermal sensor1 Alarm Threshold for hot temperature |
| 15:12 | / | / | / |
| 11:0 | R/W | 0x644 | ALARM1_T_HYST |
| | | | Thermal sensor1 Alarm threshold for hysteresis temperature |

### 3.19.4.9. Alarm threshold Control Register2 (Default Value: 0x051a0644)

| Offset: 0x58 | | | Register Name: **THS2_ALARM_CTRL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:28 | / | / | / |
| 27:16 | R/W | 0x51A | ALARM2_T_HOT. |
| | | | Thermal sensor2 Alarm Threshold for hot temperature |
| 15:12 | / | / | / |
| 11:0 | R/W | 0x644 | ALARM2_T_HYST |
| | | | Thermal sensor2 Alarm threshold for hysteresis temperature |

### 3.19.4.10. Shutdown threshold Control Register0 (Default Value: 0x03f00000)

| Offset: 0x60 | | | Register Name: **THS0_SHUTDOWN_CTRL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:28 | / | / | / |
| 27:16 | R/W | 0x3F0 | SHUT0_T_HOT. Thermal sensor0 Shutdown Threshold for hot temperature |
| 15:0 | / | / | / |

### 3.19.4.11. Shutdown threshold Control Register1 (Default Value: 0x03f00000)

| Offset: 0x64 | | | Register Name: **THS1_SHUTDOWN_CTRL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:28 | / | / | / |
| 27:16 | R/W | 0x3F0 | SHUT1_T_HOT. Thermal sensor1 Shutdown Threshold for hot temperature |
| 15:0 | / | / | / |

### 3.19.4.12. Shutdown threshold Control Register2(Default Value: 0x03f00000)

| Offset: 0x68 | | | Register Name: **THS2_SHUTDOWN_CTRL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:28 | / | / | / |
| 27:16 | R/W | 0x3F0 | SHUT2_T_HOT. Thermal sensor2 Shutdown Threshold for hot temperature |
| 15:0 | / | / | / |

### 3.19.4.13. Average filter Control Register (Default Value: 0x00000001)

| Offset: 0x70 | | | Register Name: **THS_FILTER _REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:3 | / | / | / |
| 2 | R/W | 0x0 | FILTER_EN. Filter Enable 0: Disable 1: Enable |
| 1:0 | R/W | 0x1 | FILTER_TYPE. Average Filter Type 00: 2 01: 4 |

| | | 10: 8 |
|---|---|---|
| | | 11: 16 |

### 3.19.4.14.  Thermal Sensor 0&1 calibration Data Register (Default Value: 0x08000800)

| Offset: 0x74 | | | Register Name: **THS0_1_CDATA _REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:28 | / | / | / |
| 27:16 | R/W | 0x800 | THES1_CDATA. Thermal Sensor1 calibration data |
| 15:12 | / | / | / |
| 11:0 | R/W | 0x800 | THES0_CDATA. Thermal Sensor0 calibration data |

### 3.19.4.15.  Thermal Sensor 2 calibration Data Register (Default Value: 0x00000800)

| Offset: 0x78 | | | Register Name: **THS2_CDATA _REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:12 | / | / | / |
| 11:0 | R/W | 0x800 | THES2_CDATA. Thermal Sensor2 calibration data |

### 3.19.4.16.  THS0 Data Register (Default Value: 0x00000000)

| Offset: 0x80 | | | Register Name: **THS0_DATA _REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:12 | / | / | / |
| 11:0 | R | 0x0 | THS0_DATA. Temperature measurement data of sensor0 |

### 3.19.4.17.  THS1 Data Register (Default Value: 0x00000000)

| Offset: 0x84 | | | Register Name: **THS1_DATA _REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:12 | / | / | / |
| 11:0 | R | 0x0 | THS1_DATA. Temperature measurement data of sensor1 |

**3.19.4.18.    THS2 Data Register (Default Value: 0x00000000)**

| Offset: 0x88 | | | Register Name: **THS2_DATA _REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:12 | / | / | / |
| 11:0 | R | 0x0 | THS2_DATA. Temperature measurement data of sensor2 |

## 3.19.5.    Programming Guidelines

1)   Timing must be like this: THERMAL_PER > ACQ1 + ACQ0+Conversion Time
2)   Configure THS Interrupt Control Register to set the THERMAL_PER and IRQ
3)   Configure the Alarm threshold Control Register and Shutdown threshold Control Register to set the ALARM_T_HOT and SHUT_T_HOT
4)   Configure THS Control Register to set the SENSOR_ACQ and enable the sensor
5)   The real temperature value of each sensor is Tem, then

$$T = (2719\text{-}Tem)/14.186$$

Reading back the temperature from the temperature value register requires a 2-byte read. Use 12-bit temperature data format.

# 3.20. LRADC

## 3.20.1. Overview
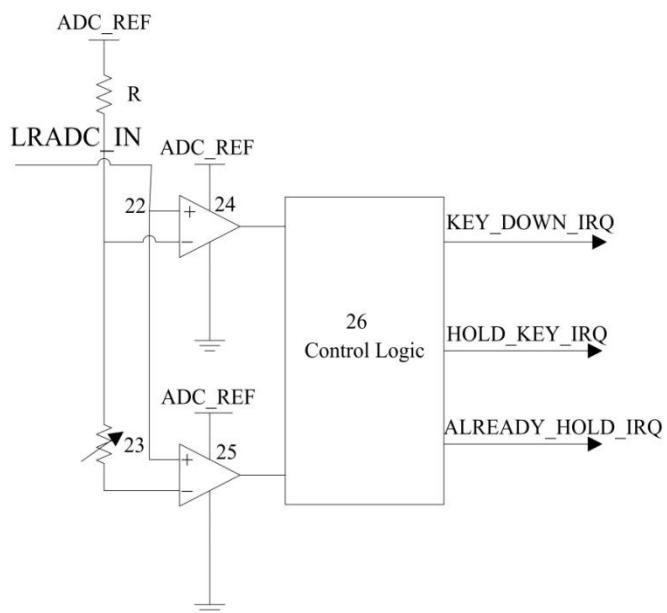
LRADC is 6-bit resolution ADC for key application. The LRADC can work up to 250Hz conversion rate.

The LRADC includes the following features:

- Supports APB 32-bits bus width
- Support interrupt
- Support Hold Key and General Key
- Support Single Key and Continue Key mode
- Support 6-bits resolution
- Voltage input range between 0V to 1.35V
- Sample rate up to 250Hz

## 3.20.2. Operation Principle

The LRADC converted data can accessed by interrupt and polling method. If software can't access the last converted data instantly, the new converted data would update the old one at new sampling data.



When ADC_IN Signal change from 1.8V to less than 1.35V (Level A), the comparator24 send first interrupt to control

logic; When ADC_IN Signal change from 1.35V to less than certain level (Program can set), the comparator25 give second interrupt. If the control Logic get the first interrupt, In a certain time range (program can set), doesn't get second interrupt, it will send hold key interrupt to the host; If the control Logic get the first interrupt, In a certain time range (program can set), get second interrupt, it will send key down interrupt to the host; If the control logic only get the second interrupt, doesn't get the first interrupt, it will send already hold interrupt to the host.

The LRADC have three mode, Normal Mode、Single Mode and Continue Mode. Normal mode is that the LRADC will report the result data of each convert all the time when the key is down. Single Mode is that the LRADC will only report the first convert result data when the key is down. Continue Mode is that the LRADC will report one of 8*(N+1) (N is program can set) sample convert result data when key is down.

The LRADC is support four sample rate such as 250Hz、125Hz、62.5Hz and 32.25Hz, you can configure the value of ADC_SAMPLE_RATE to select the fit sample rate.

## 3.20.3.  LRADC Register List

| Module Name | Base Address |
| --- | --- |
| LRADC | 0x01F03C00 |

| Register Name | Offset | Description |
| --- | --- | --- |
| LRADC_CTRL | 0x00 | LRADC Control Register |
| LRADC_INTC | 0x04 | LRADC Interrupt Control Register |
| LRADC_INTS | 0x08 | LRADC Interrupt Status Register |
| LRADC_DATA | 0x0C | LRADC Data Register |

## 3.20.4.  LRADC Register Description

### 3.20.4.1.  LRADC Control Register (Default Value: 0x01000168)

| Offset: 0x00 | | | Register Name: **LRADC_CTRL_REG** |
| --- | --- | --- | --- |
| Bit | R/W | Default/Hex | Description |
| 31: 24 | R/W | 0x1 | FIRST_CONVERT_DLY.<br>ADC First Convert Delay setting, ADC conversion is delayed by n samples |
| 23:22 | R/W | 0x0 | Reserved to 0 |
| 21:20 | / | / | / |

| 19:16 | R/W | 0x0 | CONTINUE_TIME_SELECT.<br>Continue Mode time select, one of 8*(N+1) sample as a valuable sample data |
|---|---|---|---|
| 15:14 | / | / | / |
| 13:12 | R/W | 0x0 | KEY_MODE_SELECT.<br>Key Mode Select:<br>00: Normal Mode<br>01: Single Mode<br>10: Continue Mode |
| 11:8 | R/W | 0x1 | LEVELA_B_CNT.<br>Level A to Level B time threshold select, judge ADC convert value in level A to level B in n+1 samples |
| 7 | R/W | 0X0 | ADC_HOLD_KEY_EN<br>ADC Hold Key Enable<br>0: Disable<br>1: Enable |
| 6 | R/W | 0x1 | ADC_HOLD_EN.<br>ADC Sample hold Enable<br>0: Disable<br>1: Enable |
| 5:4 | R/W | 0x2 | LEVELB_VOL.<br>Level B Corresponding Data Value setting (the real voltage value)<br>00: 0x3C (~1.285v)<br>01: 0x39 (~1.221v)<br>10: 0x36 (~1.157v)<br>11: 0x33 (~1.092v) |
| 3:2 | R/W | 0x2 | ADC_SAMPLE_RATE.<br>ADC Sample Rate<br>00: 250 Hz<br>01: 125 Hz<br>10: 62.5 Hz<br>11: 32.25 Hz |
| 1 | / | / | / |
| 0 | R/W | 0x0 | ADC_EN.<br>ADC enable<br>0: Disable<br>1: Enable |

### 3.20.4.2. LRADC Interrupt Control Register (Default Value: 0x00000000)

| Offset: 0x04 | | | Register Name: **LRADC_INTC_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:5 | / | / | / |
| 4 | R/W | 0x0 | ADC_KEYUP_IRQ_EN. |

| | | | ADC Key Up IRQ Enable<br>0: Disable<br>1: Enable |
|---|---|---|---|
| 3 | R/W | 0x0 | ADC_ALRDY_HOLD_IRQ_EN.<br>ADC Already Hold IRQ Enable<br>0: Disable<br>1: Enable |
| 2 | R/W | 0x0 | ADC_HOLD_IRQ_EN.<br>ADC Hold Key IRQ Enable<br>0: Disable<br>1: Enable |
| 1 | R/W | 0x0 | ADC_KEYDOWN_EN<br>ADC Key Down Enable<br>0: Disable<br>1: Enable |
| 0 | R/W | 0x0 | ADC_DATA_IRQ_EN.<br>ADC Data IRQ Enable<br>0: Disable<br>1: Enable |

### 3.20.4.3. LRADC Interrupt Status Register (Default Value: 0x00000000)

| Offset: 0x08 | | | Register Name:**LRADC_INTS_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:5 | / | / | / |
| 4 | R/W | 0x0 | ADC_KEYUP_PENDING.<br>ADC Key up pending Bit<br>When general key pull up, it the corresponding interrupt is enabled.<br>0: No IRQ<br>1: IRQ Pending<br>Notes: Writing 1 to the bit will clear it and its corresponding interrupt if the interrupt is enable |
| 3 | R/W | 0x0 | ADC_ALRDY_HOLD_PENDING.<br>ADC Already Hold Pending Bit<br>When hold key pull down and pull the general key down, if the corresponding interrupt is enabled.<br>0: No IRQ<br>1: IRQ Pending<br>Notes: Writing 1 to the bit will clear it and its corresponding interrupt if the interrupt is enable |
| 2 | R/W | 0x0 | ADC_HOLDKEY_PENDING.<br>ADC Hold Key pending Bit<br>When Hold key pull down, the status bit is set and the interrupt line is set if the corresponding interrupt is enabled. |

| | | | 0: NO IRQ |
|---|---|---|---|
| | | | 1: IRQ Pending |
| | | | Notes: Writing 1 to the bit will clear it and its corresponding interrupt if the interrupt is enable. |
| 1 | R/W | 0x0 | ADC_KEYDOWN_PENDING. |
| | | | ADC Key Down IRQ Pending Bit |
| | | | When General key pull down, the status bit is set and the interrupt line is set if the corresponding interrupt is enabled. |
| | | | 0: No IRQ |
| | | | 1: IRQ Pending |
| | | | Notes: Writing 1 to the bit will clear it and its corresponding interrupt if the interrupt is enable. |
| 0 | R/W | 0x0 | ADC_DATA_PENDING. |
| | | | ADC Data IRQ Pending Bit |
| | | | 0: No IRQ |
| | | | 1: IRQ Pending |
| | | | Notes: Writing 1 to the bit will clear it and its corresponding interrupt if the interrupt is enable. |

### 3.20.4.4. LRADC Data Register Register (Default Value: 0x00000000)

| Offset: 0x0C | | | Register Name: **LRADC_DATA_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:6 | / | / | / |
| 5:0 | R | 0x0 | LRADC_DATA. |
| | | | LRADC Data |

## 3.21. R_timer

### 3.21.1. Overview

R_timer 0 and R_timer 1 are general timers, and use the Internal OSC / 512 or OSC24M as source clock. R_timer 0 and R_timer 1 share a programmable 3-bit pre-scale that provides the division of the clock source. They can work in auto-reload mode or no-reload mode. When the current value in *Current Value Register* is counting down to zero, the timer will generate interrupt if set interrupt enable bit.

### 3.21.2. R_timer Register List

| Module Name | Base Address |
| --- | --- |
| R_timer | 0x01F00800 |

| Register Name | Offset | Description |
| --- | --- | --- |
| RTMR_IRQ_EN_REG | 0x0 | R_timer IRQ Enable Register |
| RTMR_IRQ_STA_REG | 0x4 | R_timer IRQ Status Register |
| RTMR0_CTRL_REG | 0x20 | R_timer 0 Control Register |
| RTMR0_INTV_VALUE_REG | 0x24 | R_timer 0 Interval Value Register |
| RTMR0_CUR_VALUE_REG | 0x28 | R_timer 0 Current Value Register |
| RTMR1_CTRL_REG | 0x40 | R_timer 1 Control Register |
| RTMR1_INTV_VALUE_REG | 0x44 | R_timer 1 Interval Value Register |
| RTMR1_CUR_VALUE_REG | 0x48 | R_timer 1 Current Value Register |

### 3.21.3. R_timer Register Description

#### 3.21.3.1. R_timer IRQ Enable Register (Default Value: 0x00000000)

| Offset: 0x00 | | | Register Name: **RTMR_IRQ_EN_REG** |
| --- | --- | --- | --- |
| Bit | R/W | Default/Hex | Description |

| 31:2 | / | / | / |
|---|---|---|---|
| 1 | R/W | 0x0 | RTMR1_IRQ_EN.<br>R_timer 1 Interrupt Enable.<br>0: No effect;<br>1: R_timer 1 Interval Value reached interrupt enable. |
| 0 | R/W | 0x0 | RTMR0_IRQ_EN.<br>R_timer 0 Interrupt Enable.<br>0: No effect;<br>1: R_timer 0 Interval Value reached interrupt enable. |

### 3.21.3.2. R_timer IRQ Status Register (Default Value: 0x00000000)

| Offset: 0x04 | | | Register Name: **RTMR_IRQ_STA_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:2 | / | / | / |
| 1 | R/W | 0x0 | RTMR1_IRQ_PEND.<br>R_timer 1 IRQ Pending. Set 1 to the bit will clear it.<br>0: No effect;<br>1: Pending, R_timer 1 interval value is reached. |
| 0 | R/W | 0x0 | RTMR0_IRQ_PEND.<br>R_timer 0 IRQ Pending. Set 1 to the bit will clear it.<br>0: No effect;<br>1: Pending, R_timer 0 interval value is reached. |

### 3.21.3.3. R_timer 0 Control Register (Default Value: 0x00000004)

| Offset: 0x20 | | | Register Name: **RTMR0_CTRL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:8 | / | / | / |
| 7 | R/W | 0x0 | RTMR0_MODE.<br>R_timer 0 mode.<br>0: Continuous mode. When interval value reached, the R_timer will not disable automatically.<br>1: Single mode. When interval value reached, the R_timer will disable automatically. |
| 6:4 | R/W | 0x0 | RTMR0_CLK_PRES.<br>Select the pre-scale of R_timer 0 clock source.<br>000: /1<br>001: /2<br>010: /4<br>011: /8<br>100: /16 |

| | | | 101: /32 |
|---|---|---|---|
| | | | 110: /64 |
| | | | 111: /128 |
| 3:2 | R/W | 0x1 | RTMR0_CLK_SRC. |
| | | | 00: Internal OSC / 512 |
| | | | 01: OSC24M. |
| | | | Others: / |
| 1 | R/W | 0x0 | RTMR0_RELOAD. |
| | | | R_timer 0 Reload. |
| | | | 0: No effect, 1: Reload R_timer 0 Interval value. |
| | | | After the bit is set, it can not be written again before it's cleared automatically. |
| 0 | R/W | 0x0 | RTMR0_EN. |
| | | | R_timer 0 Enable. |
| | | | 0: Stop/Pause, 1: Start. |
| | | | If the R_timer is started, it will reload the interval value to internal register, and the current counter will count from interval value to 0. |
| | | | If the current counter does not reach the zero, the R_timer enable bit is set to '0'; the current value counter will pause. At least wait for 2 cycles, the start bit can be set to 1. |
| | | | In R_timer pause state, the interval value register can be modified. If the R_timer is started again, and the Software hope the current value register to down-count from the new interval value, the reload bit and the enable bit should be set to 1 at the same time. |

### 3.21.3.4. R_timer 0 Interval Value Register

| Offset: 0x24 | | | Register Name: **RTMR0_INTV_VALUE_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R/W | x | RTMR0_INTV_VALUE. |
| | | | R_timer 0 Interval Value. |

**Note**: The value setting should consider the system clock and the R_timer clock source.

### 3.21.3.5. R_timer 0 Current Value Register (Default Value: 0x00000000)

| Offset: 0x28 | | | Register Name: **RTMR0_ CUR_VALUE_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R/W | 0x0 | RTMR0_CUR_VALUE. |
| | | | R_timer 0 Current Value. |

**Note**: R_timer 0 current value is a 32-bit down-counter (from interval value to 0).

**3.21.3.6. R_timer 1 Control Register (Default Value: 0x00000004)**

| Offset: 0x40 | | | Register Name: **RTMR1_ CTRL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:8 | / | / | / |
| 7 | R/W | 0x0 | RTMR1_MODE. <br> R_timer 1 mode. <br> 0: Continuous mode. When interval value reached, the R_timer will not disable automatically. <br> 1: Single mode. When interval value reached, the R_timer will disable automatically. |
| 6:4 | R/W | 0x0 | RTMR1_CLK_PRES. <br> Select the pre-scale of R_timer 1 clock source. <br> 000: /1 <br> 001: /2 <br> 010: /4 <br> 011: /8 <br> 100: /16 <br> 101: /32 <br> 110: /64 <br> 111: /128 |
| 3:2 | R/W | 0x1 | RTMR1_CLK_SRC. <br> 00: Internal OSC / 512 <br> 01: OSC24M. <br> 10: / <br> 11: / |
| 1 | R/W | 0x0 | RTMR1_RELOAD. <br> R_timer 1 Reload. <br> 0: No effect, 1: Reload R_timer 1 Interval value. <br> After the bit is set, it can not be written again before it's cleared automatically. |
| 0 | R/W | 0x0 | RTMR1_EN. <br> R_timer 1 Enable. <br> 0: Stop/Pause, 1: Start. <br> If the R_timer is started, it will reload the interval value to internal register, and the current counter will count from interval value to 0. <br> If the current counter does not reach the zero, the R_timer enable bit is set to '0'; the current value counter will pause. At least wait for 2 cycles, the start bit can be set to 1. <br> In R_timer pause state, the interval value register can be modified. If the R_timer is started again, and the Software hope the current value register to down-count from the new interval value, the reload bit and the enable bit should be set to 1 at the same time. |

### 3.21.3.7. R_timer 1 Interval Value Register

| Offset: 0x44 | | | Register Name: **RTMR1_ INTV_VALUE _REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R/W | x | RTMR1_INTV_VALUE. <br> R_timer 1 Interval Value. |

**Note**: The value setting should consider the system clock and the R_timer clock source.

### 3.21.3.8. R_timer 1 Current Value Register (Default Value: 0x00000000)

| Offset: 0x48 | | | Register Name: **RTMR1_CUR_VALUE _REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R/W | 0x0 | RTMR1_CUR_VALUE. <br> R_timer 1 Current Value. |

**Note**: R_timer 1 current value is a 32-bit down-counter (from interval value to 0).

## 3.22. R_watchdog

### 3.22.1. Overview

The R_watchdog is used to resume the controller operation when it had been disturbed by malfunctions such as noise and system errors. It features a down counter that allows a watchdog period of up to 16 seconds. It can generate a general reset or interrupt request. The watchdog generates the reset signal to reset CPUS or the whole system.

### 3.22.2. R_watchdog Register List

| Module Name | Base Address |
|---|---|
| R_watchdog | 0x01F01000 |

| Register Name | Offset | Description |
|---|---|---|
| RWDOG_IRQ_EN_REG | 0x0 | R_watchdog IRQ Enable Register |
| RWDOG_IRQ_STA_REG | 0x4 | R_watchdog Status Register |
| RWDOG_CTRL_REG | 0x10 | R_watchdog Control Register |
| RWDOG_CFG_REG | 0x14 | R_watchdog Configuration Register |
| RWDOG_MODE_REG | 0x18 | R_watchdog Mode Register |

### 3.22.3. R_watchdog Register Description

#### 3.22.3.1. R_watchdog IRQ Enable Register (Default Value: 0x00000000)

| Offset: 0x00 | | | Register Name: **RWDOG_IRQ_EN_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:1 | / | / | / |
| 0 | R/W | 0x0 | RWDOG_IRQ_EN.<br>R_watchdog Interrupt Enable.<br>0: No effect, 1: R_watchdog interrupt enable. |

### 3.22.3.2. R_watchdog Status Register (Default Value: 0x00000000)

| Offset: 0x04 | | | Register Name: **RWDOG_IRQ_STA_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:1 | / | / | / |
| 0 | R/W | 0x0 | RWDOG_IRQ _PEND.<br>R_watchdog IRQ Pending. Set 1 to the bit will clear it.<br>0: No effect, 1: Pending, r_watchdog interval value is reached. |

### 3.22.3.3. R_watchdog Control Register (Default Value: 0x00000000)

| Offset: 0x10 | | | Register Name: **RWDOG_CTRL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:13 | / | / | / |
| 12:1 | R/W | 0x0 | RWDOG_KEY_FIELD.<br>R_watchdog Key Field.<br>Should be written at value 0xA57. Writing any other value in this field aborts the write operation. |
| 0 | R/W | 0x0 | RWDOG_RESTART.<br>R_watchdog Restart.<br>0: No effect, 1: Restart the r_watchdog. |

### 3.22.3.4. R_watchdog Configuration Register (Default Value: 0x00000000)

| Offset: 0x14 | | | Register Name: **RWDOG_CFG_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:9 | / | / | / |
| 8 | R/W | 0x0 | RWDOG_CLK_SRC.<br>Select r_watchdog clock sources.<br>0: 24M / 750<br>1: Internal OSC / 512 |
| 7:2 | / | / | / |
| 1:0 | R/W | 0x1 | RWDOG_CONFIG.<br>R_watchdog generates a reset signal<br>00: /<br>01: to whole system<br>10: only interrupt<br>11: / |

**3.22.3.5. R_watchdog Mode Register (Default Value: 0x00000000)**

| Offset: 0x18 | | | Register Name: **RWDOG_MODE_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:8 | / | / | / |
| 7:4 | R/W | 0x0 | RWDOG_INTV_VALUE.<br><br>R_watchdog Interval Value. When the clock source is **24M/750**, the time of 16000 cycles is equal to 0.5 sec, and when the clock source is **Internal OSC / 512**, the time of 16000 cycles is based on the number of **Internal OSC / 512.**<br>0000: 16000 cycles<br>0001: 32000 cycles<br>0010: 64000 cycles<br>0011: 96000 cycles<br>0100: 128000 cycles<br>0101: 160000 cycles<br>0110: 192000 cycles<br>0111: 256000 cycles<br>1000: 320000 cycles<br>1001: 384000 cycles<br>1010: 448000 cycles<br>1011: 512000 cycles<br>others: / |
| 3:1 | / | / | / |
| 0 | R/W | 0x0 | RWDOG_EN.<br>R_watchdog Enable.<br>0: No effect;<br>1: Enable the r_watchdog. |

# 3.23. R_PRCM

## 3.23.1. Overview
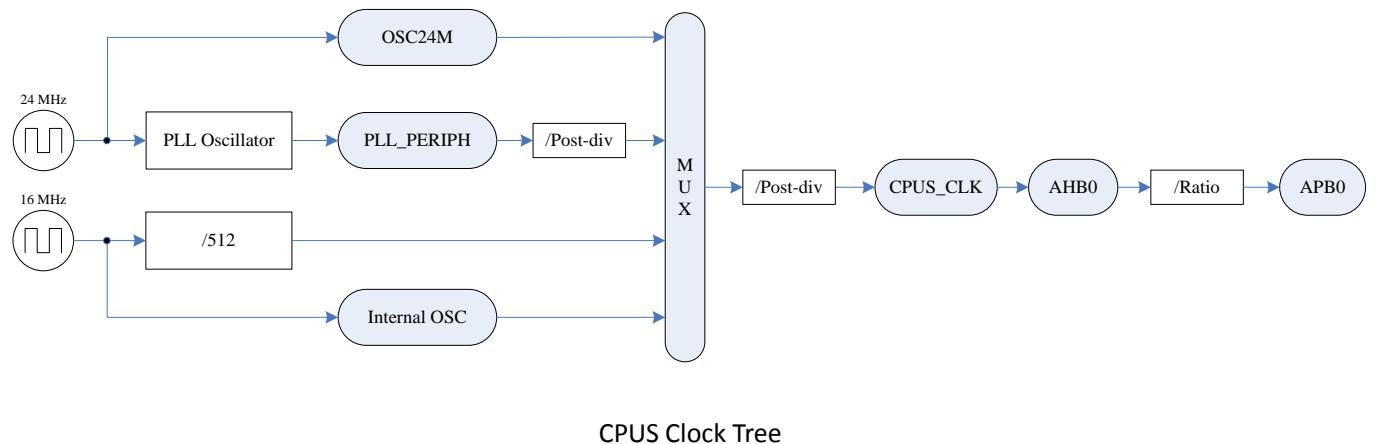
The R_PRCM module is one of the most import design aspects in this system. It provides a versatile supporting multiple power-management techniques. And it also manages the gating and enabling of the clocks to the device modules.

The system-level reset management provides correct reset routing and sequencing when one or more devices are stacked together in the same package. The device-level reset management provides reset routing to relevant devices, such as r_timer, r_uart, r_dma and so on.

The R_PRCM includes the following features:

- Support VDD_SYS power domain modules reset control
- Support CPUS clock configuration
- Support devices clock gating, division and software reset in the CPUS domain
- Support CPUX、GPU、VDD_SYS power off gating control
- Support CPUX power switch control
- Support RAM Congiguration and Test Function
- BROM Output

## 3.23.2. Block Diagram



CPUS Clock Tree

### 3.23.3. Operation Mode

#### 3.23.3.1. HOSC

24MHz clock has two clock sources, one is from an external 24MHz crystal oscillator input, the other one is about 16MHz internal RC oscillator. In the super standby mode, CPUS uses 16MHz clock source to save the power consumption of 24MHz crystal, also make CPUS operate at a high frequency, fewer super standby wake-up time, improve the response speed. There is a simple operation process:

(1)  Enable 16MHz RC oscillator: The bit[0] of **OSC24M Control Register** is set to 1;
(2)  Switch HOSC clock source: The bit[1] of **OSC24M Control Register** is set to 1;
(3)  Close the external 24MHz oscillator: The bit[2] of **PLL Control Register 1** is cleared;

The next step can be operated before the previous step has been taken effect.

#### 3.23.3.2. Pad Hold

In the super standby mode,GPIO B/C/D/E/F/G/H of VDD_SYS domain can hold a particular output state through GPIO pad hold function,but VCC power of GPIO can not he closed at this time;after exiting the super standby,if you need to use these GPIOs,need to be reconfigured.In power down,GPIO L can hold status through power off hold function;after system powered up at each time,Using GPIO L need to reset **R_PIO Hold Control Register**,so that avoid the previous power down residual configuration. Simple operation is as the following:

(1)  GPIO B/C/D/E/F/G/H pad hold: the bit[12] of **VDD_SYS Power Off Gating Register** can set to 1.
(2)  GPIO L pad hold: Please refer to **R_PIO Hold Control Register**.

#### 3.23.3.3. Power Switch

Each CPU power switch is composed by eight chains, there are chain0~chain7, which respectively correspond bit[0]~ bit[7] of **Cluster CPU Power Switch Control Register** .Chain0~chain6 is belong to daisy chain, mainly for CPU charge to reduce the impact of the PMU power load.Chain7 is belong to fishbone chain, mainly for CPU core circuitry power supply.When CPU starts to power up,the corresponding value of **Cluster CPU Power Switch Control Register** should be 0xFF,namely power off;then successively write 0xFE、0xFC、0xF8、0xF0、0x00 to the register, there is a 10us delay after each write operation completed, then it starts to the next write operation. When the CPU power down, you can be directly written 0xFF to the register.

### 3.23.3.4.  OSC24 Control

RTC module of CPUS domain contains a RC oscillator about 16MHz.The frequency is divided to SS module for generating a true random number. At the same time,24MHz clock source can be switched to 16MHz,which need to effectively configure OSC24M control Register.

## 3.23.4.  R_PRCM Register List

| Module Name | Base Address |
|---|---|
| R_PRCM | 0x01F01400 |

| Register Name | Offset | Description |
|---|---|---|
| CPUS_CLK_REG | 0x0000 | CPUS Clock Register |
| APB0_CFG_REG | 0x000C | APB0 Configuration Register |
| APB0_CLK_GATING_REG | 0x0028 | APB0 Clock Gating Register |
| PLL_CTRL_REG1 | 0x0044 | PLL Control Register 1 |
| R_CIR_RX_CLK_REG | 0x0054 | R_CIR_RX Clock Register |
| APB0_SOFT_RST_REG | 0x00B0 | APB0 Software Reset Register |
| C0CPUX_PWROFF_GATING_REG | 0x0100 | Cluster0 CPUX Power Off Gating Register |
| C1CPUX_PWROFF_GATING_REG | 0x0104 | Cluster1 CPUX Power Off Gating Register |
| VDD_SYS_PWROFF_GATING_REG | 0x0110 | VDD_SYS Power Off Gating Register |
| GPU_PWROFF_GATING_REG | 0x0118 | GPU Power Off Gating Register |
| VDD_SYS_PWROFF_RST_REG | 0x0120 | VDD_SYS Power Domain Reset Register |
| C0_CPU0_PWR_SWITCH_CTRL | 0x0140 | C0_CPU0 Power Switch Control Register |
| C0_CPU1_PWR_SWITCH_CTRL | 0x0144 | C0_CPU1 Power Switch Control Register |
| C0_CPU2_PWR_SWITCH_CTRL | 0x0148 | C0_CPU2 Power Switch Control Register |
| C0_CPU3_PWR_SWITCH_CTRL | 0x014C | C0_CPU3 Power Switch Control Register |
| C1_CPU0_PWR_SWITCH_CTRL | 0x0150 | C1_CPU0 Power Switch Control Register |
| C1_CPU1_PWR_SWITCH_CTRL | 0x0154 | C1_CPU1 Power Switch Control Register |
| C1_CPU2_PWR_SWITCH_CTRL | 0x0158 | C1_CPU2 Power Switch Control Register |
| C1_CPU3_PWR_SWITCH_CTRL | 0x015C | C1_CPU3 Power Switch Control Register |
| RPIO_HOLD_CTRL_REG | 0x01F0 | R_PIO Hold Control Register |
| OSC24M_CTRL_REG | 0x01F4 | OSC24M Control Register |

## 3.23.5.  R_PRCM Register Description

### 3.23.5.1.  CPUS Clock Register (Default Value: 0x00010000)

| Offset: 0x00 | | | Register Name: **CPUS_CLK_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:18 | / | / | / |
| 17:16 | R/W | 0x1 | CPUS_CLK_SRC_SEL.<br>CPUS Clock Source Select.<br>Internal OSC is about 16M.<br>00:    Internal OSC / 512<br>01:    OSC24M<br>10:    PLL_PERIPH/ CPUS_POST_DIV<br>11:    Internal OSC .<br>Note: CPUS_CLK = CPUS Clock Source / CPUS_CLK_RATIO |
| 15:13 | / | / | / |
| 12:8 | R/W | 0x0 | CPUS_POST_DIV<br>00000: /1<br>00001: /2<br>00010: /3<br>……<br>11111: /32. |
| 7:6 | / | / | / |
| 5:4 | R/W | 0x0 | CPUS_CLK_RATIO<br>00:/1<br>01:/2<br>10:/4<br>11:/8. |
| 3:0 | / | / | / |

### 3.23.5.2.  APB0 Configuration Register (Default Value: 0x00000000)

| Offset: 0x0C | | | Register Name: **APB0_CFG_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:2 | / | / | / |
| 1:0 | R/W | 0x0 | APB0_CLK_RATIO.<br>APB0 Clock Divide Ratio. APB0 clock source is AHB0 clock.<br>00: /1<br>01: /2<br>10: /3<br>11: /4. |

| | | | Note:This clock is used for some special module apbclk (R_UART/R_TWI/R_RSB) |
|---|---|---|---|

### 3.23.5.3. APB0 Clock Gating Register (Default Value: 0x00000080)

| Offset: 0x0028 | | | Register Name: **APB0_CLK_GATING_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:8 | / | / | / |
| 7 | R/W | 0x1 | R_TWD_GATING<br>Gating APB Clock for R_TWD<br>0: Mask<br>1: Pass. |
| 6 | R/W | 0x0 | R_TWI_GATING<br>Gating APB Clock for R_TWI<br>0: Mask<br>1: Pass. |
| 5 | / | / | / |
| 4 | R/W | 0x0 | R_UART_GATING.<br>Gating APB Clock for R_UART<br>0: Mask<br>1: Pass. |
| 3 | R/W | 0x0 | R_RSB_GATING.<br>Gating APB Clock for R_RSB<br>0: Mask<br>1: Pass. |
| 2 | R/W | 0x0 | R_TIMER_GATING.<br>Gating APB Clock for R_TIMER<br>0: Mask<br>1: Pass. |
| 1 | R/W | 0x0 | R_CIR_GATING.<br>Gating APB Clock for R_CIR-RX<br>0: Mask<br>1: Pass. |
| 0 | R/W | 0x0 | R_PIO_GATING<br>Gating APB Clock for R_PIO<br>0: Mask<br>1: Pass. |

### 3.23.5.4. PLL Control Register 1 (Default Value: 0x00040015)

| Offset: 0x0044 | | | Register Name: **PLL_CTRL_REG1** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |

| 31:24 | R/W | 0x0 | KEY_FIELD.<br>Key Field for LDO Enable bit.<br>If the key field value is 0xA7, the bit[23:0] can be modified. |
|---|---|---|---|
| 23:3 | / | / | / |
| 2 | R/W | 1 | OSC24M_EN.<br>External Crystal OSC24M Enable |
| 1 | / | / | / |
| 0 | R/W | 1 | LDO_EN.<br>0: Disable<br>1: Enable.<br>Note: PLL Power enable (power source from VCC_PLL). |

### 3.23.5.5. R_CIR_RX Clock Register (Default Value: 0x00000000)

| Offset: 0x0054 | | | Register Name: **R_CIR_RX_CLK_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x0 | SCLK_GATING.<br>Gating Special Clock (Max Clock = 100MHz)<br>0: Clock is OFF<br>1: Clock is ON.<br>SCLK= Clock Source/Pre-Divider N/Divider M. |
| 30:26 | / | / | / |
| 25:24 | R/W | 0x0 | CLK_SRC_SEL.<br>Clock Source Select<br>00: Internal OSC / 16<br>01: OSC24M<br>1X: /.<br>The Internal OSC is about 16M. |
| 23:18 | / | / | / |
| 16:17 | R/W | 0x0 | CLK_DIV_RATIO_N.<br>Clock Pre Divide Ratio (n)<br>00: /1<br>01: /2<br>10: /4<br>11: /8. |
| 15:4 | / | / | / |
| 3:0 | R/W | 0x0 | CLK_DIV_RATIO_M.<br>Clock Divide Ratio (m)<br>The pre-divided clock is divided by (m+1). The divider is from 1 to 16. |

### 3.23.5.6. APB0 Software Reset Register(Default Value: 0x00000000)

| Offset: 0x00B0 | | | Register Name: **APB0_SOFT_RST_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:7 | / | / | / |
| 6 | R/W | 0x0 | R_TWI_RST. R_TWI Reset Control 0: Assert 1: De-assert. |
| 5 | / | / | / |
| 4 | R/W | 0x0 | R_UART_RST. R_UART Reset Control 0: Assert 1: De-assert. |
| 3 | R/W | 0x0 | R_RSB_RST. R_P2WI Reset Control 0: Assert 1: De-assert. |
| 2 | R/W | 0x0 | R_TIMER_RST. R_TIMER Reset Control 0: Assert 1: De-assert. |
| 1 | R/W | 0x0 | R_CIR_RX_RST. R_CIR_RX Reset Control 0: Assert 1: De-assert. |
| 0 | / | / | / |

### 3.23.5.7. Cluster0 CPUX Power Off Gating Register (Default Value: 0x00000000)

| Offset: 0x0100 | | | Register Name: **C0CPUX_PWROFF_GATING_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:5 | / | / | / |
| 4 | R/W | 0x0 | PWROFF_GATING Gating the corresponding modules when C0_CPU0 power off. 0: Invalid 1: Valid. Note: This bit should be set to 1 before C0_CPU0 power off while it should be set to 0 after the C0_CPU0 power on. |
| 3 | R/W | 0x0 | PWROFF_GATING Gating the corresponding modules when C0_CPU3 power off. 0: Invalid 1: Valid. |

| | | | Note: This bit should be set to 1 before C0_CPU3 power off while it should be set to 0 after the C0_CPU3 power on. |
|---|---|---|---|
| 2 | R/W | 0x0 | PWROFF_GATING<br>Gating the corresponding modules when C0_CPU2 power off.<br>0: Invalid<br>1: Valid.<br>Note: This bit should be set to 1 before C0_CPU2 power off while it should be set to 0 after the C0_CPU2 power on. |
| 1 | R/W | 0x0 | PWROFF_GATING<br>Gating the corresponding modules when C0_CPU1 power off.<br>0: Invalid<br>1: Valid.<br>Note: This bit should be set to 1 before C0_CPU1 power off while it should be set to 0 after the C0_CPU1 power on. |
| 0 | R/W | 0x0 | PWROFF_GATING<br>Gating the corresponding modules when Cluster 0 power off.<br>0: Invalid<br>1: Valid.<br>Note: This bit should be set to 1 before Cluster 0 power off while it should be set to 0 after the Cluster 0 power on. |

### 3.23.5.8. Cluster1 CPUX Power Off Gating Register (Default Value: 0x0000001F)

| Offset: 0x0104 | | | Register Name: **C1CPUX_PWROFF_GATING_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:5 | / | / | / |
| 4 | R/W | 0x0 | PWROFF_GATING<br>Gating the corresponding modules when C1_CPU0 power off.<br>0: Invalid<br>1: Valid.<br>Note: This bit should be set to 1 before C1_CPU0 power off while it should be set to 0 after the C1_CPU0 power on. |
| 3 | R/W | 0x0 | PWROFF_GATING<br>Gating the corresponding modules when C1_CPU3 power off.<br>0: Invalid<br>1: Valid.<br>Note: This bit should be set to 1 before C1_CPU3 power off while it should be set to 0 after the C1_CPU3 power on. |
| 2 | R/W | 0x0 | PWROFF_GATING<br>Gating the corresponding modules when C1_CPU2 power off.<br>0: Invalid<br>1: Valid.<br>Note: This bit should be set to 1 before C1_CPU2 power off while it should be set to 0 after the C1_CPU2 power on. |

| 1 | R/W | 0x0 | PWROFF_GATING<br>Gating the corresponding modules when C1_CPU1 power off.<br>0: Invalid<br>1: Valid.<br>Note: This bit should be set to 1 before C1_CPU1 power off while it should be set to 0 after the C1_CPU1 power on. |
| 0 | R/W | 0x0 | PWROFF_GATING<br>Gating the corresponding modules when Cluster 1 power off.<br>0: Invalid<br>1: Valid.<br>Note: This bit should be set to 1 before Cluster1 power off while it should be set to 0 after the Cluster1 power on. |

### 3.23.5.9. VDD_SYS Power Off Gating Register (Default Value: 0x00000000)

| Offset: 0x0110 | | | Register Name: **VDD_SYS_PWROFF_GATING_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:13 | / | / | / |
| 12 | R/W | 0x0 | VCC_GPIO_GATING<br>Gating the VCC_GPIO (GPIO BCDEFGH) when VDD_SYS power off.<br>0: Invalid<br>1: Valid.<br>Note: This bit should be set to 1 before VDD_SYS power off while it should be set to 0 after the VDD_SYS power on. |
| 11:9 | / | / | / |
| 8 | R/W | 0x0 | VCC_PLL_GATING<br>Gating the VCC_PLL when VDD_SYS power off.<br>0: Invalid<br>1: Valid.<br>Note: This bit should be set to 1 before VDD_SYS power off while it should be set to 0 after the VDD_SYS power on. |
| 7:3 | / | / | / |
| 4 | R/W | 0x0 | VCC_PLL_LOW_VOLTAGE_GATING<br>Gating the VCC_PLL Low Voltage when VDD_SYS power off.<br>0: Invalid<br>1: Valid.<br>Note: This bit should be set to 1 before VDD_SYS power off while it should be set to 0 after the VDD_SYS power on. |
| 3 | R/W | 0x0 | VDD_CPUS_GATING<br>Gating the corresponding modules to the CPUS Power Domain when VDD_SYS power off.<br>0: Invalid<br>1: Valid.<br>Note: This bit should be set to 1 before VDD_SYS power off while it should |

| | | | be set to 0 after the VDD_SYS power on. CPUS CLK Source should switch to Internal OSC before CPUs Power Off Gating. |
|---|---|---|---|
| 2 | / | / | / |
| 1 | R/W | 0x0 | DRAM_ZQ_PAD_HOLD.<br>0:Not Hold<br>1:Hold DRAM ZQ Pad.<br>Note: This bit should be set to 1 before VDD_SYS power off while it should be set to 0 after the VDD_SYS power on. |
| 0 | R/W | 0x0 | DRAM_CH_PAD_HOLD.<br>Hold the pad of DRAM channel<br>0:Not Hold<br>1:Hold DRAM Pad.<br>Note: This bit should be set to 1 before VDD_SYS power off while it should be set to 0 after the VDD_SYS power on. |

### 3.23.5.10.  GPU Power Off Gating Register(Default Value: 0x00000000)

| Offset: 0x0118 | | | Register Name: **GPU_PWROFF_GATING_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:1 | / | / | / |
| 0 | R/W | 0x0 | PWROFF_GATING<br>Gating the corresponding modules when GPU power off.<br>0: Invalid<br>1: Valid.<br>Note: This bit should be set to 1 before GPU power off while it should be set to 0 after the GPU power on. |

### 3.23.5.11.  VDD_SYS Power Domain Reset Register (Default Value: 0x00000001)

| Offset: 0x0120 | | | Register Name: **VDD_SYS_PWROFF_RST_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:1 | / | / | / |
| 0 | R/W | 0x1 | MODULE_RST<br>VDD_SYS Power Domain Modules should be reset before VDD_SYS power on.<br>0: Assert<br>1: De-assert. |

### 3.23.5.12.  C0_CPU0 Power Switch Control Register (Default Value: 0x00000000)

| Offset: 0x0140 | Register Name: **C0_CPU0_PWR_SWITCH_CTRL** |
|---|---|

| Bit | R/W | Default/Hex | Description |
|-----|-----|-------------|-------------|
| 31:8 | / | / | / |
| 7:0 | R/W | 0x0 | C0_CPU0_PWR_SWITCH_CTRL. <br> 0x00: Power On <br> ...... <br> 0xFF: Power Down. |

### 3.23.5.13.  C0_CPU1 Power Switch Control Register (Default Value: 0x00000000)

| Offset: 0x0144 | | | Register Name: **C0_CPU1_PWR_SWITCH_CTRL** |
|-----|-----|-------------|-------------|
| Bit | R/W | Default/Hex | Description |
| 31:8 | / | / | / |
| 7:0 | R/W | 0x0 | C0_CPU1_PWR_SWITCH_CTRL. <br> 0x00: Power On <br> ...... <br> 0xFF: Power Down. |

### 3.23.5.14.  C0_CPU2 Power Switch Control Register (Default Value: 0x00000000)

| Offset: 0x0148 | | | Register Name: **C0_CPU2_PWR_SWITCH_CTRL** |
|-----|-----|-------------|-------------|
| Bit | R/W | Default/Hex | Description |
| 31:8 | / | / | / |
| 7:0 | R/W | 0x0 | C0_CPU2_PWR_SWITCH_CTRL. <br> 0x00: Power On <br> ...... <br> 0xFF: Power Down. |

### 3.23.5.15.  C0_CPU3 Power Switch Control Register (Default Value: 0x00000000)

| Offset: 0x014C | | | Register Name: **C0_CPU3_PWR_SWITCH_CTRL** |
|-----|-----|-------------|-------------|
| Bit | R/W | Default/Hex | Description |
| 31:8 | / | / | / |
| 7:0 | R/W | 0x0 | C0_CPU3_PWR_SWITCH_CTRL. <br> 0x00: Power On <br> ...... <br> 0xFF: Power Down. |

### 3.23.5.16.  C1_CPU0 Power Switch Control Register (Default Value: 0x00000000)

| Offset: 0x0150 | | | Register Name: **C1_CPU0_PWR_SWITCH_CTRL** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:8 | / | / | / |
| 7:0 | R/W | 0x0 | C1_CPU0_PWR_SWITCH_CTRL. <br> 0x00: Power On <br> ...... <br> 0xFF: Power Down. |

### 3.23.5.17.  C1_CPU1 Power Switch Control Register (Default Value: 0x00000000)

| Offset: 0x0154 | | | Register Name: **C1_CPU1_PWR_SWITCH_CTRL** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:8 | / | / | / |
| 7:0 | R/W | 0x0 | C1_CPU1_PWR_SWITCH_CTRL. <br> 0x00: Power On <br> ...... <br> 0xFF: Power Down. |

### 3.23.5.18.  C1_CPU2 Power Switch Control Register (Default Value: 0x00000000)

| Offset: 0x0158 | | | Register Name: **C1_CPU2_PWR_SWITCH_CTRL** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:8 | / | / | / |
| 7:0 | R/W | 0x0 | C1_CPU2_PWR_SWITCH_CTRL. <br> 0x00: Power On <br> ...... <br> 0xFF: Power Down. |

### 3.23.5.19.  C1_CPU3 Power Switch Control Register (Default Value: 0x00000000)

| Offset: 0x015C | | | Register Name: **C1_CPU3_PWR_SWITCH_CTRL** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:8 | / | / | / |
| 7:0 | R/W | 0x0 | C1_CPU0_PWR_SWITCH_CTRL. <br> 0x00: Power On <br> ...... <br> 0xFF: Power Down. |

### 3.23.5.20. R_PIO Hold Control Register (Default Value: 0x00000000)

| Offset: 0x01F0 | | | Register Name: **RPIO_HOLD_CTRL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x0 | WRITE_PULSE<br>Set 1 then immediately set 0 to write the value of DATA_WRITE into the corresponding PIO Register that its address is allocated by PIO_REG_ADDR. It's no need to consider the delay time between the set 1 and set 0 operation on software. |
| 30:18 | / | / | / |
| 17:16 | R/W | 0x0 | PIO_REG_ADDR<br>The Corresponding PIO Register Address.<br>Address0 bit0: R_PIO Pad Hold control(the Power Off Hold Pad : PL) |
| 15:8 | R/W | 0x0 | DATA_WRITE.<br>The data that should be written to the corresponding PIO Register. |
| 7:0 | R | 0x0 | DATA_READ.<br>The value of the corresponding PIO Register. |

**Note:** This register is in the RTC domain.

### 3.23.5.21. OSC24M Control Register (Default Value: 0x00000000)

| Offset: 0x01F4 | | | Register Name: **OSC24M_CTRL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:2 | / | / | / |
| 1 | R/W | 0x0 | OSC24M_SRC_SELECT.<br>0: 24MHz, it is from external oscillator.<br>1: 16MHz, it is from RC oscillator in the SOC.<br>Before selecting 16MHz OSC, the 16MHz OSC must be enabled. |
| 0 | RO | 0x0 | OSC16M_ENABLE.<br>0: Disable.<br>1: Enable. |

## 3.23.6. Programming Guidelines

**(1)** Release the device's clock gating after software reset has been de-assert.

**(2)** To switch device clock source need set reasonable and effective division factor.

**(3)** After wrote VDD_SYS Power gating Register, gating can be valid after 1us delay; After wrote VDD_SYS reset Register, reset can be valid after 1us delay.

**(4)** Two CPU can not operate simultaneously R_PIO Hold Control Register.

## 3.24. Port Controller

The chip has 8 ports for multi-functional input/output pins. They are shown below:

- Port B(PB): 11 input/output port
- Port C(PC): 19 input/output port
- Port D(PD): 30 input/output port
- Port E(PE): 20 input/output port
- Port F(PF): 7 input/output port
- Port G(PG): 14 input/output port
- Port H(PH): 12 input/output port
- Port L(PL): 13 input/output port

For various system configurations, these ports can be easily configured by software. All these ports can be configured as GPIO if multiplexed functions are not used. The total 4 group external PIO interrupt sources are supported and interrupt mode can be configured by software.

### 3.24.1. Port Controller Register List

| Module Name | Base Address |
| --- | --- |
| PIO | 0x01C20800 |

| Register Name | Offset | Description |
| --- | --- | --- |
| Pn_CFG0 | n*0x24+0x00 | Port n Configure Register 0 (n from 1 to 7) |
| Pn_CFG1 | n*0x24+0x04 | Port n Configure Register 1 (n from 1 to 7) |
| Pn_CFG2 | n*0x24+0x08 | Port n Configure Register 2 (n from 1 to 7) |
| Pn_CFG3 | n*0x24+0x0C | Port n Configure Register 3 (n from 1 to 7) |
| Pn_DAT | n*0x24+0x10 | Port n Data Register (n from 1 to 7) |
| Pn_DRV0 | n*0x24+0x14 | Port n Multi-Driving Register 0 (n from 1 to 7) |
| Pn_DRV1 | n*0x24+0x18 | Port n Multi-Driving Register 1 (n from 1 to 7) |
| Pn_PUL0 | n*0x24+0x1C | Port n Pull Register 0 (n from 1 to 7) |
| Pn_PUL1 | n*0x24+0x20 | Port n Pull Register 1 (n from 1 to 7) |
| PB_INT_CFG0 | 0x200+0*0x20+0x00 | PIO Interrrupt Configure Register 0 |
| PB _INT_CFG1 | 0x200+0*0x20+0x04 | PIO Interrrupt Configure Register 1 |
| PB _INT_CFG2 | 0x200+0*0x20+0x08 | PIO Interrrupt Configure Register 2 |
| PB _INT_CFG3 | 0x200+0*0x20+0x0C | PIO Interrrupt Configure Register 3 |
| PB _INT_CTL | 0x200+0*0x20+0x10 | PIO Interrupt Control Register |

| PB _INT_STA | 0x200+0*0x20+0x14 | PIO Interrupt Status Register |
|---|---|---|
| PB _INT_DEB | 0x200+0*0x20+0x18 | PIO Interrupt Debounce Register |
| PG_INT_CFG0 | 0x200+1*0x20+0x00 | PIO Interrrupt Configure Register 0 |
| PG _INT_CFG1 | 0x200+1*0x20+0x04 | PIO Interrrupt Configure Register 1 |
| PG _INT_CFG2 | 0x200+1*0x20+0x08 | PIO Interrrupt Configure Register 2 |
| PG _INT_CFG3 | 0x200+1*0x20+0x0C | PIO Interrrupt Configure Register 3 |
| PG _INT_CTL | 0x200+1*0x20+0x10 | PIO Interrupt Control Register |
| PG _INT_STA | 0x200+1*0x20+0x14 | PIO Interrupt Status Register |
| PG _INT_DEB | 0x200+1*0x20+0x18 | PIO Interrupt Debounce Register |
| PH_INT_CFG0 | 0x200+2*0x20+0x00 | PIO Interrrupt Configure Register 0 |
| PH _INT_CFG1 | 0x200+2*0x20+0x04 | PIO Interrrupt Configure Register 1 |
| PH _INT_CFG2 | 0x200+2*0x20+0x08 | PIO Interrrupt Configure Register 2 |
| PH _INT_CFG3 | 0x200+2*0x20+0x0C | PIO Interrrupt Configure Register 3 |
| PH _INT_CTL | 0x200+2*0x20+0x10 | PIO Interrupt Control Register |
| PH _INT_STA | 0x200+2*0x20+0x14 | PIO Interrupt Status Register |
| PH _INT_DEB | 0x200+2*0x20+0x18 | PIO Interrupt Debounce Register |

| Module Name | Base Address |
|---|---|
| PL_PIO | 0x01F02C00 |

| Register Name | Offset | Description |
|---|---|---|
| PL_CFG0 | 0*0x24+0x00 | Port L Configure Register 0 |
| PL_CFG1 | 0*0x24+0x04 | Port L Configure Register 1 |
| PL_CFG2 | 0*0x24+0x08 | Port L Configure Register 2 |
| PL_CFG3 | 0*0x24+0x0C | Port L Configure Register 3 |
| PL_DAT | 0*0x24+0x10 | Port L Data Register |
| PL_DRV0 | 0*0x24+0x14 | Port L Multi-Driving Register 0 |
| PL_DRV1 | 0*0x24+0x18 | Port L Multi-Driving Register 1 |
| PL_PUL0 | 0*0x24+0x1C | Port L Pull Register 0 |
| PL_PUL1 | 0*0x24+0x20 | Port L Pull Register 1 |
| PL_INT_CFG0 | 0x200+0*0x20+0x00 | PIO Interrrupt Configure Register 0 |
| PL_INT_CFG1 | 0x200+0*0x20+0x04 | PIO Interrrupt Configure Register 1 |
| PL_INT_CFG2 | 0x200+0*0x20+0x08 | PIO Interrrupt Configure Register 2 |
| PL_INT_CFG3 | 0x200+0*0x20+0x0C | PIO Interrrupt Configure Register 3 |
| PL_INT_CTL | 0x200+0*0x20+0x10 | PIO Interrupt Control Register |
| PL_INT_STA | 0x200+0*0x20+0x14 | PIO Interrupt Status Register |
| PL_INT_DEB | 0x200+0*0x20+0x18 | PIO Interrupt Debounce Register |

## 3.24.2. Port Controller Register Description

### 3.24.2.1. PB Configure Register 0 (Default Value: 0x77777777)

| Offset: 0x24 | | | Register Name: **PB_CFG0_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | / | / | / |
| 30:28 | R/W | 0x7 | PB7_SELECT<br>000:Input      001:Output<br>010:I2S0_DIN    011:TDM_DIN<br>100:Reserved   101:Reserved<br>110:PB_EINT7   111:IO Disable |
| 27 | / | / | / |
| 26:24 | R/W | 0x7 | PB6_SELECT<br>000:Input      001:Output<br>010:I2S0_DOUT   011:TDM_DOUT<br>100:Reserved   101:Reserved<br>110:PB_EINT6   111:IO Disable |
| 23 | / | / | / |
| 22:20 | R/W | 0x7 | PB5_SELECT<br>000:Input      001:Output<br>010:I2S0_BCLK   011:TDM_BCLK<br>100:Reserved   101:Reserved<br>110:PB_EINT5   111:IO Disable |
| 19 | / | / | / |
| 18:16 | R/W | 0x7 | PB4_SELECT<br>000:Input      001:Output<br>010:I2S0_LRCK   011:TDM_LRCK<br>100:Reserved   101:Reserved<br>110:PB_EINT4   111:IO Disable |
| 15 | / | / | / |
| 14:12 | R/W | 0x7 | PB3_SELECT<br>000:Input      001:Output<br>010:UART2_CTS  011:JTAG_DI0<br>100:Reserved   101:Reserved<br>110:PB_EINT3   111:IO Disable |
| 11 | / | / | / |
| 10:8 | R/W | 0x7 | PB2_SELECT<br>000:Input      001:Output<br>010:UART2_RTS  011:JTAG_DO0<br>100:Reserved   101:Reserved<br>110:PB_EINT2    111:IO Disable |

| 7 | / | / | / |
|---|---|---|---|
| 6:4 | R/W | 0x7 | PB1_SELECT<br>000:Input　　　　　001:Output<br>010:UART2_RX　　　011:JTAG_CK0<br>100:Reserved　　　　101:Reserved<br>110:PB_EINT1　　　　111:IO Disable |
| 3 | / | / | / |
| 2:0 | R/W | 0x7 | PB0_SELECT<br>000:Input　　　　　001:Output<br>010:UART2_TX　　　011:JTAG_MS0<br>100:Reserved　　　　101:Reserved<br>110:PB_EINT0　　　　111:IO Disable |

### 3.24.2.2. PB Configure Register 1 (Default Value: 0x00000777)

| Offset: 0x28 | | | Register Name: **PB_CFG1_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:11 | / | / | / |
| 10:8 | R/W | 0x7 | PB10_SELECT<br>000:Input　　　　　001:Output<br>010:UART0_RX　　　011:Reserved<br>100:Reserved　　　　101:Reserved<br>110:PB_EINT10　　　111:IO Disable |
| 7 | / | / | |
| 6:4 | R/W | 0x7 | PB9_SELECT<br>000:Input　　　　　001:Output<br>010:UART0_TX　　　011:Reserved<br>100:Reserved　　　　101:Reserved<br>110:PB_EINT9　　　　111:IO Disable |
| 3 | / | / | |
| 2:0 | R/W | 0x7 | PB8_SELECT<br>000:Input　　　　　001:Output<br>010:I2S0_MCLK　　　011:TDM_MCLK<br>100:Reserved　　　　101:Reserved<br>110:PB_EINT8　　　　111:IO Disable |

### 3.24.2.3. PB Configure Register 2 (Default Value: 0x00000000)

| Offset: 0x2C | | | Register Name: **PB_CFG2_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | / | / | / |

### 3.24.2.4. PB Configure Register 3 (Default Value: 0x00000000)

| Offset: 0x30 | | | Register Name: **PB_CFG3_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | / | / | / |

### 3.24.2.5. PB Data Register (Default Value: 0x00000000)

| Offset: 0x34 | | | Register Name: **PB_DATA_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:11 | / | / | / |
| 10:0 | R/W | 0x0 | PB_DAT<br>If the port is configured as input, the corresponding bit is the pin state. If the port is configured as output, the pin state is the same as the corresponding bit. The read bit value is the value setup by software. If the port is configured as functional pin, the undefined value will be read. |

### 3.24.2.6. PB Multi-Driving Register 0 (Default Value: 0x00155555)

| Offset: 0x38 | | | Register Name: **PB_DRV0_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:22 | / | / | Reserved |
| [2i+1:2i]<br>(i=0~10) | R/W | 0x1 | PB_DRV<br>PB[n] Multi-Driving Select (n = 0~10)<br>00: Level 0　　　　　　　01: Level 1<br>10: Level 2　　　　　　　11: Level 3 |

### 3.24.2.7. PB Multi-Driving Register 1 (Default Value: 0x00000000)

| Offset: 0x3C | | | Register Name: **PB_DRV1_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | / | / | / |

### 3.24.2.8. PB PULL Register 0 (Default Value: 0x00000000)

| Offset: 0x40 | | | Register Name: **PB_PULL0_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:22 | / | / | Reserved |
| [2i+1:2i] | R/W | 0x0 | PB_PULL |

| (i=0~10) | | | PB[n] Pull-up/down Select (n = 0~10) |
|---|---|---|---|
| | | | 00: Pull-up/down disable 01: Pull-up |
| | | | 10: Pull-down 11: Reserved |

### 3.24.2.9. PB PULL Register 1 (Default Value: 0x00000000)

| Offset: 0x44 | | | Register Name: **PB_PULL1 _REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | / | / | / |

### 3.24.2.10. PC Configure Register 0 (Default Value: 0x77777777)

| Offset: 0x48 | | | Register Name: **PC_CFG0_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | / | / | / |
| 30:28 | R/W | 0x7 | PC7_SELECT<br>000:Input 001:Output<br>010:NAND_RB1 011:Reserved<br>100:Reserved 101:Reserved<br>110:Reserved 111:IO Disable |
| 27 | / | / | / |
| 26:24 | R/W | 0x7 | PC6_SELECT<br>000:Input 001:Output<br>010:NAND_RB0 011:SDC2_CMD<br>100:Reserved 101:Reserved<br>110:Reserved 111:IO Disable |
| 23 | / | / | / |
| 22:20 | R/W | 0x7 | PC5_SELECT<br>000:Input 001:Output<br>010:NAND_RE 011:SDC2_CLK<br>100:Reserved 101:Reserved<br>110:Reserved 111:IO Disable |
| 19 | / | / | / |
| 18:16 | R/W | 0x7 | PC4_SELECT<br>000:Input 001:Output<br>010:NAND_CE0 011:Reserved<br>100:Reserved 101:Reserved<br>110:Reserved 111:IO Disable |
| 15 | / | / | / |
| 14:12 | R/W | 0x7 | PC3_SELECT<br>000:Input 001:Output<br>010:NAND_CE1 011:SPI0_CS |

| | | | |
|---|---|---|---|
| | | | 100:Reserved 101:Reserved |
| | | | 110:Reserved 111:IO Disable |
| 11 | / | / | / |
| 10:8 | R/W | 0x7 | PC2_SELECT<br>000:Input 001:Output<br>010:NAND_CLE 011:SPI0_CLK<br>100:Reserved 101:Reserved<br>110:Reserved 111:IO Disable |
| 7 | / | / | / |
| 6:4 | R/W | 0x7 | PC1_SELECT<br>000:Input 001:Output<br>010:NAND_ALE 011:SPI0_MISO<br>100:Reserved 101:Reserved<br>110:Reserved 111:IO Disable |
| 3 | / | / | / |
| 2:0 | R/W | 0x7 | PC0_SELECT<br>000:Input 001:Output<br>010:NAND_WE 011:SPI0_MOSI<br>100:Reserved 101:Reserved<br>110:Reserved 111:IO Disable |

### 3.24.2.11. PC Configure Register 1 (Default Value: 0x77777777)

| Offset: 0x4C | | | Register Name: **PC_CFG1_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | / | / | / |
| 30:28 | R/W | 0x7 | PC15_SELECT<br>000:Input 001:Output<br>010:NAND_DQ7 011:SDC2_D7<br>100:Reserved 101:Reserved<br>110:Reserved 111:IO Disable |
| 27 | / | / | / |
| 26:24 | R/W | 0x7 | PC14_SELECT<br>000:Input 001:Output<br>010:NAND_DQ6 011:SDC2_D6<br>100:Reserved 101:Reserved<br>110:Reserved 111:IO Disable |
| 23 | / | / | / |
| 22:20 | R/W | 0x7 | PC13_SELECT<br>000:Input 001:Output<br>010:NAND_DQ5 011:SDC2_D5<br>100:Reserved 101:Reserved<br>110:Reserved 111:IO Disable |
| 19 | / | / | / |

| 18:16 | R/W | 0x7 | PC12_SELECT<br>000:Input     001:Output<br>010:NAND_DQ4  011:SDC2_D4<br>100:Reserved   101:Reserved<br>110:Reserved   111:IO Disable |
|---|---|---|---|
| 15 | / | / | / |
| 14:12 | R/W | 0x7 | PC11_SELECT<br>000:Input     001:Output<br>010:NAND_DQ3  011:SDC2_D3<br>100:Reserved   101:Reserved<br>110:Reserved   111:IO Disable |
| 11 | / | / | / |
| 10:8 | R/W | 0x7 | PC10_SELECT<br>000:Input     001:Output<br>010:NAND_DQ2  011:SDC2_D2<br>100:Reserved   101:Reserved<br>110:Reserved   111:IO Disable |
| 7 | / | / | / |
| 6:4 | R/W | 0x7 | PC9_SELECT<br>000:Input     001:Output<br>010:NAND_DQ1  011:SDC2_D1<br>100:Reserved   101:Reserved<br>110:Reserved   111:IO Disable |
| 3 | / | / | / |
| 2:0 | R/W | 0x7 | PC8_SELECT<br>000:Input     001:Output<br>010:NAND_DQ0  011:SDC2_D0<br>100:Reserved   101:Reserved<br>110:Reserved   111:IO Disable |

### 3.24.2.12. PC Configure Register 2 (Default Value: 0x00000777)

| Offset: 0x50 | | | Register Name: **PC_CFG2 _REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:11 | / | / | / |
| 10:8 | R/W | 0x7 | PC18_SELECT<br>000:Input     001:Output<br>010:NAND_CE3   011:Reserved<br>100:Reserved   101:Reserved<br>110:Reserved   111:IO Disable |
| 7 | / | / | / |
| 6:4 | R/W | 0x7 | PC17_SELECT<br>000:Input     001:Output<br>010:NAND_CE2   011:Reserved |

| | | | 100:Reserved | 101:Reserved |
|---|---|---|---|---|
| | | | 110:Reserved | 111:IO Disable |
| 3 | / | / | / | |
| | | | PC16_SELECT | |
| | | | 000:Input | 001:Output |
| | | | 010:NAND_DQS | 011:SDC2_RST |
| | | | 100:Reserved | 101:Reserved |
| 2:0 | R/W | 0x7 | 110:Reserved | 111:IO Disable |

### 3.24.2.13.  PC Configure Register 3 (Default Value: 0x00000000)

| Offset: 0x54 | | | Register Name: **PC_CFG3 _REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | / | / | / |

### 3.24.2.14.  PC Data Register (Default Value: 0x00000000)

| Offset: 0x58 | | | Register Name: **PC_DATA _REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:19 | / | / | / |
| 18:0 | R/W | 0x0 | PC_DAT<br>If the port is configured as input, the corresponding bit is the pin state. If the port is configured as output, the pin state is the same as the corresponding bit. The read bit value is the value setup by software. If the port is configured as functional pin, the undefined value will be read. |

### 3.24.2.15.  PC Multi-Driving Register 0 (Default Value: 0x55555555)

| Offset: 0x5C | | | Register Name: **PC_ DRV0 _REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| [2i+1:2i]<br>(i=0~15) | R/W | 0x1 | PC_DRV<br>PC[n] Multi-Driving SELECT (n = 0~15)<br>00: Level 0        01: Level 1<br>10: Level 2        11: Level 3 |

### 3.24.2.16.  PC Multi-Driving Register 1 (Default Value: 0x00000015)

| Offset: 0x60 | | | Register Name: **PC_ DRV1 _REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |

| 31:6 | / | / | / |
|------|---|---|---|
| [2i+1:2i]<br>(i=0~2) | R/W | 0x1 | PC_DRV<br>PC[n] Multi-Driving Select (n = 16~18)<br>00: Level 0                01: Level 1<br>10: Level 2                11: Level 3 |

### 3.24.2.17. PC PULL Register 0 (Default Value: 0x00005140)

| Offset: 0x64 | | | Register Name: **PC_ PULL0 _REG** |
|------|---|---|---|
| Bit | R/W | Default/Hex | Description |
| [2i+1:2i]<br>(i=0~15) | R/W | 0x5140 | PC_PULL<br>PC[n] Pull-up/down Select (n = 0~15)<br>00: Pull-up/down disable     01: Pull-up<br>10: Pull-down             11: Reserved |

### 3.24.2.18. PC PULL Register 1 (Default Value: 0x00000014)

| Offset: 0x68 | | | Register Name: **PC_ PULL1 _REG** |
|------|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:6 | / | / | Reserved |
| [2i+1:2i]<br>(i=0~2) | R/W | 0x14 | PC_PULL<br>PC[n] Pull-up/down Select (n = 16~18)<br>00: Pull-up/down disable     01: Pull-up<br>10: Pull-down             11: Reserved |

### 3.24.2.19. PD Configure Register 0 (Default Value: 0x77777777)

| Offset: 0x6C | | | Register Name: **PD_CFG0 _REG** |
|------|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | / | / | / |
| 30:28 | R/W | 0x7 | PD7_SELECT<br>000:Input                001:Output<br>010:LCD_D7           011:Reserved<br>100: RGMII-RXCTL/MII-RXDV 101:Reserved<br>110:Reserved         111:IO Disable |
| 27 | / | / | Reserved |
| 26:24 | R/W | 0x7 | PD6_SELECT<br>000:Input                001:Output<br>010:LCD_D6            011:Reserved<br>100: RGMII-RXCK/MII-RXCK 101:Reserved<br>110:Reserved         111:IO Disable |

| 23 | / | / | / |
|---|---|---|---|
| 22:20 | R/W | 0x7 | PD5_SELECT<br>000:Input                    001:Output<br>010:LCD_D5                 011:Reserved<br>100: RGMII-RXD0/MII-RXD0 101:Reserved<br>110:Reserved                 111:IO Disable |
| 19 | / | / | / |
| 18:16 | R/W | 0x7 | PD4_SELECT<br>000:Input                    001:Output<br>010:LCD_D4                 011:Reserved<br>100: RGMII-RXD1/MII-RXD1 101:Reserved<br>110:Reserved                 111:IO Disable |
| 15 | / | / | / |
| 14:12 | R/W | 0x7 | PD3_SELECT<br>000:Input                    001:Output<br>010:LCD_D3                 011:Reserved<br>100: RGMII-RXD2/MII-RXD2 101:Reserved<br>110:Reserved                 111:IO Disable |
| 11 | / | / | / |
| 10:8 | R/W | 0x7 | PD2_SELECT<br>000:Input                    001:Output<br>010:LCD_D2                 011:Reserved<br>100: RGMII-RXD3/MII-RXD3 101:Reserved<br>110:Reserved                 111:IO Disable |
| 7:0 | R/W | 0x77 | / |

### 3.24.2.20.    PD Configure Register 1 (Default Value: 0x77777777)

| Offset: 0x70 | | | Register Name: **PD_CFG1 _REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | / | / | / |
| 30:28 | R/W | 0x7 | PD15_SELECT<br>000:Input                    001:Output<br>010:LCD_D15               011:Reserved<br>100:RGMII-NULL/MII-CRS        101:Reserved<br>110:Reserved            111:IO Disable |
| 27 | / | / | / |
| 26:24 | R/W | 0x7 | PD14_SELECT<br>000:Input                    001:Output<br>010:LCD_D14               011:Reserved<br>100:RGMII-TXD0/MII-TXD0        101:Reserved<br>110:Reserved            111:IO Disable |
| 23 | / | / | / |
| 22:20 | R/W | 0x7 | PD13_SELECT |

| | | | |
|---|---|---|---|
| | | | 000:Input                 001:Output |
| | | | 010:LCD_D13               011:Reserved |
| | | | 100:RGMII-TXD1/MII-TXD1       101:Reserved |
| | | | 110:Reserved              111:IO Disable |
| 19 | / | / | / |
| | | | PD12_SELECT |
| | | | 000:Input                 001:Output |
| | | | 010:LCD_D12               011:Reserved |
| | | | 100:RGMII-TXD2/MII-TXD2       101:Reserved |
| 18:16 | R/W | 0x7 | 110:Reserved              111:IO Disable |
| 15 | / | / | / |
| | | | PD11_SELECT |
| | | | 000:Input                 001:Output |
| | | | 010:LCD_D11               011:Reserved |
| | | | 100:RGMII-TXD3/MII-TXD3       101:Reserved |
| 14:12 | R/W | 0x7 | 110:Reserved              111:IO Disable |
| 11 | / | / | / |
| | | | PD10_SELECT |
| | | | 000:Input                 001:Output |
| | | | 010:LCD_D10               011:Reserved |
| | | | 100:RGMII-NULL/MII-RXERR       101:Reserved |
| 10:8 | R/W | 0x7 | 110:Reserved              111:IO Disable |
| 7:0 | R/W | 0x77 | / |

### 3.24.2.21.   PD Configure Register 2 (Default Value: 0x77777777)

| Offset: 0x74 | | | Register Name: **PD_CFG2 _REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | / | / | / |
| | | | PD23_SELECT |
| | | | 000:Input                 001:Output |
| | | | 010:LCD_D23               011:LVDS_VN2 |
| | | | 100:GMDIO                 101:Reserved |
| 30:28 | R/W | 0x7 | 110:Reserved              111:IO Disable |
| 27 | / | / | / |
| | | | PD22_SELECT |
| | | | 000:Input                 001:Output |
| | | | 010:LCD_D22               011:LVDS_VP2 |
| | | | 100:GMDC                 101:Reserved |
| 26:24 | R/W | 0x7 | 110:Reserved              111:IO Disable |
| 23 | / | / | / |
| | | | PD21_SELECT |
| | | | 000:Input                 001:Output |
| 22:20 | R/W | 0x7 | 010:LCD_D21               011:LVDS_VN1 |

| | | | |
|---|---|---|---|
| | | | 100:GCLKIN/ECOL          101:Reserved<br>110:Reserved          111:IO Disable |
| 19 | / | / | / |
| 18:16 | R/W | 0x7 | PD20_SELECT<br>000:Input          001:Output<br>010:LCD_D20          011:LVDS_VP1<br>100:GNULL/ETXERR          101:Reserved<br>110:Reserved          111:IO Disable |
| 15 | / | / | / |
| 14:12 | R/W | 0x7 | PD19_SELECT<br>000:Input          001:Output<br>010:LCD_D19          011:LVDS_VN0<br>100:GTXCTL/ETXEN          101:Reserved<br>110:Reserved          111:IO Disable |
| 11 | / | / | / |
| 10:8 | R/W | 0x7 | PD18_SELECT<br>000:Input          001:Output<br>010:LCD_D18          011:LVDS_VP0<br>100:GTXCK/ETXCK          101:Reserved<br>110:Reserved          111:IO Disable |
| 7:0 | R/W | 0x77 | / |

### 3.24.2.22.    PD Configure Register 3 (Default Value: 0x00777777)

| Offset: 0x78 | | | Register Name: **PD_CFG3 _REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:24 | / | / | / |
| 23 | / | / | / |
| 22:20 | R/W | 0x7 | PD29_SELECT<br>000:Input          001:Output<br>010:Reserved          011:Reserved<br>100:Reserved          101:Reserved<br>110:Reserved          111:IO Disable |
| 19 | / | / | / |
| 18:16 | R/W | 0x7 | PD28_SELECT<br>000:Input          001:Output<br>010:PWM          011:Reserved<br>100:Reserved          101:Reserved<br>110:Reserved          111:IO Disable |
| 15 | / | / | / |
| 14:12 | R/W | 0x7 | PD27_SELECT<br>000:Input          001:Output<br>010:LCD_VSYNC          011:LVDS_VN3<br>100:Reserved          101:Reserved |

| | | | |
|---|---|---|---|
| | | | 110:Reserved          111:IO Disable |
| 11 | / | / | Reserved |
| 10:8 | R/W | 0x7 | PD26_SELECT<br>000:Input                  001:Output<br>010:LCD_HSYNC          011:LVDS_VP3<br>100:Reserved            101:Reserved<br>110:Reserved            111:IO Disable |
| 7 | / | / | / |
| 6:4 | R/W | 0x7 | PD25_SELECT<br>000:Input                  001:Output<br>010:LCD_DE              011:LVDS_VNC<br>100:Reserved            101:Reserved<br>110:Reserved            111:IO Disable |
| 3 | / | / | / |
| 2:0 | R/W | 0x7 | PD24_SELECT<br>000:Input                  001:Output<br>010:LCD_CLK             011:LVDS_VPC<br>100:Reserved            101:Reserved<br>110:Reserved            111:IO Disable |

### 3.24.2.23.  PD Data Register (Default Value: 0x00000000)

| Offset: 0x7C | | | Register Name: **PD_DATA _REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:30 | / | / | / |
| 29:0 | R/W | 0x0 | PD_DAT<br>If the port is configured as input, the corresponding bit is the pin state. If the port is configured as output, the pin state is the same as the corresponding bit. The read bit value is the value setup by software. If the port is configured as functional pin, the undefined value will be read. |

### 3.24.2.24.  PD Multi-Driving Register 0 (Default Value: 0x55555555)

| Offset: 0x80 | | | Register Name: **PD_ DRV0 _REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| [2i+1:2i]<br>(i=0~15) | R/W | 0x1 | PD_DRV<br>PD[n] Multi-Driving SELECT (n = 0~15)<br>00: Level 0                01: Level 1<br>10: Level 2                11: Level 3 |

### 3.24.2.25.  PD Multi-Driving Register 1 (Default Value: 0x05555555)

| Offset: 0x84 | | | Register Name: **PD_ DRV1 _REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:28 | / | / | / |
| [2i+1:2i]<br>(i=0~13) | R/W | 0x1 | PD_DRV<br>PD[n] Multi-Driving Select (n = 16~29)<br>00: Level 0                01: Level 1<br>10: Level 2                11: Level 3 |

### 3.24.2.26.  PD PULL Register 0 (Default Value: 0x00000000)

| Offset: 0x88 | | | Register Name: **PD_ PULL0 _REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| [2i+1:2i]<br>(i=0~15) | R/W | 0x0 | PD_PULL<br>PD[n] Pull-up/down Select (n = 0~15)<br>00: Pull-up/down disable     01: Pull-up<br>10: Pull-down              11: Reserved |

### 3.24.2.27.  PD PULL Register 1 (Default Value: 0x00000000)

| Offset: 0x8C | | | Register Name: **PD_ PULL1 _REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:28 | / | / | Reserved |
| [2i+1:2i]<br>(i=0~13) | R/W | 0x0 | PD_PULL<br>PD[n] Pull-up/down Select (n = 16~29)<br>00: Pull-up/down disable     01: Pull-up<br>10: Pull-down              11: Reserved |

### 3.24.2.28.  PE Configure Register 0 (Default Value: 0x77777777)

| Offset: 0x90 | | | Register Name: **PE_CFG0 _REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | / | / | / |
| 30:28 | R/W | 0x7 | PE7_SELECT<br>000:Input                 001:Output<br>010:CSI_D3                011:Reserved<br>100: CCIR-D1              101:Reserved<br>110:Reserved              111:IO Disable |
| 27 | / | / | / |

| | | | |
|---|---|---|---|
| 26:24 | R/W | 0x7 | PE6_SELECT<br>000:Input          001:Output<br>010:CSI_D2       011:Reserved<br>100: CCIR-D0     101:Reserved<br>110:Reserved     111:IO Disable |
| 23 | / | / | / |
| 22:20 | R/W | 0x7 | PE5_SELECT<br>000:Input          001:Output<br>010: CSI_D1      011:Reserved<br>100: Reserved    101:Reserved<br>110:Reserved     111:IO Disable |
| 19 | / | / | / |
| 18:16 | R/W | 0x7 | PE4_SELECT<br>000:Input          001:Output<br>010: CSI_D0      011:Reserved<br>100: Reserved    101:Reserved<br>110:Reserved     111:IO Disable |
| 15 | / | / | / |
| 14:12 | R/W | 0x7 | PE3_SELECT<br>000:Input          001:Output<br>010:CSI_VSYNC  011:Reserved<br>100: CCIR-VSYNC 101:Reserved<br>110:Reserved     111:IO Disable |
| 11 | / | / | / |
| 10:8 | R/W | 0x7 | PE2_SELECT<br>000:Input          001:Output<br>010:CSI_HSYNC  011:Reserved<br>100:CCIR-HSYNC 101:Reserved<br>110:Reserved     111:IO Disable |
| 7 | / | / | / |
| 6:4 | R/W | 0x7 | PE1_SELECT<br>000:Input          001:Output<br>010:CSI_MCLK   011:Reserved<br>100:CCIR-DE     101:Reserved<br>110:Reserved     111:IO Disable |
| 3 | / | / | / |
| 2:0 | R/W | 0x7 | PE0_SELECT<br>000:Input          001:Output<br>010:CSI_PCLK   011:Reserved<br>100:CCIR-CLK   101:Reserved<br>110:Reserved     111:IO Disable |

### 3.24.2.29.  PE Configure Register 1 (Default Value: 0x77777777)

| Offset: 0x94 | | | Register Name: **PE_CFG1 _REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | / | / | / |
| 30:28 | R/W | 0x7 | PE15_SELECT<br>000:Input　　　　　001:Output<br>010: CSI_SDA　　　011: TWI2_SDA<br>100:Reserved　　　101:Reserved<br>110:Reserved　　　111:IO Disable |
| 27 | / | / | / |
| 26:24 | R/W | 0x7 | PE14_SELECT<br>000:Input　　　　　001:Output<br>010: CSI_SCK　　　011: TWI2_SCK<br>100:Reserved　　　101:Reserved<br>110:Reserved　　　111:IO Disable |
| 23 | / | / | / |
| 22:20 | R/W | 0x7 | PE13_SELECT<br>000:Input　　　　　001:Output<br>010: CSI_D9　　　011: UART4_CTS<br>100: CCIR-D7　　　101:Reserved<br>110:Reserved　　　111:IO Disable |
| 19 | / | / | / |
| 18:16 | R/W | 0x7 | PE12_SELECT<br>000:Input　　　　　001:Output<br>010: CSI_D8　　　011: UART4_RTS<br>100: CCIR-D6　　　101:Reserved<br>110:Reserved　　　111:IO Disable |
| 15 | / | / | / |
| 14:12 | R/W | 0x7 | PE11_SELECT<br>000:Input　　　　　001:Output<br>010:CSI_D7　　　011: UART4_RX<br>100: CCIR-D5　　　101:Reserved<br>110:Reserved　　　111:IO Disable |
| 11 | / | / | / |
| 10:8 | R/W | 0x7 | PE10_SELECT<br>000:Input　　　　　001:Output<br>010:CSI_D6　　　011: UART4_TX<br>100: CCIR-D4　　　101:Reserved<br>110:Reserved　　　111:IO Disable |
| 7 | / | / | / |
| 6:4 | R/W | 0x7 | PE9_SELECT<br>000:Input　　　　　001:Output<br>010:CSI_D5　　　011: Reserved |

| | | | 100: CCIR-D3 | 101:Reserved |
| --- | --- | --- | --- | --- |
| | | | 110:Reserved | 111:IO Disable |
| 3 | / | / | / | |
| | | | PE8_SELECT | |
| | | | 000:Input | 001:Output |
| | | | 010:CSI_D4 | 011: Reserved |
| | | | 100: CCIR-D2 | 101:Reserved |
| 2:0 | R/W | 0x7 | 110:Reserved | 111:IO Disable |

### 3.24.2.30. PE Configure Register 2 (Default Value: 0x00007777)

| Offset: 0x98 | | | Register Name: **PE_CFG2_REG** | |
| --- | --- | --- | --- | --- |
| Bit | R/W | Default/Hex | Description | |
| 31:15 | / | / | / | |
| | | | PE19_SELECT | |
| | | | 000:Input | 001:Output |
| | | | 010:Reserved | 011:Reserved |
| | | | 100:Reserved | 101:Reserved |
| 14:12 | R/W | 0x7 | 110:Reserved | 111:IO Disable |
| 11 | / | / | / | |
| | | | PE18_SELECT | |
| | | | 000:Input | 001:Output |
| | | | 010:Reserved | 011:OWA_DOUT |
| | | | 100:Reserved | 101:Reserved |
| 10:8 | R/W | 0x7 | 110:Reserved | 111:IO Disable |
| 7 | / | / | / | |
| | | | PE17_SELECT | |
| | | | 000:Input | 001:Output |
| | | | 010:Reserved | 011:Reserved |
| | | | 100:Reserved | 101:Reserved |
| 6:4 | R/W | 0x7 | 110:Reserved | 111:IO Disable |
| 3 | / | / | / | |
| | | | PE16_SELECT | |
| | | | 000:Input | 001:Output |
| | | | 010:Reserved | 011: Reserved |
| | | | 100:Reserved | 101:Reserved |
| 2:0 | R/W | 0x7 | 110:Reserved | 111:IO Disable |

### 3.24.2.31. PE Configure Register 3 (Default Value: 0x00000000)

| Offset: 0x9C | | | Register Name: **PE_CFG3_REG** | |
| --- | --- | --- | --- | --- |
| Bit | R/W | Default/Hex | Description | |

| 31:0 | / | / | / |
|------|---|---|---|

### 3.24.2.32. PE Data Register (Default Value: 0x00000000)

| Offset: 0xA0 | | | Register Name: **PE_DATA _REG** |
|------|-----|-------------|-------------|
| Bit | R/W | Default/Hex | Description |
| 31:20 | / | / | / |
| 19:0 | R/W | 0x0 | PE_DAT<br>If the port is configured as input, the corresponding bit is the pin state. If the port is configured as output, the pin state is the same as the corresponding bit. The read bit value is the value setup by software. If the port is configured as functional pin, the undefined value will be read. |

### 3.24.2.33. PE Multi-Driving Register 0 (Default Value: 0x55555555)

| Offset: 0xA4 | | | Register Name: **PE_ DRV0 _REG** |
|------|-----|-------------|-------------|
| Bit | R/W | Default/Hex | Description |
| [2i+1:2i]<br>(i=0~15) | R/W | 0x1 | PE_DRV<br>PE[n] Multi-Driving SELECT (n = 0~15)<br>00: Level 0            01: Level 1<br>10: Level 2            11: Level 3 |

### 3.24.2.34. PE Multi-Driving Register 1 (Default Value: 0x00000055)

| Offset: 0xA8 | | | Register Name: **PE_ DRV1 _REG** |
|------|-----|-------------|-------------|
| Bit | R/W | Default/Hex | Description |
| 31:8 | / | / | / |
| [2i+1:2i]<br>(i=0~3) | R/W | 0x1 | PE_DRV<br>PE[n] Multi-Driving Select (n = 16~19)<br>00: Level 0            01: Level 1<br>10: Level 2            11: Level 3 |

### 3.24.2.35. PE PULL Register 0 (Default Value: 0x00000000)

| Offset: 0xAC | | | Register Name: **PE_ PULL0 _REG** |
|------|-----|-------------|-------------|
| Bit | R/W | Default/Hex | Description |
| [2i+1:2i]<br>(i=0~15) | R/W | 0x0 | PE_PULL<br>PE[n] Pull-up/down Select (n = 0~15)<br>00: Pull-up/down disable     01: Pull-up |

| | | | 10: Pull-down            11: Reserved |
|---|---|---|---|

### 3.24.2.36.   PE PULL Register 1 (Default Value: 0x00000000)

| Offset: 0xB0 | | | Register Name: **PE_ PULL1 _REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:8 | / | / | Reserved |
| [2i+1:2i]<br>(i=0~3) | R/W | 0x0 | PE_PULL<br>PE[n] Pull-up/down Select (n = 16~19)<br>00: Pull-up/down disable      01: Pull-up<br>10: Pull-down               11: Reserved |

### 3.24.2.37.   PF Configure Register 0 (Default Value: 0x07373733)

| Offset: 0xB4 | | | Register Name: **PF_CFG0_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:27 | / | / | / |
| 26:24 | R/W | 0x7 | PF6_SELECT<br>000:Input                001:Output<br>010:Reserved            011:Reserved<br>100:Reserved            101:Reserved<br>110:Reserved            111:IO Disable |
| 23 | | | |
| 22:20 | R/W | 0x3 | PF5_SELECT<br>000:Input                001:Output<br>010:SDC0_D2            011:JTAG_CK1<br>100:Reserved            101:Reserved<br>110:Reserved            111:IO Disable |
| 19 | / | / | / |
| 18:16 | R/W | 0x7 | PF4_SELECT<br>000:Input                001:Output<br>010:SDC0_D3            011:UART0_RX<br>100:Reserved            101:Reserved<br>110:Reserved            111:IO Disable |
| 15 | / | / | / |
| 14:12 | R/W | 0x3 | PF3_SELECT<br>000:Input                001:Output<br>010:SDC0_CMD            011:JTAG_DO1<br>100:Reserved            101:Reserved<br>110:Reserved            111:IO Disable |
| 11 | / | / | / |
| 10:8 | R/W | 0x7 | PF2_SELECT |

| | | | 000:Input | 001:Output |
|---|---|---|---|---|
| | | | 010:SDC0_CLK | 011:UART0_TX |
| | | | 100:Reserved | 101:Reserved |
| | | | 110:Reserved | 111:IO Disable |
| 7 | / | / | / | |
| | | | PF1_SELECT | |
| | | | 000:Input | 001:Output |
| | | | 010:SDC0_D0 | 011:JTAG_DI1 |
| | | | 100:Reserved | 101:Reserved |
| 6:4 | R/W | 0x3 | 110:Reserved | 111:IO Disable |
| 3 | / | / | / | |
| | | | PF0_SELECT | |
| | | | 000:Input | 001:Output |
| | | | 010:SDC0_D1 | 011:JTAG_MS1 |
| | | | 100:Reserved | 101:Reserved |
| 2:0 | R/W | 0x3 | 110:Reserved | 111:IO Disable |

### 3.24.2.38. PF Configure Register 1 (Default Value: 0x00000000)

| Offset: 0xB8 | | | Register Name: **PF_CFG1 _REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | / | / | / |

### 3.24.2.39. PF Configure Register 2(Default Value: 0x00000000)

| Offset: 0xBC | | | Register Name: **PF_CFG2 _REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | / | / | / |

### 3.24.2.40. PF Configure Register 3(Default Value: 0x00000000)

| Offset: 0xC0 | | | Register Name: **PF_CFG3_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | / | / | / |

### 3.24.2.41. PF Data Register (Default Value: 0x00000000)

| Offset: 0xC4 | | | Register Name: **PF_DATA_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |

| 31:7 | / | / | / |
|------|---|---|---|
| 6:0 | R/W | 0x0 | PF_DAT<br>If the port is configured as input, the corresponding bit is the pin state. If the port is configured as output, the pin state is the same as the corresponding bit. The read bit value is the value setup by software. If the port is configured as functional pin, the undefined value will be read. |

### 3.24.2.42.　PF Multi-Driving Register 0 (Default Value: 0x00001555)

| Offset: 0xC8 | | | Register Name: **PF_DRV0_REG** |
|--------------|---|---|--------------------------------|
| Bit | R/W | Default/Hex | Description |
| 31:14 | / | / | / |
| [2i+1:2i]<br>(i=0~6) | R/W | 0x1 | PF_DRV<br>PF[n] Multi-Driving SELECT (n = 0~6)<br>00: Level 0 　　　　　　01: Level 1<br>10: Level 2 　　　　　　11: Level 3 |

### 3.24.2.43.　PF Multi-Driving Register 1 (Default Value: 0x00000000)

| Offset: 0xCC | | | Register Name: **PF_DRV1_REG** |
|--------------|---|---|--------------------------------|
| Bit | R/W | Default/Hex | Description |
| 31:0 | / | / | / |

### 3.24.2.44.　PF PULL Register 0 (Default Value: 0x00000000)

| Offset: 0xD0 | | | Register Name: **PF_ PULL0_REG** |
|--------------|---|---|----------------------------------|
| Bit | R/W | Default/Hex | Description |
| 31:14 | / | / | / |
| [2i+1:2i]<br>(i=0~6) | R/W | 0x0 | PF_PULL<br>PF[n] Pull-up/down Select (n = 0~6)<br>00: Pull-up/down disable 　01: Pull-up<br>10: Pull-down 　　　　　　11: Reserved |

### 3.24.2.45.　PF PULL Register 1 (Default Value: 0x00000000)

| Offset: 0xD4 | | | Register Name: **PF_PULL1_REG** |
|--------------|---|---|---------------------------------|
| Bit | R/W | Default/Hex | Description |
| 31:0 | / | / | Reserved |

### 3.24.2.46. PG Configure Register 0 (Default Value: 0x77777777)

| Offset: 0xD8 | | | Register Name: **PG_CFG0_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | / | / | / |
| 30:28 | R/W | 0x7 | PG7_SELECT<br>000:Input         001:Output<br>010:UART1_RX    011:SPI1_CLK<br>100:Reserved    101:Reserved<br>110:PG_EINT7    111:IO Disable |
| 27 | / | / | / |
| 26:24 | R/W | 0x7 | PG6_SELECT<br>000:Input         001:Output<br>010:UART1_TX    011:SPI1_CS<br>100:Reserved    101:Reserved<br>110:PG_EINT6    111:IO Disable |
| 23 | / | / | / |
| 22:20 | R/W | 0x7 | PG5_SELECT<br>000:Input         001:Output<br>010:SDC1_D3    011:Reserved<br>100:Reserved    101:Reserved<br>110:PG_EINT5    111:IO Disable |
| 19 | / | / | / |
| 18:16 | R/W | 0x7 | PG4_SELECT<br>000:Input         001:Output<br>010:SDC1_D2    011:Reserved<br>100:Reserved    101:Reserved<br>110:PG_EINT4    111:IO Disable |
| 15 | / | / | / |
| 14:12 | R/W | 0x7 | PG3_SELECT<br>000:Input         001:Output<br>010:SDC1_D1    011:Reserved<br>100:Reserved    101:Reserved<br>110:PG_EINT3    111:IO Disable |
| 11 | / | / | / |
| 10:8 | R/W | 0x7 | PG2_SELECT<br>000:Input         001:Output<br>010:SDC1_D0    011:Reserved<br>100:Reserved    101:Reserved<br>110:PG_EINT2    111:IO Disable |
| 7 | / | / | / |
| 6:4 | R/W | 0x7 | PG1_SELECT<br>000:Input         001:Output<br>010:SDC1_CMD    011:Reserved |

| | | | |
|---|---|---|---|
| | | | 100:Reserved          101:Reserved |
| | | | 110:PG_EINT1          111:IO Disable |
| 3 | / | / | / |
| | | | PG0_SELECT |
| | | | 000:Input          001:Output |
| | | | 010:SDC1_CLK      011:Reserved |
| | | | 100:Reserved      101:Reserved |
| 2:0 | R/W | 0x7 | 110:PG_EINT0      111:IO Disable |

### 3.24.2.47.    PG Configure Register 1 (Default Value: 0x00777777)

| Offset: 0xDC | | | Register Name: **PG_CFG1_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:23 | / | / | / |
| | | | PG13_SELECT |
| | | | 000:Input          001:Output |
| | | | 010:I2S1_DIN       011:UART3_CTS |
| | | | 100:Reserved      101:Reserved |
| 22:20 | R/W | 0x7 | 110:PG_EINT13     111:IO Disable |
| 19 | / | / | / |
| | | | PG12_SELECT |
| | | | 000:Input          001:Output |
| | | | 010:I2S1_DOUT    011:UART3_RTS |
| | | | 100:Reserved      101:Reserved |
| 18:16 | R/W | 0x7 | 110:PG_EINT12     111:IO Disable |
| 15 | / | / | / |
| | | | PG11_SELECT |
| | | | 000:Input          001:Output |
| | | | 010:I2S1_LRCK     011:UART3_RX |
| | | | 100:Reserved      101:Reserved |
| 14:12 | R/W | 0x7 | 110:PG_EINT11     111:IO Disable |
| 11 | / | / | / |
| | | | PG10_SELECT |
| | | | 000:Input          001:Output |
| | | | 010:I2S1_BCLK     011:UART3_TX |
| | | | 100:Reserved      101:Reserved |
| 10:8 | R/W | 0x7 | 110:PG_EINT10     111:IO Disable |
| 7 | / | / | / |
| | | | PG9_SELECT |
| | | | 000:Input          001:Output |
| | | | 010:UART1_CTS    011:SPI1_MISO |
| | | | 100:Reserved      101:Reserved |
| 6:4 | R/W | 0x7 | 110:PG_EINT9      111:IO Disable |
| 3 | / | / | / |
| 2:0 | R/W | 0x7 | PG8_SELECT |

| | | | |
|---|---|---|---|
| 000:Input | | 001:Output | |
| 010:UART1_RTS | | 011:SPI1_MOSI | |
| 100:Reserved | | 101:Reserved | |
| 110:PG_EINT8 | | 111:IO Disable | |

### 3.24.2.48.　PG Configure Register 2 (Default Value: 0x00000000)

| Offset: 0xE0 | | | Register Name: **PG_CFG2_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | / | / | / |

### 3.24.2.49.　PG Configure Register 3 (Default Value: 0x00000000)

| Offset: 0xE4 | | | Register Name: **PG_CFG3_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | / | / | / |

### 3.24.2.50.　PG Data Register (Default Value: 0x00000000)

| Offset: 0xE8 | | | Register Name: **PG_DATA_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:14 | / | / | / |
| 13:0 | R/W | 0x0 | PG_DAT<br>If the port is configured as input, the corresponding bit is the pin state. If the port is configured as output, the pin state is the same as the corresponding bit. The read bit value is the value setup by software. If the port is configured as functional pin, the undefined value will be read. |

### 3.24.2.51.　PG Multi-Driving Register 0 (Default Value: 0x05555555)

| Offset: 0xEC | | | Register Name: **PG_ DRV0_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:28 | / | / | / |
| [2i+1:2i]<br>(i=0~13) | R/W | 0x1 | PF_DRV<br>PF[n] Multi-Driving SELECT (n = 0~13)<br>00: Level 0　　　　　　　01: Level 1<br>10: Level 2　　　　　　　11: Level 3 |

### 3.24.2.52.   PG Multi-Driving Register 1 (Default Value: 0x00000000)

| Offset: 0xF0 | | | Register Name: **PG_ DRV1_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | / | / | / |

### 3.24.2.53.   PG PULL Register 0 (Default Value: 0x00000000)

| Offset: 0xF4 | | | Register Name: **PG_ PULL0_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:28 | / | / | / |
| [2i+1:2i] (i=0~13) | R/W | 0x5140 | PF_PULL<br>PF[n] Pull-up/down Select (n = 0~13)<br>00: Pull-up/down disable     01: Pull-up<br>10: Pull-down                11: Reserved |

### 3.24.2.54.   PG PULL Register 1 (Default Value: 0x00000000)

| Offset: 0xF8 | | | Register Name: **PG_ PULL1_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | / | / | / |

### 3.24.2.55.   PH Configure Register 0 (Default Value: 0x77777777)

| Offset: 0xFC | | | Register Name: **PH_CFG0_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | / | / | / |
| 30:28 | R/W | 0x7 | PH7_SELECT<br>000:Input                001:Output<br>010:HSDA                011:Reserved<br>100:Reserved            101:Reserved<br>110:PH_EINT7          111:IO Disable |
| 27 | / | / | / |
| 26:24 | R/W | 0x7 | PH6_SELECT<br>000:Input                001:Output<br>010:HSCL                011:Reserved<br>100:Reserved            101:Reserved<br>110: PH_EINT6          111:IO Disable |
| 23 | / | / | / |
| 22:20 | R/W | 0x7 | PH5_SELECT |

| | | | 000:Input              001:Output |
|---|---|---|---|
| | | | 010:TWI2_SDA          011:Reserved |
| | | | 100:Reserved          101:Reserved |
| | | | 110: PH_EINT5         111:IO Disable |
| 19 | / | / | / |
| | | | PH4_SELECT |
| | | | 000:Input              001:Output |
| | | | 010:TWI2_SCK          011:Reserved |
| | | | 100:Reserved          101:Reserved |
| 18:16 | R/W | 0x7 | 110: PH_EINT4         111:IO Disable |
| 15 | / | / | / |
| | | | PH3_SELECT |
| | | | 000:Input              001:Output |
| | | | 010:TWI1_SDA          011:Reserved |
| | | | 100:Reserved          101:Reserved |
| 14:12 | R/W | 0x7 | 110: PH_EINT3         111:IO Disable |
| 11 | / | / | / |
| | | | PH2_SELECT |
| | | | 000:Input              001:Output |
| | | | 010:TWI1_SCK          011:Reserved |
| | | | 100:Reserved          101:Reserved |
| 10:8 | R/W | 0x7 | 110: PH_EINT2         111:IO Disable |
| 7 | / | / | / |
| | | | PH1_SELECT |
| | | | 000:Input              001:Output |
| | | | 010:TWI0_SDA          011:Reserved |
| | | | 100:Reserved          101:Reserved |
| 6:4 | R/W | 0x7 | 110: PH_EINT1         111:IO Disable |
| 3 | / | / | / |
| | | | PH0_SELECT |
| | | | 000:Input              001:Output |
| | | | 010:TWI0_SCK          011:Reserved |
| | | | 100:Reserved          101:Reserved |
| 2:0 | R/W | 0x7 | 110: PH_EINT0         111:IO Disable |

### 3.24.2.56.   PH Configure Register 1 (Default Value: 0x00007777)

| Offset: 0x100 | | | Register Name: **PH_CFG1_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:7 | / | / | / |
| | | | PH11_SELECT |
| | | | 000:Input              001:Output |
| | | | 010:Reserved          011:Reserved |
| 14:12 | R/W | 0x7 | 100:Reserved          101:Reserved |

| | | | 110: PH_EINT11          111:IO Disable |
|---|---|---|---|
| 11 | / | / | / |
| 10:8 | R/W | 0x7 | PH10_SELECT<br>000:Input                001:Output<br>010:Reserved            011:Reserved<br>100:Reserved            101:Reserved<br>110: PH_EINT10          111:IO Disable |
| 7 | / | / | / |
| 6:4 | R/W | 0x7 | PH9_SELECT<br>000:Input                001:Output<br>010:Reserved            011:Reserved<br>100:Reserved            101:Reserved<br>110: PH_EINT9            111:IO Disable |
| 3 | / | / | / |
| 2:0 | R/W | 0x7 | PH8_SELECT<br>000:Input                001:Output<br>010:HCEC                011:Reserved<br>100:Reserved            101:Reserved<br>110: PH_EINT8            111:IO Disable |

### 3.24.2.57.   PH Configure Register 2 (Default Value: 0x00000000)

| Offset: 0x104 | | | Register Name: **PH_CFG2_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | / | / | / |

### 3.24.2.58.   PH Configure Register 3 (Default Value: 0x00000000)

| Offset: 0x108 | | | Register Name: **PH_CFG3_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | / | / | / |

### 3.24.2.59.   PH Data Register (Default Value: 0x00000000)

| Offset: 0x10C | | | Register Name: **PH_DATA_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:12 | / | / | / |
| 11:0 | R/W | 0x0 | PH_DAT<br>If the port is configured as input, the corresponding bit is the pin state. If the port is configured as output, the pin state is the same as the corresponding bit. The read bit value is the value setup by software. If the |

| | | | port is configured as functional pin, the undefined value will be read. |
|---|---|---|---|

### 3.24.2.60. PH Multi-Driving Register 0 (Default Value: 0x00555555)

| Offset: 0x110 | | | Register Name: **PH_ DRV0_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:24 | / | / | / |
| [2i+1:2i]<br>(i=0~11) | R/W | 0x1 | PH_DRV<br>PH[n] Multi-Driving SELECT (n = 0~11)<br>00: Level 0        01: Level 1<br>10: Level 2        11: Level 3 |

### 3.24.2.61. PH Multi-Driving Register 1 (Default Value: 0x00000000)

| Offset: 0x114 | | | Register Name: **PH_ DRV1_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | / | / | / |

### 3.24.2.62. PH PULL Register 0 (Default Value: 0x00000000)

| Offset: 0x118 | | | Register Name: **PH_ PULL0_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:24 | / | / | / |
| [2i+1:2i]<br>(i=0~11) | R/W | 0x0 | PH_PULL<br>PH[n] Pull-up/down Select (n = 0~11)<br>00: Pull-up/down disable    01: Pull-up<br>10: Pull-down            11: Reserved |

### 3.24.2.63. PH PULL Register 1 (Default Value: 0x00000000)

| Offset: 0x11C | | | Register Name: **PH_ PULL1_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | / | / | / |

### 3.24.2.64. PB External Interrupt Configure Register 0 (Default Value: 0x00000000)

| Offset: 0x200 | | | Register Name: **PB_EINT_CFG0_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |

| | | | EINT_CFG |
|---|---|---|---|
| | | | External INTn Mode (n = 0~7) |
| | | | 0x0: Positive Edge |
| | | | 0x1: Negative Edge |
| | | | 0x2: High Level |
| | | | 0x3: Low Level |
| [4i+3:4i] | | | 0x4: Double Edge (Positive/ Negative) |
| (i=0~7) | R/W | 0 | Others: Reserved |

### 3.24.2.65.  PB External Interrupt Configure Register 1 (Default Value: 0x00000000)

| Offset: 0x204 | | | Register Name: **PB_EINT_CFG1_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:12 | / | / | / |
| | | | EINT_CFG |
| | | | External INTn Mode (n = 8~10) |
| | | | 0x0: Positive Edge |
| | | | 0x1: Negative Edge |
| | | | 0x2: High Level |
| | | | 0x3: Low Level |
| [4i+3:4i] | | | 0x4: Double Edge (Positive/ Negative) |
| (i=0~2) | R/W | 0 | Others: Reserved |

### 3.24.2.66.  PB External Interrupt Configure Register 2 (Default Value: 0x00000000)

| Offset: 0x208 | | | Register Name: **PB_EINT_CFG2_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | / | / | / |

### 3.24.2.67.  PB External Interrupt Configure Register 3 (Default Value: 0x00000000)

| Offset: 0x20C | | | Register Name: **PB_EINT_CFG3_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | / | / | / |

### 3.24.2.68.  PB External Interrupt Control Register (Default Value: 0x00000000)

| Offset: 0x210 | | | Register Name: **PB_EINT _CTL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |

| 31:11 | / | / | / |
|---|---|---|---|
| [n]<br>(n=0~10) | R/W | 0 | EINT_CTL<br>External INTn Enable (n = 0~10)<br>0: Disable<br>1: Enable |

### 3.24.2.69.    PB External Interrupt Status Register (Default Value: 0x00000000)

| Offset: 0x214 | | | Register Name: **PB_EINT _STATUS_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:11 | / | / | / |
| [n]<br>(n=0~10) | R/W | 0 | EINT_STATUS<br>External INTn Pending Bit (n = 0~10)<br>0: No IRQ pending<br>1: IRQ pending<br>Write '1' to clear |

### 3.24.2.70.    PB External Interrupt Debounce Register (Default Value: 0x00000000)

| Offset: 0x218 | | | Register Name: **PB_EINT _DEB_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:7 | / | / | / |
| 6:4 | R/W | 0 | DEB_CLK_PRE_SCALE<br>Debounce Clock Pre-scale n<br>The selected clock source is prescaled by 2^n. |
| 3:1 | / | / | / |
| 0 | R/W | 0 | PIO_INT_CLK_SELECT<br>PIO Interrupt Clock Select<br>0: LOSC 32Khz<br>1: HOSC 24Mhz |

### 3.24.2.71.    PG External Interrupt Configure Register 0 (Default Value: 0x00000000)

| Offset: 0x220 | | | Register Name: **PG_EINT_CFG0_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| [4i+3:4i]<br>(i=0~7) | R/W | 0 | EINT_CFG<br>External INTn Mode (n = 0~7)<br>0x0: Positive Edge<br>0x1: Negative Edge<br>0x2: High Level<br>0x3: Low Level |

| | | | 0x4: Double Edge (Positive/ Negative) |
|---|---|---|---|
| | | | Others: Reserved |

### 3.24.2.72. PG External Interrupt Configure Register 1 (Default Value: 0x00000000)

| Offset: 0x224 | | | Register Name: **PG_EINT_CFG1_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:24 | / | / | / |
| [4i+3:4i]<br>(i=0~5) | R/W | 0 | EINT_CFG<br>External INTn Mode (n = 8~13)<br>0x0: Positive Edge<br>0x1: Negative Edge<br>0x2: High Level<br>0x3: Low Level<br>0x4: Double Edge (Positive/ Negative)<br>Others: Reserved |

### 3.24.2.73. PG External Interrupt Configure Register 2 (Default Value: 0x00000000)

| Offset: 0x228 | | | Register Name: **PG_EINT_CFG2_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | / | / | / |

### 3.24.2.74. PG External Interrupt Configure Register 3 (Default Value: 0x00000000)

| Offset: 0x22C | | | Register Name: **PG_EINT_CFG3_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | / | / | / |

### 3.24.2.75. PG External Interrupt Control Register (Default Value: 0x00000000)

| Offset: 0x230 | | | Register Name: **PG_EINT _CTL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:14 | / | / | / |
| [n]<br>(n=0~13) | R/W | 0 | EINT_CTL<br>External INTn Enable (n = 0~13)<br>0: Disable<br>1: Enable |

### 3.24.2.76.  PG External Interrupt Status Register (Default Value: 0x00000000)

| Offset: 0x234 | | | Register Name: **PG_EINT _STATUS_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:14 | / | / | / |
| [n] (n=0~13) | R/W | 0 | EINT_STATUS External INTn Pending Bit (n = 0~13) 0: No IRQ pending 1: IRQ pending Write '1' to clear |

### 3.24.2.77.  PG External Interrupt Debounce Register (Default Value: 0x00000000)

| Offset: 0x238 | | | Register Name: **PG_EINT _DEB_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:7 | / | / | / |
| 6:4 | R/W | 0 | DEB_CLK_PRE_SCALE Debounce Clock Pre-scale n The selected clock source is prescaled by 2^n. |
| 3:1 | / | / | / |
| 0 | R/W | 0 | PIO_INT_CLK_SELECT PIO Interrupt Clock Select 0: LOSC 32Khz 1: HOSC 24Mhz |

### 3.24.2.78.  PH External Interrupt Configure Register 0 (Default Value: 0x00000000)

| Offset: 0x240 | | | Register Name: **PH_EINT_CFG0_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| [4i+3:4i] (i=0~7) | R/W | 0 | EINT_CFG External INTn Mode (n = 0~7) 0x0: Positive Edge 0x1: Negative Edge 0x2: High Level 0x3: Low Level 0x4: Double Edge (Positive/ Negative) Others: Reserved |

### 3.24.2.79.    PH External Interrupt Configure Register 1 (Default Value: 0x00000000)

| Offset: 0x244 | | | Register Name: **PH_EINT_CFG1_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:16 | / | / | / |
| [4i+3:4i]<br>(i=0~3) | R/W | 0 | EINT_CFG<br>External INTn Mode (n = 8~11)<br>0x0: Positive Edge<br>0x1: Negative Edge<br>0x2: High Level<br>0x3: Low Level<br>0x4: Double Edge (Positive/ Negative)<br>Others: Reserved |

### 3.24.2.80.    PH External Interrupt Configure Register 2 (Default Value: 0x00000000)

| Offset: 0x248 | | | Register Name: **PH_EINT_CFG2_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | / | / | / |

### 3.24.2.81.    PH External Interrupt Configure Register 3 (Default Value: 0x00000000)

| Offset: 0x24C | | | Register Name: **PH_EINT_CFG3_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | / | / | / |

### 3.24.2.82.    PH External Interrupt Control Register (Default Value: 0x00000000)

| Offset: 0x250 | | | Register Name: **PH_EINT_CTL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:12 | / | / | / |
| [n]<br>(n=0~11) | R/W | 0 | EINT_CTL<br>External INTn Enable (n = 0~11)<br>0: Disable<br>1: Enable |

### 3.24.2.83.    PH External Interrupt Status Register (Default Value: 0x00000000)

| Offset: 0x254 | Register Name: **PH_EINT_STATUS_REG** |
|---|---|

| Bit | R/W | Default/Hex | Description |
|---|---|---|---|
| 31:12 | / | / | / |
| [n]<br>(n=0~11) | R/W | 0 | EINT_STATUS<br>External INTn Pending Bit (n = 0~11)<br>0: No IRQ pending<br>1: IRQ pending<br>Write '1' to clear |

### 3.24.2.84. PH External Interrupt Debounce Register (Default Value: 0x00000000)

| Offset: 0x258 | | | Register Name: **PH_EINT _DEB_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:7 | / | / | / |
| 6:4 | R/W | 0 | DEB_CLK_PRE_SCALE<br>Debounce Clock Pre-scale n<br>The selected clock source is prescaled by 2^n. |
| 3:1 | / | / | / |
| 0 | R/W | 0 | PIO_INT_CLK_SELECT<br>PIO Interrupt Clock Select<br>0: LOSC 32Khz<br>1: HOSC 24Mhz |

### 3.24.2.85. PB Group Configuration Register (Default Value: 0x0000000D)

| Offset: 0x304 | | | Register Name: **PB_GRP_CONFIG_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:4 | / | / | / |
| 3:0 | R/W | 0xD | IO_BIAS_CONFIG<br>IO Pad Bias Configuration Value<br>0x0: for 1.8V Power Supply;<br>0x6: for 2.5V Power Supply;<br>0x9: for 2.8V Power Supply;<br>0xA: for 3.0V Power Supply;<br>0xD: for 3.3V Power Supply;<br>Others: Reserved.<br>*Note: The configuration value must greater than or equal to the corresponding value of the voltage supplied to the IO Group.* |

### 3.24.2.86. PC Group Configuration Register (Default Value: 0x00000006)

| Offset: 0x308 | | | Register Name: **PC_GRP_CONFIG_REG** |
|---|---|---|---|

| Bit | R/W | Default/Hex | Description |
|---|---|---|---|
| 31:4 | / | / | / |
| 3:0 | R/W | 0x6 | IO_BIAS_CONFIG<br>IO Pad Bias Configuration Value<br>0x0: for 1.8V Power Supply;<br>0x6: for 2.5V Power Supply;<br>0x9: for 2.8V Power Supply;<br>0xA: for 3.0V Power Supply;<br>0xD: for 3.3V Power Supply;<br>Others: Reserved.<br>*Note: The configuration value must greater than or equal to the corresponding value of the voltage supplied to the IO Group.* |

### 3.24.2.87.   PD Group Configuration Register (Default Value: 0x0000000D)

| Offset: 0x30C | | | Register Name: **PD_GRP_CONFIG_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:4 | / | / | / |
| 3:0 | R/W | 0xD | IO_BIAS_CONFIG<br>IO Pad Bias Configuration Value<br>0x0: for 1.8V Power Supply;<br>0x6: for 2.5V Power Supply;<br>0x9: for 2.8V Power Supply;<br>0xA: for 3.0V Power Supply;<br>0xD: for 3.3V Power Supply;<br>Others: Reserved.<br>*Note: The configuration value must greater than or equal to the corresponding value of the voltage supplied to the IO Group.* |

### 3.24.2.88.   PE Group Configuration Register (Default Value: 0x0000000D)

| Offset: 0x310 | | | Register Name: **PE_GRP_CONFIG_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:4 | / | / | / |
| 3:0 | R/W | 0xD | IO_BIAS_CONFIG<br>IO Pad Bias Configuration Value<br>0x0: for 1.8V Power Supply;<br>0x6: for 2.5V Power Supply;<br>0x9: for 2.8V Power Supply;<br>0xA: for 3.0V Power Supply;<br>0xD: for 3.3V Power Supply;<br>Others: Reserved. |

| | | | Note: The configuration value must greater than or equal to the corresponding value of the voltage supplied to the IO Group. |
|---|---|---|---|

### 3.24.2.89.  PF Group Configuration Register (Default Value: 0x0000000D)

| Offset: 0x314 | | | Register Name: **PF_GRP_CONFIG_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:4 | / | / | / |
| 3:0 | R/W | 0xD | IO_BIAS_CONFIG<br>IO Pad Bias Configuration Value<br>0x0: for 1.8V Power Supply;<br>0x6: for 2.5V Power Supply;<br>0x9: for 2.8V Power Supply;<br>0xA: for 3.0V Power Supply;<br>0xD: for 3.3V Power Supply;<br>Others: Reserved.<br>*Note: The configuration value must greater than or equal to the corresponding value of the voltage supplied to the IO Group.* |

### 3.24.2.90.  PG Group Configuration Register (Default Value: 0x0000000D)

| Offset: 0x318 | | | Register Name: **PG_GRP_CONFIG_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:4 | / | / | / |
| 3:0 | R/W | 0xD | IO_BIAS_CONFIG<br>IO Pad Bias Configuration Value<br>0x0: for 1.8V Power Supply;<br>0x6: for 2.5V Power Supply;<br>0x9: for 2.8V Power Supply;<br>0xA: for 3.0V Power Supply;<br>0xD: for 3.3V Power Supply;<br>Others: Reserved.<br>*Note: The configuration value must greater than or equal to the corresponding value of the voltage supplied to the IO Group.* |

### 3.24.2.91.  PH Group Configuration Register (Default Value: 0x0000000D)

| Offset: 0x31C | | | Register Name: **PH_GRP_CONFIG_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:4 | / | / | / |
| 3:0 | R/W | 0xD | IO_BIAS_CONFIG |

System

| | | | IO Pad Bias Configuration Value |
|---|---|---|---|
| | | | 0x0: for 1.8V Power Supply; |
| | | | 0x6: for 2.5V Power Supply; |
| | | | 0x9: for 2.8V Power Supply; |
| | | | 0xA: for 3.0V Power Supply; |
| | | | 0xD: for 3.3V Power Supply; |
| | | | Others: Reserved. |
| | | | *Note: The configuration value must greater than or equal to the corresponding value of the voltage supplied to the IO Group.* |

### 3.24.2.92. PL Configure Register 0 (Default Value: 0x77777777)

| Offset: 0x00 | | | Register Name: **PL_CFG0 _REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | / | / | / |
| 30:28 | R/W | 0x7 | PL7_SELECT<br>000:Input 001:Output<br>010:S_JTAG_DI 011:Reserved<br>100:Reserved 101:Reserved<br>110:S_PL_EINT7 111:IO Disable |
| 27 | / | / | / |
| 26:24 | R/W | 0x7 | PL6_SELECT<br>000:Input 001:Output<br>010:S_JTAG_DO 011:Reserved<br>100:Reserved 101:Reserved<br>110:S_PL_EINT6 111:IO Disable |
| 23 | / | / | / |
| 22:20 | R/W | 0x7 | PL5_SELECT<br>000:Input 001:Output<br>010:S_JTAG_CK 011:Reserved<br>100:Reserved 101:Reserved<br>110:S_PL_EINT5 111:IO Disable |
| 19 | / | / | / |
| 18:16 | R/W | 0x7 | PL4_SELECT<br>000:Input 001:Output<br>010:S_JTAG_MS 011:Reserved<br>100:Reserved 101:Reserved<br>110:S_PL_EINT4 111:IO Disable |
| 15 | / | / | / |
| 14:12 | R/W | 0x7 | PL3_SELECT<br>000:Input 001:Output<br>010:S_UART_RX 011:Reserved<br>100:Reserved 101:Reserved<br>110:S_PL_EINT3 111:IO Disable |

| 11 | / | / | / |
|---|---|---|---|
| 10:8 | R/W | 0x7 | PL2_SELECT<br>000:Input      001:Output<br>010:S_UART_TX      011:Reserved<br>100:Reserved      101:Reserved<br>110:S_PL_EINT2      111:IO Disable |
| 7 | / | / | / |
| 6:4 | R/W | 0x7 | PL1_SELECT<br>000:Input      001:Output<br>010:S_RSB_SDA      011:S_TWI_SDA<br>100:Reserved      101:Reserved<br>110:S_PL_EINT1      111:IO Disable |
| 3 | / | / | / |
| 2:0 | R/W | 0x7 | PL0_SELECT<br>000:Input      001:Output<br>010:S_RSB_SCK      011:S_TWI_SCK<br>100:Reserved      101:Reserved<br>110:S_PL_EINT0      111:IO Disable |

### 3.24.2.93. PL Configure Register 1 (Default Value: 0x00077777)

| Offset: 0x04 | | | Register Name: **PL_CFG1_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | / | / | / |
| 18:16 | R/W | 0x7 | PL12_SELECT<br>000:Input      001:Output<br>010:S_CIR_RX      011:Reserved<br>100:Reserved      101:Reserved<br>110:S_PL_EINT12      111:IO Disable |
| 15 | / | / | / |
| 14:12 | R/W | 0x7 | PL11_SELECT<br>000:Input      001:Output<br>010: Reserved      011:Reserved<br>100:Reserved      101:Reserved<br>110:S_PL_EINT11      111:IO Disable |
| 11 | / | / | / |
| 10:8 | R/W | 0x7 | PL10_SELECT<br>000:Input      001:Output<br>010:S_PWM      011:Reserved<br>100:Reserved      101:Reserved<br>110:S_PL_EINT10      111:IO Disable |
| 7 | / | / | / |
| 6:4 | R/W | 0x7 | PL9_SELECT<br>000:Input      001:Output |

| | | | 010:S_TWI_SDA 011:Reserved |
|---|---|---|---|
| | | | 100:Reserved 101:Reserved |
| | | | 110:S_PL_EINT9 111:IO Disable |
| 3 | / | / | / |
| | | | PL8_SELECT |
| | | | 000:Input 001:Output |
| | | | 010:S_TWI_SCK 011:Reserved |
| | | | 100:Reserved 101:Reserved |
| 2:0 | R/W | 0x7 | 110:S_PL_EINT8 111:IO Disable |

### 3.24.2.94. PL Configure Register 2 (Default Value: 0x00000000)

| Offset: 0x08 | | | Register Name: **PL_CFG2_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | / | / | / |

### 3.24.2.95. PL Configure Register 3 (Default Value: 0x00000000)

| Offset: 0x0C | | | Register Name: **PL_CFG3_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | / | / | / |

### 3.24.2.96. PL Data Register (Default Value: 0x00000000)

| Offset: 0x10 | | | Register Name: **PL_DATA_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:19 | / | / | / |
| 12:0 | R/W | 0x0 | PL_DAT<br>If the port is configured as input, the corresponding bit is the pin state. If the port is configured as output, the pin state is the same as the corresponding bit. The read bit value is the value setup by software. If the port is configured as functional pin, the undefined value will be read. |

### 3.24.2.97. PL Multi-Driving Register 0 (Default Value: 0x01555555)

| Offset: 0x14 | | | Register Name: **PL_ DRV0_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:26 | / | / | / |
| [2i+1:2i] | R/W | 0x1 | PL_DRV |

| (i=0~12) | | | PL[n] Multi-Driving SELECT (n = 0~12) |
| | | | 00: Level 0            01: Level 1 |
| | | | 10: Level 2            11: Level 3 |

### 3.24.2.98. PL Multi-Driving Register 1 (Default Value: 0x00000000)

| Offset: 0x18 | | | Register Name: **PL_DRV1_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | / | / | / |

### 3.24.2.99. PL PULL Register 0 (Default Value: 0x00000005)

| Offset: 0x1C | | | Register Name: **PL_ PULL0_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:26 | / | / | / |
| [2i+1:2i] (i=0~12) | R/W | 0x5 | PF_PULL<br>PF[n] Pull-up/down Select (n = 0~12)<br>00: Pull-up/down disable    01: Pull-up<br>10: Pull-down            11: Reserved |

### 3.24.2.100. PL PULL Register 1 (Default Value: 0x00000000)

| Offset: 0x20 | | | Register Name: **PL_ PULL1_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | / | / | / |

### 3.24.2.101. PL External Interrupt Configure Register 0 (Default Value: 0x00000000)

| Offset: 0x200 | | | Register Name: **PL_EINT_CFG0_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| [4i+3:4i] (i=0~7) | R/W | 0 | EINT_CFG<br>External INTn Mode (n = 0~7)<br>0x0: Positive Edge<br>0x1: Negative Edge<br>0x2: High Level<br>0x3: Low Level<br>0x4: Double Edge (Positive/ Negative)<br>Others: Reserved |

### 3.24.2.102. PL External Interrupt Configure Register 1 (Default Value: 0x00000000)

| Offset: 0x204 | | | Register Name: **PL_EINT_CFG1_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:20 | / | / | / |
| [4i+3:4i] (i=0~4) | R/W | 0 | EINT_CFG<br>External INTn Mode (n = 8~12)<br>0x0: Positive Edge<br>0x1: Negative Edge<br>0x2: High Level<br>0x3: Low Level<br>0x4: Double Edge (Positive/ Negative)<br>Others: Reserved |

### 3.24.2.103. PL External Interrupt Configure Register 2 (Default Value: 0x00000000)

| Offset: 0x208 | | | Register Name: **PL_EINT_CFG2_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | / | / | / |

### 3.24.2.104. PL External Interrupt Configure Register 3 (Default Value: 0x00000000)

| Offset: 0x20C | | | Register Name: **PL_EINT_CFG3_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | / | / | / |

### 3.24.2.105. PL External Interrupt Control Register (Default Value: 0x00000000)

| Offset: 0x210 | | | Register Name: **PL_EINT_CTL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:13 | / | / | / |
| [n] (n=0~12) | R/W | 0 | EINT_CTL<br>External INTn Enable (n = 0~12)<br>0: Disable<br>1: Enable |

### 3.24.2.106. PL External Interrupt Status Register (Default Value: 0x00000000)

| Offset: 0x214 | | | Register Name: **PL_EINT_STATUS_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |

| 31:13 | / | / | / |
|---|---|---|---|
| [n]<br>(n=0~12) | R/W | 0 | EINT_STATUS<br>External INTn Pending Bit (n = 0~12)<br>0: No IRQ pending<br>1: IRQ pending<br>Write '1' to clear it. |

**3.24.2.107. PL External Interrupt Debounce Register (Default Value: 0x00000000)**

| Offset: 0x218 | | | Register Name: **PL_EINT_DEB_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:7 | / | / | / |
| 6:4 | R/W | 0 | DEB_CLK_PRE_SCALE<br>Debounce Clock Pre-scale n<br>The selected clock source is prescaled by 2^n. |
| 3:1 | / | / | / |
| 0 | R/W | 0 | PIO_INT_CLK_SELECT<br>PIO Interrupt Clock Select<br>0: LOSC 32Khz<br>1: HOSC 24Mhz |

**3.24.2.108. PL Group Configuration Register (Default Value: 0x0000000D)**

| Offset: 0x300 | | | Register Name: **PL_GRP_CONFIG_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:4 | / | / | / |
| 3:0 | R/W | 0xD | IO_BIAS_CONFIG<br>IO Pad Bias Configuration Value<br>0x0: for 1.8V Power Supply;<br>0x6: for 2.5V Power Supply;<br>0x9: for 2.8V Power Supply;<br>0xA: for 3.0V Power Supply;<br>0xD: for 3.3V Power Supply;<br>Others: Reserved.<br>Note: The configuration value must greater than or equal to the corresponding value of the voltage supplied to the IO Group. |

# Chapter 4 Memory

This section describes the A83T memory from three aspects:

- SDRAM
- NAND Flash
- SD/MMC

## 4.1. SDRAM

### 4.1.1. Overview

The SDRAM Controller (DRAMC) provides a simple, flexible, burst-optimized interface to all industy-standard SDRAM. It supports up to a 16G bits memory address space.

The DRAMC automatically handles memory management, initialization, and refresh operations. It gives the host CPU a simple command interface, hiding details of the required address, page, and burst handling procedures. All memory parameters are runtime-configurable, including timing, memory setting, SDRAM type, and Extended-Mode-Register settings. To simplify chip system integration, DDR controller works in half rate mode.

The DRAMC includes the following features:

- 32-bits data width
- Support 2 Chip Select
- Support DDR3/DDR3L/LPDDR2/LPDDR3 SDRAM
- Support Different Memory Device's Power Voltage of 1.2V 1.35V 1.5V and 1.8V
- Support clock frequency up to 800MHz(DDR3)/667MHz(LPDDR3)/533MHz(LPDDR2)
- Support Memory Capacity up to 16G bits (2G Bytes)
- Support 16 address lines and three bank address lines per channel
- Automatically generates initialization and refresh sequences
- Runtime-configurable parameters setting for application flexibility
- Support Memory Dynamic Frequency Scale
- Priority of transferring through multiple ports is programmable
- Random read or write operation is supported

## 4.2. NAND Flash

### 4.2.1. Overview

The NDFC is the NAND Flash Controller which supports all NAND flash memory available in the market. New type flash can be supported by software re-configuration.

The On-the-fly error correction code (ECC) is built-in NDFC for enhancing reliability. BCH is implemented and it can detect and correct up to 64 bits error per 512 or 1024 bytes data. The on chip ECC and parity checking circuitry of NDFC frees CPU for other tasks. The ECC function can be disabled by software.

The data can be transferred by DMA or by CPU memory-mapped IO method. The NDFC provides automatic timing control for reading or writing external Flash. The NDFC maintains the proper relativity for CLE, CE# and ALE control signal lines. Three kind of modes are supported for serial read access. The conventional serial access is mode 0 and mode 1 is for EDO type and mode 2 for extension EDO type. NDFC can monitor the status of R/B# signal line.

Block management and wear leveling management are implemented in software.

The NAND Flash Controller (NDFC) includes the following features:

- Supports all SLC/MLC/TLC flash and EF-NAND memory available in the market
- Software configure seed for randomize engine
- Software configure method for adaptability to a variety of system and memory types
- Supports 8-bit Data Bus Width
- Supports 1024, 2048, 4096, 8192, 16384 bytes size per page
- Supports Conventional and EDO serial access method for serial reading Flash
- On-the-fly BCH error correction code which correcting up to 64 bits per 512 or 1024 bytes
- Corrected Error bits number information report
- ECC automatic disable function for all 0xff data
- NDFC status information is reported by its' registers and interrupt is supported
- One Command FIFO
- External DMA is supported for transferring data
- Two 256x32-bit RAM for Pipeline Procession
- Support SDR, ONFI DDR and Toggle DDR NAND
- Support self –debug for NDFC debug

## 4.2.2.  Block Diagram

The NAND Flash Controller (NDFC) system block diagram is shown below:
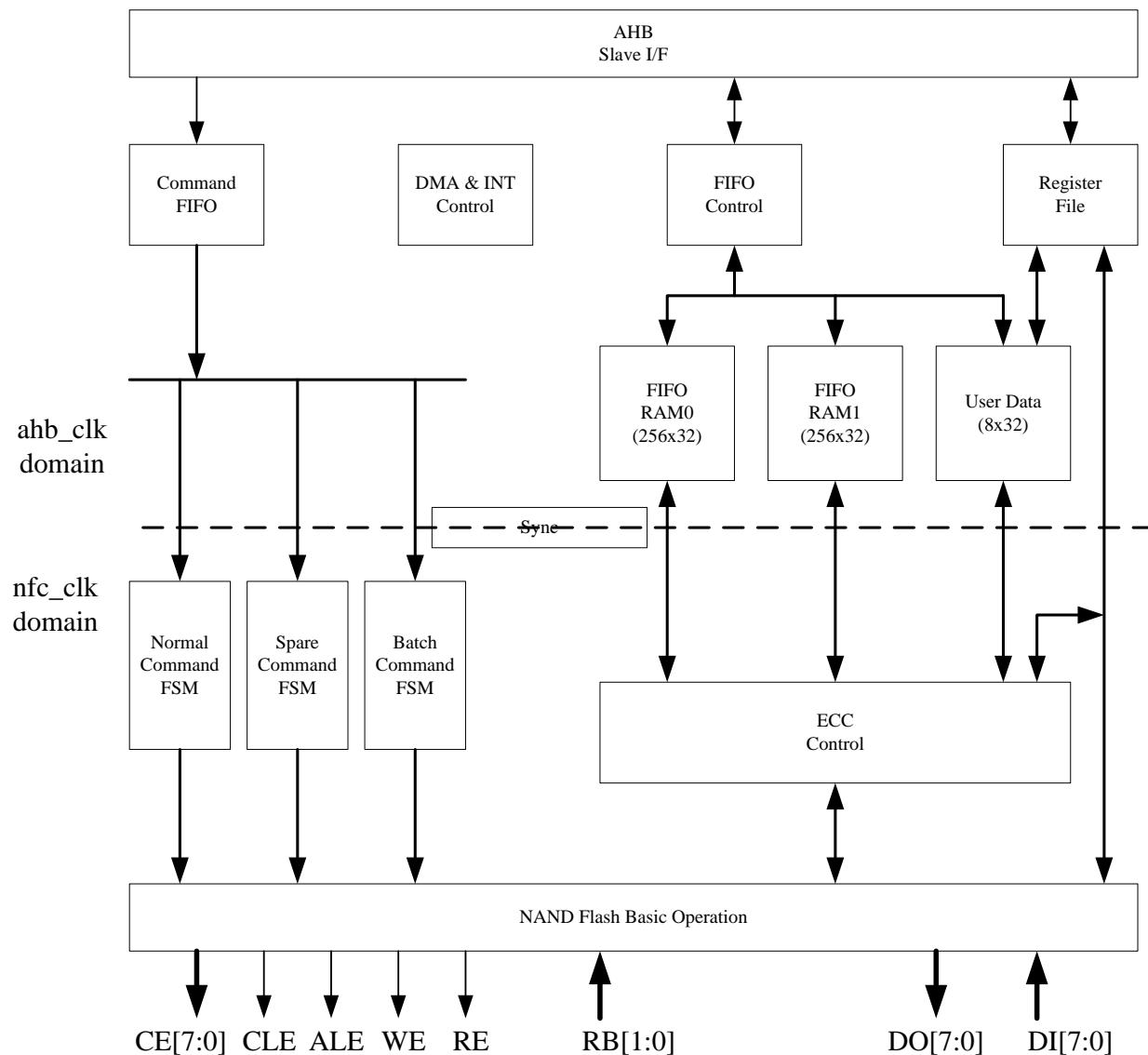


Figure 4-1. NDFC Block Diagram

## 4.2.3. NDFC Timing Diagram

Typically, there are two kinds of serial access method. One method is conventional method which fetching data at the rise edge of NDFC_RE# signal line. Another one is EDO type which fetching data at the next fall edge of NDFC_RE# signal line.
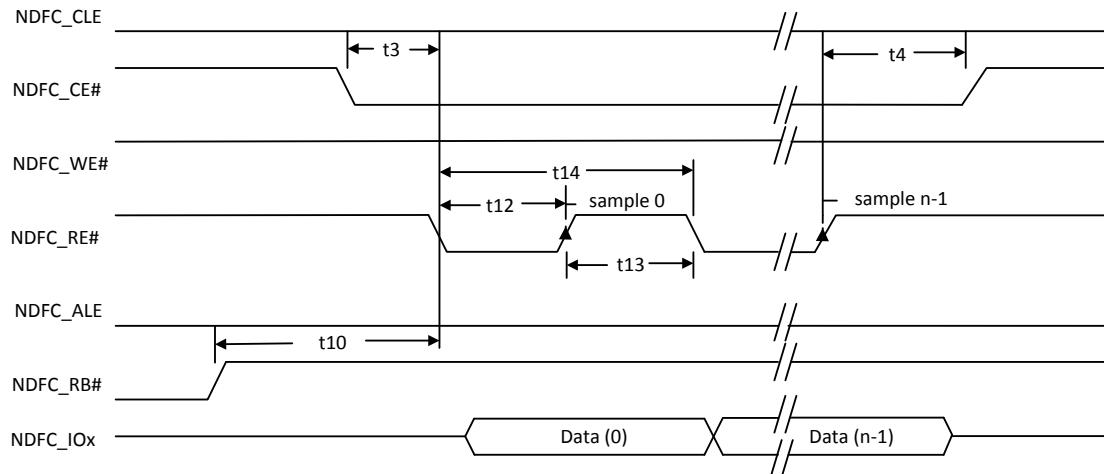


Figure 4-2. Conventional Serial Access Cycle Diagram (SAM0)
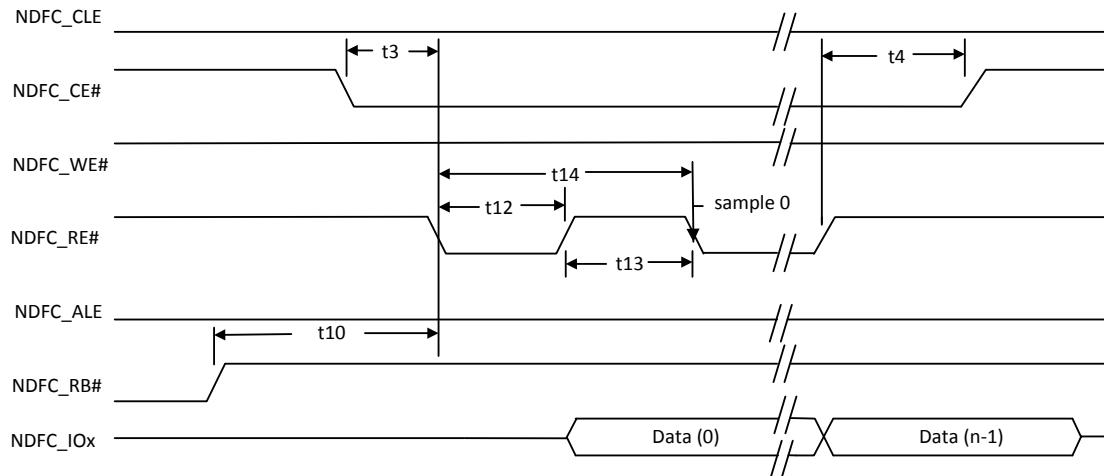


Figure 4-3. EDO type Serial Access after Read Cycle (SAM1)
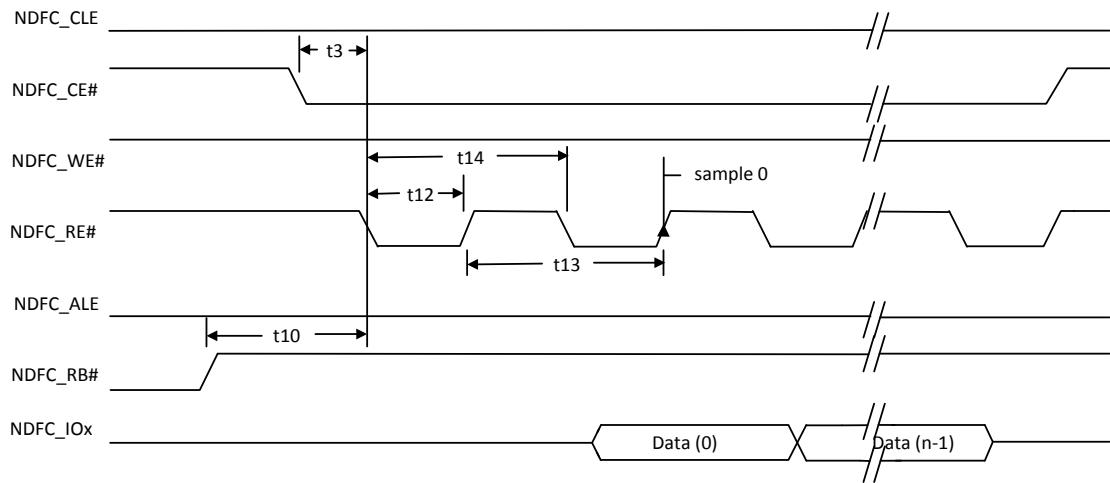
Figure 4-4. Extending EDO type Serial Access Mode (SAM2)
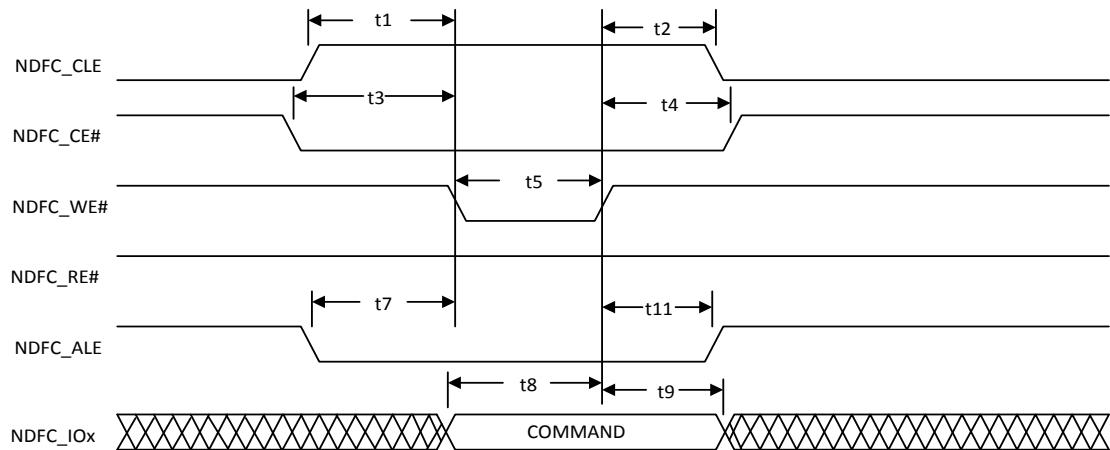


Figure 4-5. Command Latch Cycle
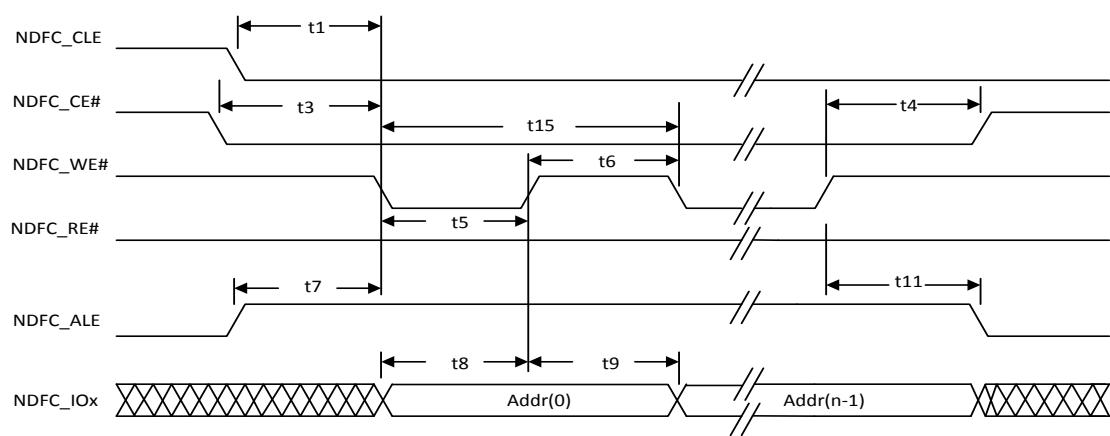


Figure 4-6. Address Latch Cycle
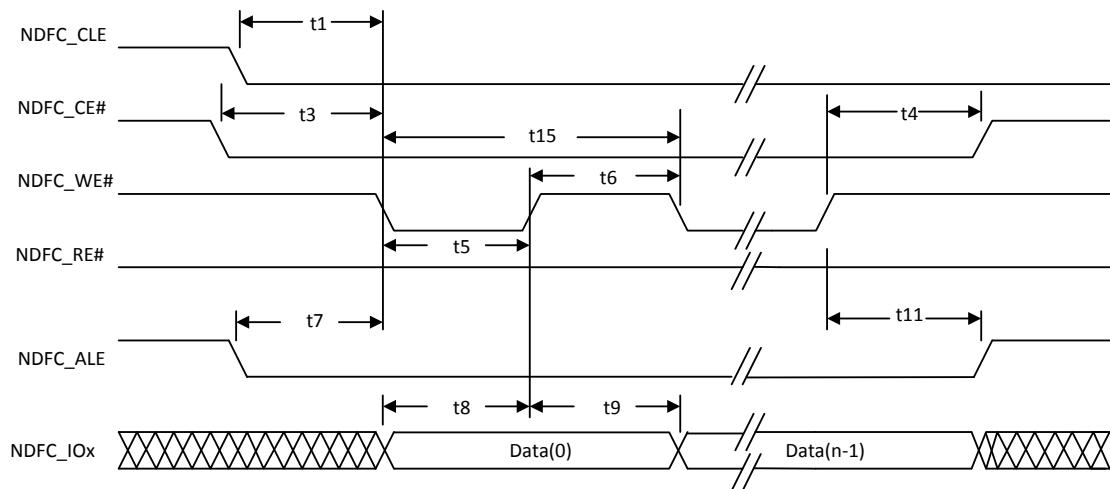
Figure 4-7. Write Data to Flash Cycle
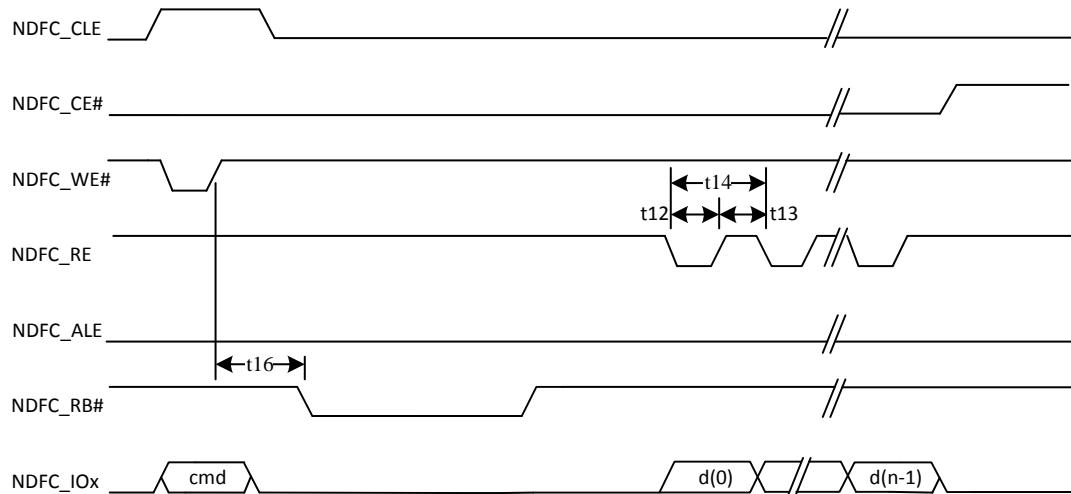


Figure 4-8. Waiting R/B# ready Diagram
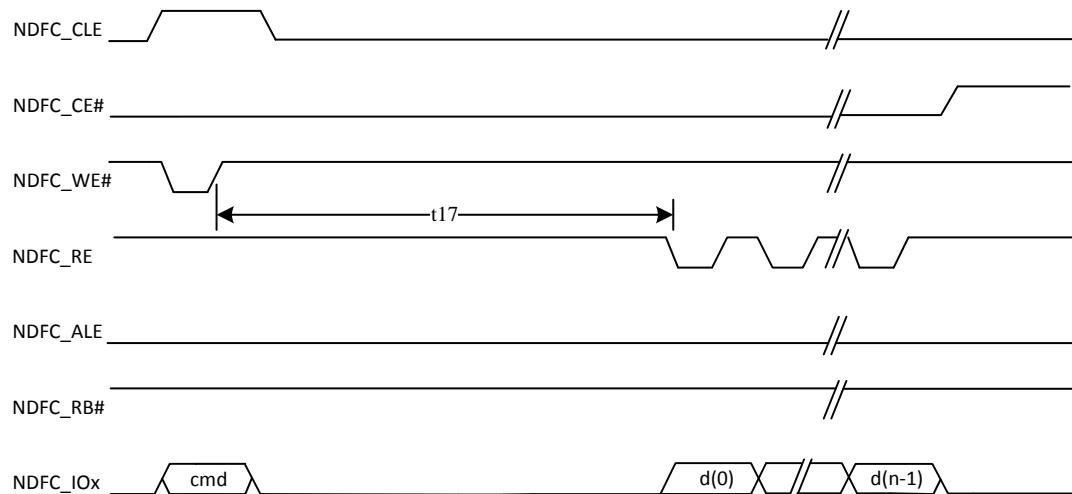
Figure 4-9. WE# high to RE# low Timing Diagram
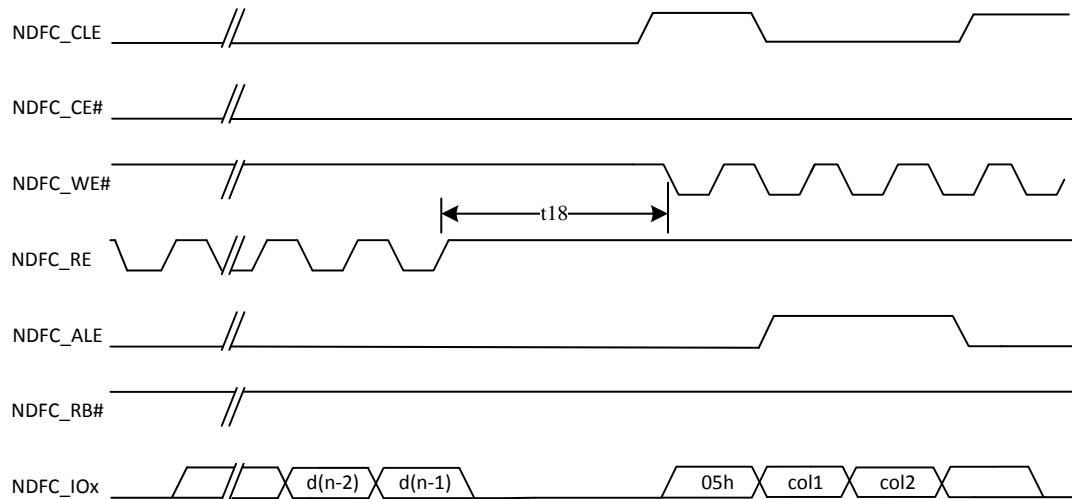


Figure 4-10. RE# high to WE# low Timing Diagram
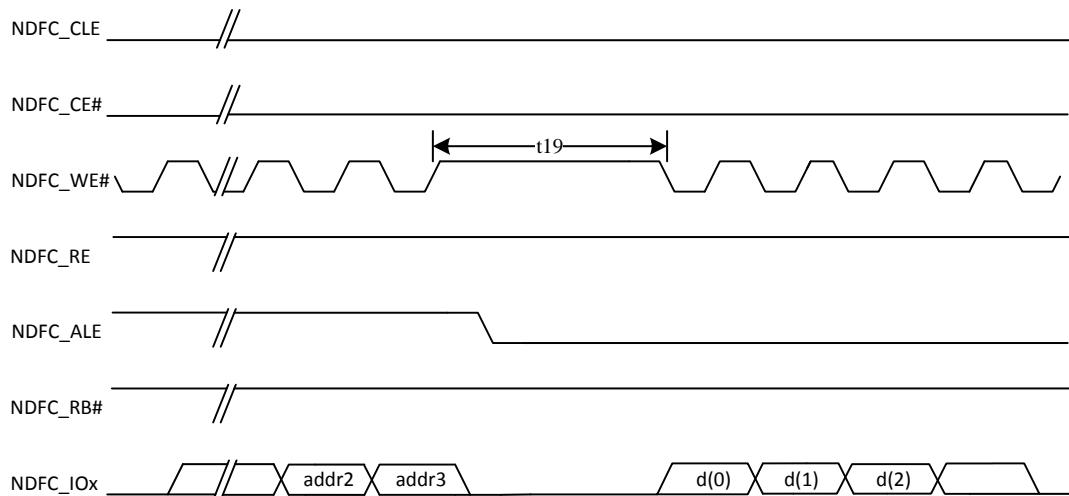
Figure 4-11. Address to Data Loading Timing Diagram

**Timing cycle list:**

| ID | Parameter | Timing | Notes |
|----|-----------|--------|-------|
| t1 | NDFC_CLE setup time | T | |
| t2 | NDFC_CLE hold time | T | |
| t3 | NDFC_CE setup time | T | |
| t4 | NDFC_CE hold time | T | |
| t5 | NDFC_WE# pulse width | T | |
| t6 | NDFC_WE# hold time | T | |
| t7 | NDFC_ALE setup time | T | |
| t8 | Data setup time | T | |
| t9 | Data hold time | T | |
| t10 | Ready to NDFC_RE# low | 3T | |
| t11 | NDFC_ALE hold time | T | |
| t12 | NDFC_RE# pulse width | T | |
| t13 | NDFC_RE# hold time | T | |
| t14 | Read cycle time | 2T | |
| t15 | Write cycle time | 2T | |
| t16 | NDFC_WE# high to R/B# busy | tWB | Specified by timing configure register (NDFC_TIMING_CFG) |
| t17 | NDFC_WE# high to NDFC_RE# low | tWHR | Specified by timing configure register (NDFC_TIMING_CFG) |
| t18 | NDFC_RE# high to NDFC_WE# low | tRHW | Specified by timing configure register (NDFC_TIMING_CFG) |
| t19 | Address to Data Loading time | tADL | Specified by timing configure register (NDFC_TIMING_CFG) |

**Notes**:

 T is the clock period duration of NDFC_CLK (x2).

## 4.2.4. NDFC Operation Guide
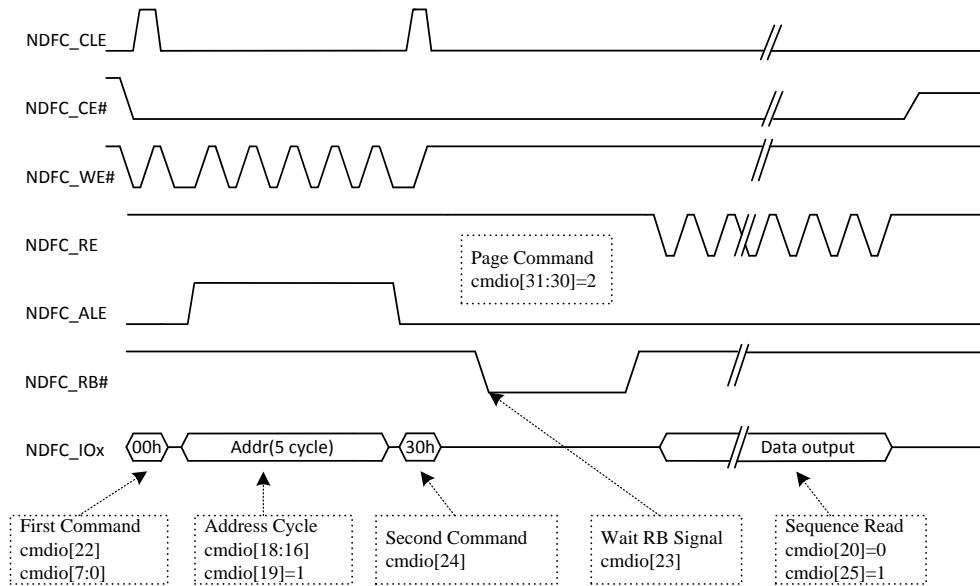


Figure 4-12. Page Read Command Diagram
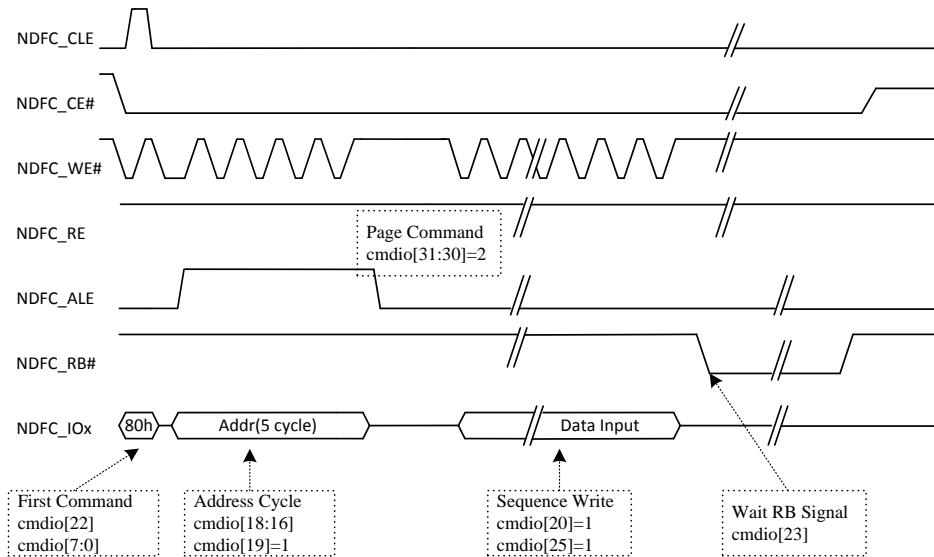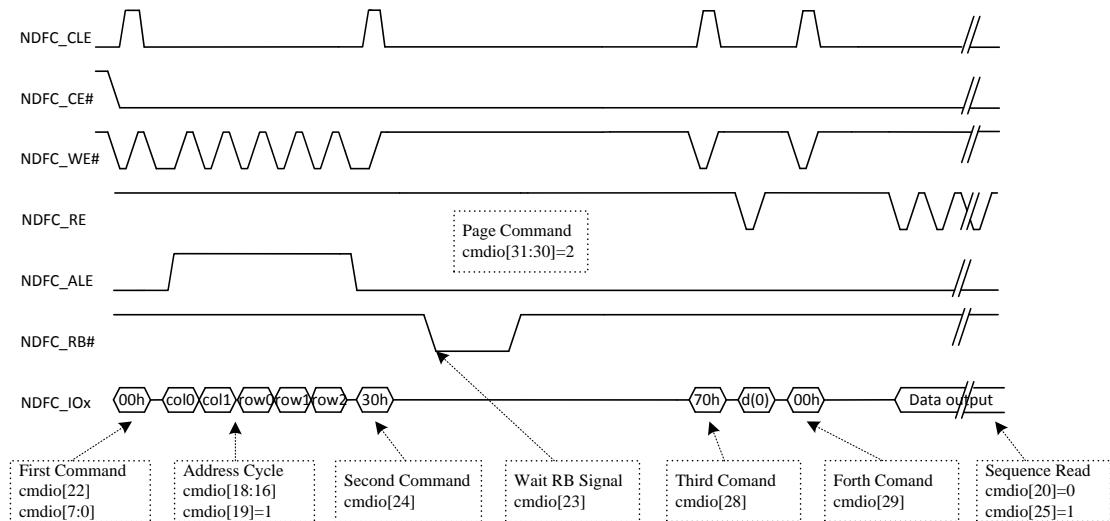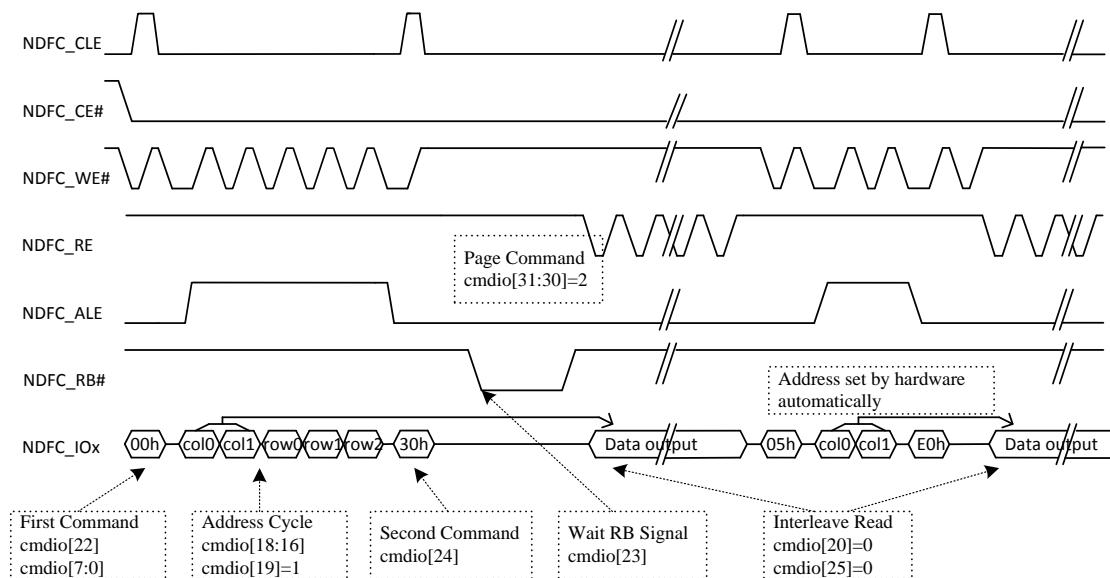


Figure 4-13. Page Program Diagram

Figure 4-14. EF-NAND Page Read Diagram



Figure 4-15. Interleave Page Read Diagram

## 4.2.5. NDFC Register List

| Module Name | Base Address |
|---|---|
| NDFC | 0x01C03000 |

| Register Name | Offset | Description |
|---|---|---|

| NDFC_CTL | 0x00 | NDFC Configure and Control Register |
|---|---|---|
| NDFC_ST | 0x04 | NDFC Status Information Register |
| NDFC_INT | 0x08 | NDFC Interrupt Control Register |
| NDFC_TIMING_CTL | 0x0C | NDFC Timing Control Register |
| NDFC_TIMING_CFG | 0x10 | NDFC Timing Configure Register |
| NDFC_ADDR_LOW | 0x14 | NDFC Low Word Address Register |
| NDFC_ADDR_HIGH | 0x18 | NDFC High Word Address Register |
| NDFC_BLOCK_NUM | 0x1C | NDFC Data Block Number Register |
| NDFC_CNT | 0x20 | NDFC Data Counter for data transfer Register |
| NDFC_CMD | 0x24 | Set up NDFC commands Register |
| NDFC_RCMD_SET | 0x28 | Read Command Set Register for vendor's NAND memory |
| NDFC_WCMD_SET | 0x2C | Write Command Set Register for vendor's NAND memory |
|  |  |  |
| NDFC_ECC_CTL | 0x34 | ECC Configure and Control Register |
| NDFC_ECC_ST | 0x38 | ECC Status and Operation information Register |
| NDFC_EFR | 0x3C | Enhanced Feature Register |
| NDFC_ERR_CNT0 | 0x40 | Corrected Error Bit Counter Register 0 |
| NDFC_ERR_CNT1 | 0x44 | Corrected Error Bit Counter Register 1 |
| NDFC_USER_DATAn | 0x50+4*n | User Data Field Register n (n from 0 to 15) |
| NDFC_EFNAND_STA | 0x90 | EFNAND Status Register |
| NDFC_SPARE_AREA | 0xA0 | Spare Area Configure Register |
| NDFC_PAT_ID | 0xA4 | Pattern ID Register |
| NDFC_RDATA_STA_CTL | 0xA8 | Read Data Status Control Register |
| NDFC_RDATA_STA_0 | 0xAC | Read Data Status Register 0 |
| NDFC_RDATA_STA_1 | 0xB0 | Read Data Status Register 1 |
| NDFC_MDMA_ADDR | 0xC0 | MBUS DMA Address Register |
| NDFC_MDMA_CNT | 0xC4 | MBUS DMA Data Counter Register |
|  |  |  |
| NDFC_IO_DATA | 0x300 | Data Input/ Output Port Address Register |
| RAM0_BASE | 0x400 | 1024 Bytes RAM0 base |
| RAM1_BASE | 0x800 | 1024 Bytes RAM1 base |

## 4.2.6.  NDFC Register Description

### 4.2.6.1.  NDFC Control Register(Default Value: 0x00000000)

| Offset: 0x00 | | | Register Name: NDFC_CTL |
|---|---|---|---|
| Bit | Read/Write | Default | Description |
| 31:28 | / | / | / |
| 27:24 | R/W | 0 | NDFC_CE_SEL |

| | | | |
|---|---|---|---|
| | | | Chip Select for 8 NAND Flash Chips<br>0 -7: NDFC Chip Select Signal 0-7 is selected<br>8-15: NDFC CS[7:0] not selected. GPIO pins can be used for CS. NDFC can support up to 16 CS. |
| 23:22 | / | / | / |
| 21 | R/W | 0 | NDFC_DDR_RM<br>DDR Repeat data mode<br>0: Lower byte<br>1: Higher byte |
| 20 | R/W | 0 | NDFC_DDR_REN<br>DDR Repeat Enable<br>0: Disable<br>1: Enable |
| 19:18 | R/W | 0 | NF_TYPE<br>NAND Flash Type<br>0x0: Normal SDR NAND<br>0x1: Reserved<br>0x2: ONFI DDR NAND<br>0x3: Toggle DDR NAND |
| 17 | R/W | 0 | NDFC_CLE_POL<br>NDFC Command Latch Enable (CLE) Signal Polarity Select<br>0: High active<br>1: Low active |
| 16 | R/W | 0 | NDFC_ALE_POL<br>NDFC Address Latch Enable (ALE) Signal Polarity Select<br>0: High active<br>1: Low active |
| 15 | R/W | 0 | NDFC_DMA_TYPE<br>0: Dedicated DMA<br>1: Normal DMA |
| 14 | R/W | 0 | NDFC_RAM_METHOD<br>Access internal RAM method<br>0: Access internal RAM by AHB bus<br>1: Access internal RAM by DMA bus |
| 13:12 | / | / | / |
| 11:8 | R/W | 0x0 | NDFC_PAGE_SIZE<br>0x0: 1024 bytes<br>0x1: 2048 bytes<br>0x2: 4096 bytes<br>0x3: 8192 bytes<br>0x4: 16384 bytes<br>Notes: The page size is for main field data. |
| 7 | / | / | / |
| 6 | R/W | 0 | NDFC_CE_ACT<br>Chip Select Signal CE# Control During NAND operation |

| | | | 0: De-active Chip Select Signal NDFC_CE# during data loading, serial access and other no operation stage for power consumption. NDFC automatic control Chip Select Signals.<br>1: Chip select signal NDFC_CE# is always active after NDFC is enabled |
|---|---|---|---|
| 5 | / | / | / |
| 4:3 | R/W | 0 | NDFC_RB_SEL<br>NDFC external R/B Signal select<br>The value 0-3 selects the external R/B signal. The same R/B signal can be used for multiple chip select flash. |
| 2 | R/W | 0 | NDFC_BUS_WIDTH<br>0: 8-bit bus<br>1: 16-bit bus |
| 1 | R/W | 0 | NDFC_RESET<br>NDFC Reset<br>Write 1 to reset NDFC and clear to 0 after reset |
| 0 | R/W | 0 | NDFC_EN<br>NDFC Enable Control<br>0: Disable NDFC<br>1: Enable NDFC |

### 4.2.6.2. NDFC Status Register(Default Value: 0x00000000)

| Offset: 0x04 | | | Register Name: NDFC_ST |
|---|---|---|---|
| Bit | Read/Write | Default | Description |
| 31:14 | / | / | / |
| 13 | R | / | NDFC_RDATA_STA_0<br>0: The number of bit 1 during current read operation is greater threshold value.<br>1: The number of bit 1 during current read operation is less than or equal to threshold value.<br>This field only is valid when NDFC_RDATA_STA_EN is 1.<br>The threshold value is configured in NDFC_RDATA_STA_TH. |
| 12 | R | / | NDFC_RDATA_STA_1<br>0: The number of bit 0 during current read operation is greater threshold value.<br>1: The number of bit 0 during current read operation is less than or equal to than threshold value.<br>This field only is valid when NDFC_RDATA_STA_EN is 1.<br>The threshold value is configured in NDFC_RDATA_STA_TH. |
| 11 | R | / | NDFC_RB_STATE3<br>NAND Flash R/B 3 Line State<br>0: NAND Flash in BUSY State<br>1: NAND Flash in READY State |
| 10 | R | / | NDFC_RB_STATE2 |

| | | | NAND Flash R/B 2 Line State |
|---|---|---|---|
| | | | 0: NAND Flash in BUSY State |
| | | | 1: NAND Flash in READY State |
| 9 | R | / | NDFC_RB_STATE1<br>NAND Flash R/B 1 Line State<br>0: NAND Flash in BUSY State<br>1: NAND Flash in READY State |
| 8 | R | / | NDFC_RB_STATE0<br>NAND Flash R/B 0 Line State<br>0: NAND Flash in BUSY State<br>1: NAND Flash in READY State |
| 7:5 | / | / | / |
| 4 | R | 0 | NDFC_STA<br>0: NDFC FSM in IDLE state<br>1: NDFC FSM in BUSY state<br>When NDFC_STA is 0, NDFC can accept new command and process command. |
| 3 | R | 0 | NDFC_CMD_FIFO_STATUS<br>0: Command FIFO not full and can receive new command<br>1: Full and waiting NDFC to process commands in FIFO<br>Since there is only one 32-bit FIFO for command. When NDFC latches one command, command FIFO is free and can accept another new command. |
| 2 | R/W | 0 | NDFC_DMA_INT_FLAG<br>When it is 1, it means that a pending DMA is completed. It will be clear after writing 1 to this bit or it will be automatically clear before FSM processing an new command. |
| 1 | R/W | 0 | NDFC_CMD_INT_FLAG<br>When it is 1, it means that NDFC has finished one Normal Command Mode or one Batch Command Work Mode. It will be clear after writing 1 to this bit or it will be automatically clear before FSM processing an new command. |
| 0 | R/W | 0 | NDFC_RB_B2R<br>When it is 1, it means that NDFC_R/B# signal is transferred from BUSY state to READY state. It will be clear after writing 1 to this bit. |

### 4.2.6.3. NDFC Interrupt and DMA Enable Register(Default Value: 0x00000000)

| Offset: 0x08 | | | Register Name: NDFC_INT |
|---|---|---|---|
| Bit | Read/Write | Default | Description |
| 31:3 | / | / | / |
| 2 | R/W | 0 | NDFC_DMA_INT_ENABLE<br>Enable or disable interrupt when a pending DMA is completed. |
| 1 | R/W | 0 | NDFC_CMD_INT_ENABLE<br>Enable or disable interrupt when NDFC has finished the procession of a single command in Normal Command Work Mode or one Batch Command Work |

| | | | |
|---|---|---|---|
| | | 0 | Mode.<br>0: Disable<br>1: Enable |
| 0 | R/W | 0 | NDFC_B2R_INT_ENABLE<br>Enable or disable interrupt when NDFC_RB# signal is transferring from BUSY state to READY state<br>0: Disable<br>1: Enable |

### 4.2.6.4. NDFC Timing Control Register(Default Value: 0x00000000)

| Offset: 0x0C | | | Register Name: NDFC_TIMING_CTL |
|---|---|---|---|
| Bit | Read/Write | Default | Description |
| 31:12 | / | / | / |
| 11:8 | R/W | 0x0 | NDFC_READ_PIPE<br><br>In SDR mode:<br>0: Normal<br>1: EDO<br>2: E-EDO<br>Other : Reserved<br><br>In DDR mode:<br>1~15 is valid.(These bits configure the number of clock when data is valid after RE#'s falling edge) |
| 7:6 | / | / | / |
| 5:0 | R/W | 0x0 | NDFC_DC_CTL<br>NDFC Delay Chain Control. (These bits are only valid in DDR data interface, and configure the relative phase between DQS and DQ[0…7] ) |

### 4.2.6.5. NDFC Timing Configure Register(Default Value: 0x00000095)

| Offset: 0x10 | | | Register Name: NDFC_TIMING_CFG |
|---|---|---|---|
| Bit | Read/Write | Default | Description |
| 31:20 | / | / | / |
| 19:18 | R/W | 0 | T_WC<br>Write Cycle Time<br>0: 1*T<br>1: 2*T<br>2: 3*T<br>3: 4*T |
| 17:16 | R/W | 0 | T_CCS |

| | | | | |
|---|---|---|---|---|
| | | | | Change Column Setup Time<br>0: 16*2T<br>1: 24*2T<br>2: 32*2T<br>3: 64*2T |
| 15:14 | R/W | | 0 | T_CLHZ<br>CLE High to Output Hi-z<br>0: 2*2T<br>1: 8*2T<br>2: 16*2T<br>3: 31*2T |
| 13:12 | R/W | | 0 | T_CS<br>CE Setup Time<br>0: 2*T<br>1: 8*T<br>2: 16*T<br>3: 31*T |
| 11 | | | | T_CDQSS<br>DQS Setup Time for data input start<br>0: 8*T<br>1: 24*T |
| 10:8 | R/W | | 0 | T_CAD<br>Command, Address, Data Delay<br>000: 4*2T<br>001: 8*2T<br>010: 12*2T<br>011: 16*2T<br>100: 24*2T<br>101: 32*2T<br>110/111: 64*2T |
| 7:6 | R/W | | 0x2 | T_RHW<br>RE# high to WE# low cycle number<br>00: 4T<br>01: 8T<br>10: 12T<br>11: 20T |
| 5:4 | R/W | | 0x1 | T_WHR<br>WE# high to RE# low cycle number<br>00: 8T<br>01: 16T<br>10: 24T<br>11: 32T |
| 3:2 | R/W | | 0x1 | T_ADL<br>Address to Data Loading cycle number<br>00: 8T |

| | | | 01: 16T |
|---|---|---|---|
| | | | 10: 24T |
| | | | 11: 32T |
| | | | T_WB |
| | | | WE# high to busy cycle number |
| | | | 00:6T |
| | | | 01: 12T |
| | | | 10: 16T |
| 1:0 | R/W | 0x1 | 11: 20T |

#### 4.2.6.6. NDFC Address Low Word Register(Default Value: 0x00000000)

| Offset: 0x14 | | | Register Name: NDFC_ADDR_LOW |
|---|---|---|---|
| Bit | Read/Write | Default | Description |
| 31:24 | R/W | 0 | ADDR_DATA4<br>NAND Flash 4th Cycle Address Data |
| 23:16 | R/W | 0 | ADDR_DATA3<br>NAND Flash 3rd Cycle Address Data |
| 15:8 | R/W | 0 | ADDR_DATA2<br>NAND Flash 2nd Cycle Address Data |
| 7:0 | R/W | 0 | ADDR_DATA1<br>NAND Flash 1st Cycle Address Data |

#### 4.2.6.7. NDFC Address High Word Register(Default Value: 0x00000000)

| Offset: 0x18 | | | Register Name: NDFC_ADDR_HIGH |
|---|---|---|---|
| Bit | Read/Write | Default | Description |
| 31:24 | R/W | 0 | ADDR_DATA8<br>NAND Flash 8th Cycle Address Data |
| 23:16 | R/W | 0 | ADDR_DATA7<br>NAND Flash 7th Cycle Address Data |
| 15:8 | R/W | 0 | ADDR_DATA6<br>NAND Flash 6th Cycle Address Data |
| 7:0 | R/W | 0 | ADDR_DATA5<br>NAND Flash 5th Cycle Address Data |

#### 4.2.6.8. NDFC Data Block Number Register(Default Value: 0x00000000)

| Offset: 0x1C | | | Register Name: NDFC_DATA_BLOCK_NUM |
|---|---|---|---|
| Bit | Read/Write | Default | Description |
| 31:6 | / | / | / |

| | | | NDFC_DATA_BLOCK_NUM |
|---|---|---|---|
| | | | DATA BLOCK Number |
| | | | It is used for batch command procession. |
| | | | 0: no data |
| | | | 1: 1 data blocks |
| | | | 2: 2 data blocks |
| | | | … |
| | | | 16: 16 data blocks |
| | | | Others: Reserved |
| 4:0 | R/W | 0 | Notes: 1 data block = 512 or 1024 bytes main field data |

### 4.2.6.9.  NDFC Data Counter Register(Default Value: 0x00000000)

| Offset: 0x20 | | | Register Name: NDFC_CNT |
|---|---|---|---|
| Bit | Read/Write | Default | Description |
| 31:10 | / | / | / |
| 9:0 | R/W | 0 | NDFC_DATA_CNT<br>Transfer Data Byte Counter<br>The length can be set from 1 byte to 1024 bytes. However, 1024 bytes is set when it is zero. |

### 4.2.6.10.  NDFC Command IO Register(Default Value: 0x00000000)

| Offset: 0x24 | | | Register Name: NDFC_CMD |
|---|---|---|---|
| Bit | Read/Write | Default | Description |
| 31:30 | R/W | 0 | NDFC_CMD_TYPE<br>00: Common Command for normal operation<br>01: Special Command for Flash Spare Field Operation<br>10: Page Command for batch process operation<br>11: Reserved |
| 29 | R/W | 0 | NDFC_SEND_FOURTH_CMD<br>0: Don't send third set command<br>1: Send it on the external memory's bus<br>Notes：It is used for EF-NAND page read. |
| 28 | R/W | 0 | NDFC_SEND_THIRD_CMD<br>0: Don't send third set command<br>1: Send it on the external memory's bus<br>Notes：It is used for EF-NAND page read. |
| 27 | R/W | 0 | NDFC_ROW_ADDR_AUTO<br>Row Address Auto Increase for Page Command<br>0: Normal operation<br>1: Row address increasing automatically |

| 26 | R/W | 0 | NDFC_DATA_METHOD<br>Data swap method when the internal RAM and system memory<br>It is only active for Common Command and Special Command.<br>0: No action<br>1: DMA transfer automatically<br>It only is active when NDFC_RAM_METHOD is 1.<br>If this bit is set to 1, NDFC should setup DRQ to fetching data before output to Flash or    NDFC should setup DRQ to sending out to system memory after fetching data from Flash.<br>If this bit is set to 0, NDFC output the data in internal RAM or do nothing after fetching data from Flash. |
|---|---|---|---|
| 25 | R/W | 0 | NDFC_SEQ<br>User data & BCH check word position. It only is active for Page Command, don't care about this bit for other two commands<br>0: Interleave Method (on page spare area)<br>1: Sequence Method (following data block) |
| 24 | R/W | 0 | NDFC_SEND_SECOND_CMD<br>0: Don't send second set command<br>1: Send it on the external memory's bus |
| 23 | R/W | 0 | NDFC_WAIT_FLAG<br>0: NDFC can transfer data regardless of the internal NDFC_RB wire<br>1: NDFC can transfer data when the internal NDFC_RB wire is READY; otherwise it can't when the internal NDFC_RB wire is BUSY. |
| 22 | R/W | 0 | NDFC_SEND_FIRST_CMD<br>0: Don't send first set command<br>1: Send it on the external memory's bus |
| 21 | R/W | 0 | NDFC_DATA_TRANS<br>0: No data transfer on external memory bus<br>1: Data transfer and direction is decided by the field NDFC_ACCESS_DIR |
| 20 | R/W | 0 | NDFC_ACCESS_DIR<br>0: Read NAND Flash<br>1: Write NAND Flash |
| 19 | R/W | 0 | NDFC_SEND_ADR<br>0: Don't send ADDRESS<br>1: Send N cycles ADDRESS, the number N is specified by NDFC_ADR_NUM field |
| 18:16 | R/W | 0 | NDFC_ADR_NUM<br>Address Cycles' Number<br>000: 1 cycle address field<br>001: 2 cycles address field<br>010: 3 cycles address field<br>011: 4 cycles address field<br>100: 5 cycles address field<br>101: 6 cycles address field<br>110: 7 cycles address field |

| | | | 111: 8 cycles address field |
|---|---|---|---|
| 15:8 | R/W | 0 | NDFC_CMD_HIGH_BYTE<br>NDFC Command high byte data<br>If 8-bit command is supported, the high byte should be zero for 16-bit bus width NAND Flash. For 8-bit bus width NAND Flash, high byte command is discarded. |
| 7:0 | R/W | 0 | NDFC_CMD_LOW_BYTE<br>NDFC Command low byte data<br>This command will be sent to external Flash by NDFC. |

### 4.2.6.11. NDFC Command Set Register 0(Default Value: 0x00E00530)

| Offset: 0x28 | | | Register Name: NDFC_CMD_SET0 |
|---|---|---|---|
| Bit | Read/Write | Default | Description |
| 31:24 | / | / | / |
| 23:16 | R/W | 0xE0 | NDFC_RANDOM_READ_CMD1<br>Used for Batch Read Operation |
| 15:8 | R/W | 0x05 | NDFC_RANDOM_READ_CMD0<br>Used for Batch Read Operation |
| 7:0 | R/W | 0x30 | NDFC_READ_CMD<br>Used for Batch Read Operation |

### 4.2.6.12. NDFC Command Set Register 1(Default Value: 0x70008510)

| Offset: 0x2C | | | Register Name: NDFC_CMD_SET1 |
|---|---|---|---|
| Bit | Read/Write | Default | Description |
| 31:16 | R/W | 0x70 | NDFC_READ_CMD0<br>Used for EF-NAND Page Read operation |
| 23:16 | R/W | 0x00 | NDFC_READ_CMD1<br>Used for EF-NAND Page Read operation |
| 15:8 | R/W | 0x85 | NDFC_RANDOM_WRITE_CMD<br>Used for Batch Write Operation |
| 7:0 | R/W | 0x10 | NDFC_PROGRAM_CMD<br>Used for Batch Write Operation |

### 4.2.6.13. NDFC IO Data Register(Default Value: 0x00000000)

| Offset: 0x30 | | | Register Name: NDFC_IO_DATA |
|---|---|---|---|
| Bit | Read/Write | Default | Description |
| 31:0 | R/W | 0 | NDFC_IO_DATA<br>Read/ Write data into internal RAM |

| | | | Access unit is 32-bit. |

### 4.2.6.14. NDFC ECC Control Register(Default Value: 0x4a800008)

| Offset: 0x34 | | | Register Name: NDFC_ECC_CTL |
|---|---|---|---|
| Bit | Read/Write | Default | Description |
| 31 | / | / | / |
| 30:16 | R/W | 0x4a80 | NDFC_RANDOM_SEED<br>The seed value for randomize engine. It is only active when NDFC_RANDOM_EN is set to '1'. |
| 15:12 | R/W | 0 | NDFC_ECC_MODE<br>0x0: BCH-16 for one ECC Data Block<br>0x1: BCH-24 for one ECC Data Block<br>0x2 : BCH-28 for one ECC Data Block<br>0x3 : BCH-32 for one ECC Data Block<br>0x4 : BCH-40 for one ECC Data Block<br>0x5 : BCH-48 for one ECC Data Block<br>0x6 : BCH-56 for one ECC Data Block<br>0x7 : BCH-60 for one ECC Data Block<br>0x8 : BCH-64 for one ECC Data Block<br>Others: Reserved |
| 11 | R/W | 0 | NDFC_RANDOM_SIZE<br>0: ECC block size<br>1: Page size |
| 10 | R/W | 0 | NDFC_RANDOM_DIRECTION<br>0: LSB first<br>1: MSB first |
| 9 | R/W | 0 | NDFC_RANDOM_EN<br>0: Disable Data Randomize<br>1: Enable Data Randomize |
| 8:6 | / | / | / |
| 5 | R/W | 0 | NDFC_ECC_BLOCK_SIZE<br>0: 1024 bytes of one ECC data block<br>1: 512 bytes of one ECC data block |
| 4 | R/W | 0 | NDFC_ECC_EXCEPTION<br>0: Normal ECC<br>1: For ECC, there is an exception. If all data is 0xff or 0x00 for the block. When reading this page, ECC assumes that it is right. For this case, no error information is reported.<br>Notes: It only is active when ECC is ON |
| 3 | R/W | 1 | NDFC_ECC_PIPELINE<br>Pipeline function enable or disable for batch command<br>0: Error Correction function no pipeline with next block operation<br>1: Error Correction pipeline |

| 2:1 | / | / | / |
|---|---|---|---|
| 0 | R/W | 0 | NDFC_ECC_EN<br>0: ECC is OFF<br>1: ECC is ON |

### 4.2.6.15. NDFC ECC Status Register(Default Value: 0x00000000)

| Offset: 0x38 | | | Register Name: NDFC_ECC_ST |
|---|---|---|---|
| Bit | Read/Write | Default | Description |
| 31:16 | R | 0 | NDFC_PAT_FOUND<br>Special pattern (all 0x00 or all x0ff) Found Flag for 16 Data Blocks<br>0: No Found<br>1: Special pattern is found<br>When this field is '1', this means that the special data is found for reading external NAND flash. The register of NDFC_PAT_ID would indicates which pattern is found. |
| 15:0 | R | 0 | NDFC_ECC_ERR<br>Error information bit of 16 Data Blocks<br>0: ECC can correct these error bits or there is no error bit<br>1: Error bits number beyond of ECC correction capability and can't correct them<br>Notes: The LSB of this register is corresponding the 1st ECC data block. 1 ECC Data Block = 512 or 1024 bytes. |

### 4.2.6.16. NDFC Enhanced Feature Register(Default Value: 0x00000000)

| Offset: 0x3C | | | Register Name: NDFC_EFR |
|---|---|---|---|
| Bit | Read/Write | Default | Description |
| 31:9 | R/W | 0 | / |
| 8 | R/W | 0 | NDFC_WP_CTRL<br>NAND Flash Write Protect Control Bit<br>0: Write Protect is active<br>1: Write Protect is not active<br>Notes: When this bit is '0', WP signal line is low level and external NAND flash is on protected state. |
| 7 | / | / | / |
| 6:0 | R/W | 0 | NDFC_ECC_DEBUG<br>For the purpose of debugging ECC engine, special bits error are inserted before writing external Flash Memory.<br>0: No error is inserted (ECC Normal Operation)<br>n: N bits error are inserted |

**4.2.6.17. NDFC Error Counter Register 0(Default Value: 0x00000000)**

| Offset: 0x40 | | | Register Name: NDFC_ERR_CNT0 |
|---|---|---|---|
| Bit | Read/Write | Default | Description |
| [8i+7:8i]<br>(i=0~3) | R | 0 | ECC_COR_NUM<br>ECC Corrected Bits Number for ECC Data Block[n] (n from 0 to 3)<br>0: No corrected bits<br>1: 1 corrected bit<br>2: 2 corrected bits<br>…<br>64: 64 corrected bits<br>Others: Reserved<br>Notes: 1 ECC Data Block = 512 or 1024 bytes |

**4.2.6.18. NDFC Error Counter Register 1(Default Value: 0x00000000)**

| Offset: 0x44 | | | Register Name: NDFC_ERR_CNT1 |
|---|---|---|---|
| Bit | Read/Write | Default | Description |
| [8i+7:8i]<br>(i=0~3) | R | 0 | ECC_COR_NUM<br>ECC Corrected Bits Number for ECC Data Block[n] (n from 4 to 7)<br>0: No corrected bits<br>1: 1 corrected bit<br>2: 2 corrected bits<br>…<br>64: 64 corrected bits<br>Others: Reserved<br>Notes: 1 ECC Data Block = 512 or 1024 bytes |

**4.2.6.19. NDFC Error Counter Register 2(Default Value: 0x00000000)**

| Offset: 0x48 | | | Register Name: NDFC_ERR_CNT2 |
|---|---|---|---|
| Bit | Read/Write | Default | Description |
| [8i+7:8i]<br>(i=0~3) | R | 0 | ECC_COR_NUM<br>ECC Corrected Bits Number for ECC Data Block[n] (n from 8 to 11)<br>0: No corrected bits<br>1: 1 corrected bit<br>2: 2 corrected bits<br>…<br>64: 64 corrected bits<br>Others: Reserved<br>Notes: 1 ECC Data Block = 512 or 1024 bytes |

#### 4.2.6.20. NDFC Error Counter Register 3(Default Value: 0x00000000)

| Offset: 0x4C | | | Register Name: NDFC_ERR_CNT3 |
|---|---|---|---|
| Bit | Read/Write | Default | Description |
| [8i+7:8i]<br><br>(i=0~3) | R | 0 | ECC_COR_NUM<br>ECC Corrected Bits Number for ECC Data Block[n] (n from 12 to 15)<br>0: No corrected bits<br>1: 1 corrected bit<br>2: 2 corrected bits<br>…<br>64: 64 corrected bits<br>Others: Reserved<br>Notes: 1 ECC Data Block = 512 or 1024 bytes |

#### 4.2.6.21. NDFC User Data Register [n]( Default Value: 0xffffffff)

| Offset: 0x50 + 0x4*n | | | Register Name: NDFC_USER_DATAn |
|---|---|---|---|
| Bit | Read/Write | Default | Description |
| 31:0 | R/W | 0xffffffff | USER_DATA<br>User Data for ECC Data Block[n] (n from 0 to 15)<br>Notes: 1 ECC Data Block = 512 or 1024 bytes |

**Notes:** n from 0 to 15

#### 4.2.6.22. NDFC EFNAND STATUS Register(Default Value: 0x00000000)

| Offset: 0x90 | | | Register Name: NDFC_EFNAND_STATUS |
|---|---|---|---|
| Bit | Read/Write | Default | Description |
| 31:8 | / | / | / |
| 7:0 | R | 0x0 | EF_NAND_STATUS<br>The Status Value for EF-NAND Page Read operation |

#### 4.2.6.23. NDFC Spare Area Register(Default Value: 0x00000400)

| Offset: 0xA0 | | | Register Name: NDFC_SPARE_AREA |
|---|---|---|---|
| Bit | Read/Write | Default | Description |
| 31:16 | / | / | / |
| 15:0 | R/W | 0x400 | NDFC_SPARE_ADR<br>This value indicates the spare area first byte address for NDFC interleave page operation. |

**4.2.6.24.  NDFC Pattern ID Register(Default Value: 0x00000000)**

| Offset: 0xA4 | | | Register Name: NDFC_PAT_ID |
|---|---|---|---|
| Bit | Read/Write | Default | Description |
| [2i+1:2i] (i=0~15) | R | 0 | PAT_ID<br>Special Pattern ID for 16 ECC data block<br>0: All 0x00 is found<br>1: All 0xFF is found<br>Others: Reserved |

**4.2.6.25.  NDFC Read Data Status Control Register(Default Value: 0x01000000)**

| Offset: 0xA8 | | | Register Name: NDFC_RDATA_STA_CTL |
|---|---|---|---|
| Bit | Read/Write | Default | Description |
| 31:25 | / | / | / |
| 24 | R/W | 1 | NDFC_RDATA_STA_EN<br>0: Disable to count the number of bit 1 and bit 0 during current read operation;<br>1: Enable to count the number of bit 1 and bit 0 during current read operation;<br>The number of bit 1 and bit 0 during current read operation can be used to check whether a page is blank or bad. |
| 23:18 | / | / | / |
| 17:0 | R/W | 0 | NDFC_RDATA_STA_TH<br>The threshold value to generate data status.<br>If the number of bit 1 during current read operation is less than or equal to threshold value, the bit 13 of NDFC_ST register will be set.<br>If the number of bit 0 during current read operation is less than or equal to threshold value, the bit 12 of NDFC_ST register will be set. |

**4.2.6.26.  NDFC Read Data Status Register 0(Default Value: 0x00000000)**

| Offset: 0xAC | | | Register Name: NDFC_RDATA_STA_0 |
|---|---|---|---|
| Bit | Read/Write | Default | Description |
| 31:0 | R | 0 | BIT_CNT_1<br>The number of input bit 1 during current command. It will be cleared automatically when next command is executed. |

**4.2.6.27.  NDFC Read Data Status Register 1(Default Value: 0x00000000)**

| Offset: 0xB0 | | | Register Name: NDFC_RDATA_STA_1 |
|---|---|---|---|
| Bit | Read/Write | Default | Description |
| 31:0 | R | 0 | BIT_CNT_0 <br> The number of input bit 0 during current command. It will be cleared automatically when next command is executed. |

**4.2.6.28.  NDFC MBUS DMA Address Register(Default Value: 0x00000000)**

| Offset: 0xC0 | | | Register Name: NDFC_MDMA_ADDR |
|---|---|---|---|
| Bit | Read/Write | Default | Description |
| 31:0 | R/W | 0 | MDMA_ADDR <br> MBUS DMA address |

**4.2.6.29.  NDFC MBUS DMA Byte Counter Register(Default Value: 0x00000000)**

| Offset: 0xC4 | | | Register Name: NDFC_MDMA_CNT |
|---|---|---|---|
| Bit | Read/Write | Default | Description |
| 14:0 | R/W | 0 | MDMA_CNT <br> MBUS DMA data counter |

## 4.3.  SD/MMC
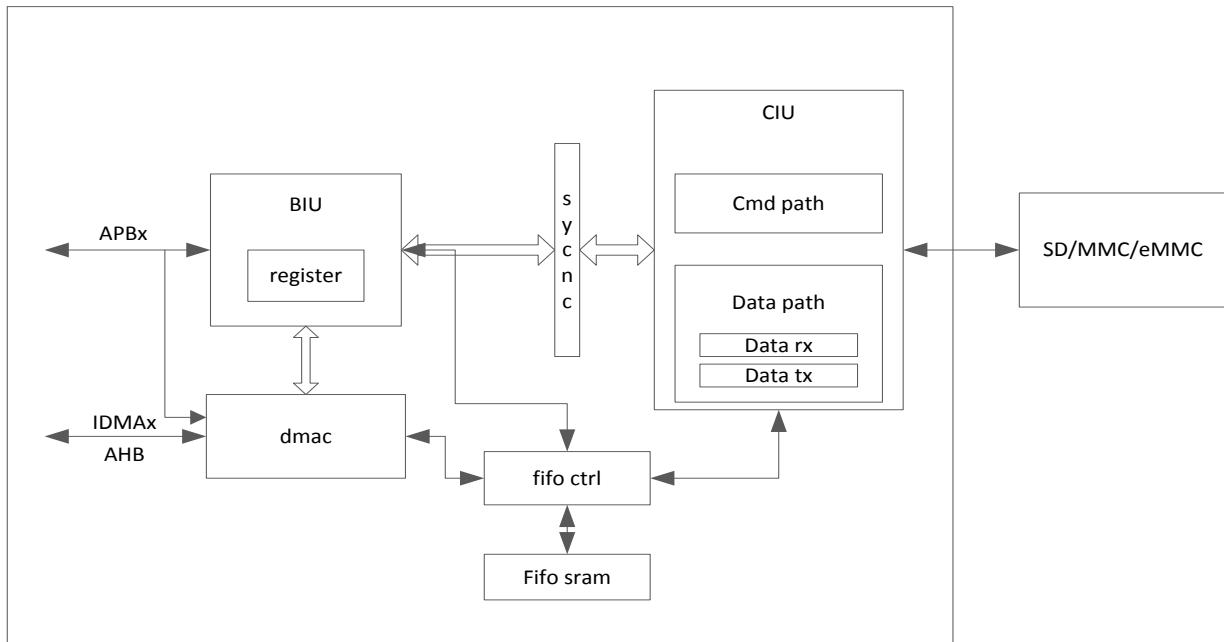
### 4.3.1.  Overview

The SD/MMC controller can be configured either as a Secure Digital Multimedia Card controller, which simultaneously supports Secure Digital I/O (SDIO), Multimedia Cards (MMC), and eMMC Card.

The SD/MMC controller includes the following features:

- Supports Secure Digital memory protocol commands (up to SD2.0)
- Supports Secure Digital I/O protocol commands
- Supports Multimedia Card protocol commands (up to eMMC4.41)
- Supports eMMC boot operation
- Supports one SD (Verson1.0 to 2.0) or MMC (Verson3.3 to eMMC4.41)
- Supports hardware CRC generation and error detection
- Supports host pull-up control
- Supports SDIO interrupts in 1-bit and 4-bit modes
- Supports SDIO suspend and resume operation
- Supports SDIO read wait
- Supports block size of 1 to 65535 bytes
- Supports descriptor-based internal DMA controller

### 4.3.2.  Block Diagram

(BIU：Bus Interface Unit; CIU：Card Interface Unit)

Figure 4-16. SD/MMC Controller Block Diagram

## 4.3.3.  SD/MMC Controller Timing Diagram

Please refer to relative specifications:

- Physical Layer Specification Ver2.00
- SDIO Specification Ver2.00
- Multimedia Cards (MMC – version 4.2)
- JEDEC Standard – JESD84-44, EMBEDDED MULTI-MEDIA CARD (e•MMC)

## 4.3.4.  SD/MMC Controller Special Requirement

### 4.3.4.1.  SD/MMC Pin List

| Port Name | Width | Direction | Description |
|-----------|-------|-----------|-------------|
| SD_CCLK | 1 | OUT | Clock output for SD/SDIO/MMC card |

| SD_CCMD | 1 | IN/OUT | CMD line |
| SD_CDATA | 1/4/8 | IN/OUT | Data line |
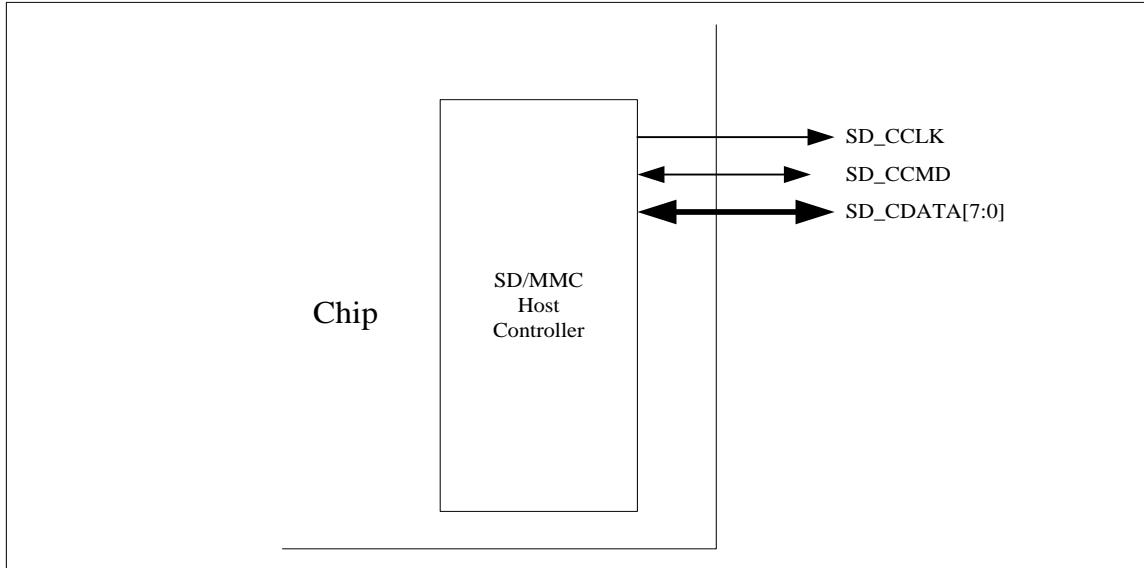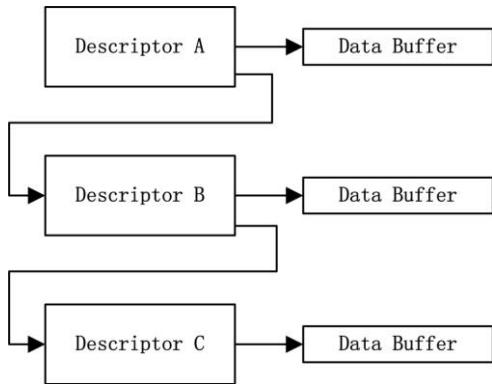


Figure 4-17. SD/MMC Pin Diagram

## 4.3.5. Internal DMA Controller Description
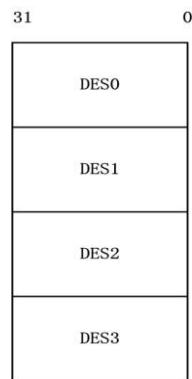
SD/MMC controller has an internal DMA controller (IDMAC) to transfer data between host memory and SDMMC port. With a descriptor, IDMAC can efficiently move data from source to destination by automatically loading next DMA transfer arguments, which need less CPU intervention. Before transfer data in IDMAC, host driver should construct a descriptor list, configure arguments of every DMA transfer, then launch the descriptor and start the DMA. IDMAC has an interrupt controller, when enabled, it can interrupt the HOST CPU in situations such as data transmission completed or some errors happened.

### 4.3.5.1. IDMAC Descriptor Structure

The IDMAC uses a descriptor with a chain structure, and each descriptor points to a unique buffer and the next descriptor.

This figure illustrates the internal formats of a descriptor. The descriptor addresses must be aligned to the bus width used for 32-bit buses. Each descriptor contains 16 bytes of control and status information.



DES0 is a notation used to denote the [31:0] bits, DES1 to denote [63:32] bits, DES2 to denote [95:64]bits, and DES3 to denote [127:96]bits in a descriptor.

### 4.3.5.2. DES0 definition

| Bits | Name | Descriptor |
|------|------|------------|
| 31 | HOLD | DES_OWN_FLAG<br>When set, this bit indicates that the descriptor is owned by the IDMAC. When this bit is reset, it indicates that the descriptor is owned by the host. This bit is cleared when transfer is over. |
| 30 | ERROR | ERR_FLAG<br>When some error happened in transfer, this bit will be set. |
| 29:5 | / | / |
| 4 | Chain Flag | CHAIM_MOD<br>When set, this bit indicates that the second address in descriptor is the next descriptor address. Must be set 1. |
| 3 | First DES Flag | FIRST_FLAG<br>When set, this bit indicates that this descriptor contains the first buffer |

| | | of data. Must be set to 1 in first DES. |
|---|---|---|
| 2 | Last DES Flag | LAST_FLAG<br>When set, this bit indicates that the buffers pointed to by this descriptor are the last data buffer |
| 1 | Disable Interrupt on completion | CUR_TXRX_OVER_INT_DIS<br>When set, this bit will prevent the setting of the TX/RX interrupt bit of the IDMAC status register for data that ends in the buffer pointed to by this descriptor |
| 0 | / | / |

### 4.3.5.3. DES1 definition

| Bits | Name | Descriptor |
|---|---|---|
| 31:13 | / | / |
| 15:0 | Buffer size | BUFF_SIZE<br>These bits indicate the data buffer byte size, which must be a multiple of 4 bytes. If this filed is 0, the DMA ignores this buffer and proceeds to the next descriptor. |

### 4.3.5.4. DES2 definition

| Bits | Name | Descriptor |
|---|---|---|
| 31:0 | Buffer address pointer | BUFF_ADDR<br>These bits indicate the physical address of data buffer. The IDMAC ignores DES2[1:0], corresponding to the bus width of 32. |

### 4.3.5.5. DES3 definition

| Bits | Name | Descriptor |
|---|---|---|
| 31:0 | Next descriptor address | NEXT_DESP_ADDR<br>These bits indicate the pointer to the physical memory where the next descriptor is present. |

## 4.3.6. SD/MMC Register List

| Module Name | Base Address |
|---|---|
| SDC0 | 0x01C0F000 |

| SDC1 | 0x01C10000 |
|------|------------|
| SDC2 | 0x01C11000 |

| Register Name | Offset | Description |
|---------------|--------|-------------|
| SD_GCTL | 0x00 | Control register |
| SD_CKCR | 0x04 | Clock Control register |
| SD_TMOR | 0x08 | Time out register |
| SD_BWDR | 0x0C | Bus Width register |
| SD_BKSR | 0x10 | Block size register |
| SD_BYCR | 0x14 | Byte count register |
| SD_CMDR | 0x18 | Command register |
| SD_CAGR | 0x1c | Command argument register |
| SD_RESP0 | 0x20 | Response 0 register |
| SD_RESP1 | 0x24 | Response 1 register |
| SD_RESP2 | 0x28 | Response 2 register |
| SD_RESP3 | 0x2C | Response 3 register |
| SD_IMKR | 0x30 | Interrupt mask register |
| SD_MISR | 0x34 | Masked interrupt status register |
| SD_RISR | 0x38 | Raw interrupt status register |
| SD_STAR | 0x3C | Status register |
| SD_FWLR | 0x40 | FIFO Water Level register |
| SD_FUNS | 0x44 | FIFO Function Select register |
| SD_A12A | 0x58 | Auto command 12 argument |
| SD_NTSR | 0x5c | SD NewTiming Set Register |
| SD_SDBG | 0x60 | SD NewTiming Set Debug Register |
| SD_HWRST | 0x78 | Hardware Reset Register |
| SD_DMAC | 0x80 | BUS Mode Control |
| SD_DLBA | 0x84 | Descriptor List Base Address |
| SD_IDST | 0x88 | DMAC Status |
| SD_IDIE | 0x8c | DMAC Interrupt Enable |
| SD_THLDC | 0x100 | Card Threshold Control register |
| SD_DSBD | 0x10c | eMMC4.5 DDR Start Bit Detection Control |
| SD_FIFO | 0x200 | Read/ Write FIFO |

## 4.3.7.   SD/MMC Register Description

### 4.3.7.1.  SD Global Control Register(Default Value: 0x00000300)

| Offset: 0x0000 | Register Name: **SD_CTRL** |
|----------------|----------------------------|

| Bit | R/W | Default/Hex | Description |
|---|---|---|---|
| 31 | R/W | 0 | FIFO_AC_MOD<br>FIFO Access Mode<br>1-AHB bus<br>0-DMA bus |
| 30:11 | - | - | / |
| 10 | R/W | 0 | DDR_MOD_SEL<br>DDR Mode Select<br>0 – SDR mode<br>1 – DDR mode |
| 9 | - | - | reserved |
| 8 | R/W | 1 | CD_DBC_ENB<br>Card Detect (Data[3] status) De-bounce Enable<br>0 - disable de-bounce<br>1 – enable de-bounce |
| 7:6 | - | - | / |
| 5 | R/W | 0 | DMA_ENB<br>DMA Global Enable<br>0 – Disable DMA to transfer data, using AHB bus<br>1 – Enable DMA to transfer data |
| 4 | R/W | 0 | INT_ENB<br>Global Interrupt Enable<br>0 – Disable interrupts<br>1 – Enable interrupts |
| 3 | - | - | / |
| 2 | R/W | 0 | DMA_RST<br>DMA Reset |
| 1 | R/W | 0 | FIFO_RST<br>FIFO Reset<br>0 – No change<br>1 – Reset FIFO<br>*This bit is auto-cleared after completion of reset operation.* |
| 0 | R/W | 0 | SOFT_RST<br>Software Reset<br>0 – No change<br>1 – Reset SD/MMC controller<br>*This bit is auto-cleared after completion of reset operation.* |

### 4.3.7.2. SD Clock Control Register(Default Value: 0x00000000)

| Offset: 0x0004 | | | Register Name: **SD_CLKDIV** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0 | MASK_DATA0<br>0 - Do not mask data0 when updata clock ; |

| | | | 1 - Mask data0 when updata clock; |
| | | | Default value : 0; |
| 30:18 | / | / | / |
| 17 | R/W | 0 | CCLK_CTRL |
| | | | Card Clock Output Control |
| | | | 0 – Card clock always on |
| | | | 1 – Turn off card clock when FSM in IDLE state |
| 16 | R/W | 0 | CCLK_ENB |
| | | | Card Clock Enable |
| | | | 0 – Card Clock off |
| | | | 1 – Card Clock on |
| 15:8 | / | / | / |
| 7:0 | R/W | 0 | CCLK_DIV |
| | | | Card clock divider |
| | | | n – Source clock is divided by 2*n.(n=0~255) |

### 4.3.7.3. SD Timeout Register (Default Value: 0xFFFFFF40)

| Offset: 0x0008 | | | Register Name: **SD_TMOUT** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:8 | R/W | 0xffffff | DTO_LMT |
| | | | Data Timeout Limit |
| 7:0 | R/W | 0x40 | RTO_LMT |
| | | | Response Timeout Limit |

### 4.3.7.4. SD Bus Width Register (Default Value: 0x00000000)

| Offset: 0x000c | | | Register Name: **SD_CTYPE** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:2 | / | / | / |
| 1:0 | R/W | 0 | CARD_WID |
| | | | Card width |
| | | | 2'b00 – 1-bit width |
| | | | 2'b01 – 4-bit width |
| | | | 2'b1x – 8-bit width |

### 4.3.7.5. SD Block Size Register (Default Value: 0x00000200)

| Offset: 0x0010 | | | Register Name: **SD_BLKSIZ** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:16 | / | / | / |

| 15:0 | R/W | 0x200 | BLK_SZ<br>Block size |

#### 4.3.7.6. SD Block Count Register (Default Value: 0x00000200)

| Offset: 0x0014 | | | Register Name: **SD_BYTCNT** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R/W | 0x200 | BYTE_CNT<br>Byte counter<br>Number of bytes to be transferred; should be integer multiple of Block Size for block transfers. |

#### 4.3.7.7. SD Command Register (Default Value: 0x00000000)

| Offset: 0x0018 | | | Register Name: **SD_CMD** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0 | CMD_LOAD<br>Start Command.<br>This bit is auto cleared when current command is sent. If there is no any response error happened, a command complete interrupt bit (CMD_OVER) will be set in interrupt register. You should not write any other command before this bit is cleared, or a command busy interrupt bit (CMD_BUSY) will be set in interrupt register. |
| 30 | / | / | / |
| 29 | R/W | 0 | Use Hold Register<br>0 - CMD and DATA sent to card bypassing HOLD Register<br>1 - CMD and DATA sent to card through the HOLD Register |
| 28 | R/W | 0 | VOL_SW<br>Voltage Switch<br>0 – normal command<br>1 – Voltage switch command, set for CMD11 only |
| 27 | R/W | 0 | BOOT_ABT<br>Boot Abort<br>Setting this bit will terminate the boot operation. |
| 26 | R/W | 0 | EXP_BOOT_ACK<br>Expect Boot Acknowledge.<br>When Software sets this bit along in mandatory boot operation, controller expects a boot acknowledge start pattern of 0-1-0 from the selected card. |
| 25:24 | R/W | 0 | BOOT_MOD<br>Boot Mode<br>2'b00 – normal command<br>2'b01 - Mandatory Boot operation |

| | | | |
|---|---|---|---|
| | | | 2'b10 - Alternate Boot operation |
| | | | 2'b11 - reserved |
| 23 | R/W | 0 | CCS_EXP<br>ccs_expected<br>0 – Interrupts are not enabled in CE-ATA device (nIEN = 1 in ATA control register), or command does not expect CCS from device<br>1 – Interrupts are enabled in CE-ATA device (nIEN = 0), and RW_BLK command expects command completion signal from CE-ATA device<br>If the command expects Command Completion Signal (CCS) from the CE-ATA device, the software should set this control bit. SD/MMC sets Data Transfer Over bit in RINTSTS register and generates interrupt to host if Data Transfer Over interrupt is not masked. |
| 22 | R/W | 0 | RD_CEATA_DEV<br>read_ceata_device<br>0 – Host is not performing read access (RW_REG or RW_BLK) towards CE-ATA device<br>1 – Host is performing read access (RW_REG or RW_BLK) towards CE-ATA device<br>Software should set this bit to indicate that CE-ATA device is being accessed for read transfer. This bit is used to disable read data timeout indication while performing CE-ATA read transfers. Maximum value of I/O transmission delay can be no less than 10 seconds. SD/MMC should not indicate read data timeout while waiting for data from CE-ATA device. |
| 21 | R/W | 0 | PRG_CLK<br>Change Clock<br>0 – Normal command<br>1 – Change Card Clock; when this bit is set, controller will change clock domain and clock output. No command will be sent. |
| 20:16 | - | - | / |
| 15 | R/W | 0 | SEND_INIT_SEQ<br>Send Initialization<br>0 – normal command sending<br>1 – Send initialization sequence before sending this command. |
| 14 | R/W | 0 | STOP_ABT_CMD<br>Stop Abort Command<br>0 – normal command sending<br>1 – send Stop or abort command to stop current data transfer in progress.(CMD12, CMD52 for writing "I/O Abort" in SDIO CCCR) |
| 13 | R/W | 0 | WAIT_PRE_OVER<br>Wait Data Transfer Over<br>0 – Send command at once, do not care of data transferring<br>1 – Wait for data transfer completion before sending current command |
| 12 | R/W | 0 | STOP_CMD_FLAG<br>Send Stop CMD Automatically (CMD12)<br>0 – Do not send stop command at end of data transfer |

| | | | 1 – Send stop command automatically at end of data transfer |
|---|---|---|---|
| 11 | R/W | 0 | TRANS_MODE<br>Transfer Mode<br>0 – Block data transfer command<br>1 – Stream data transfer command |
| 10 | R/W | 0 | TRANS_DIR<br>Transfer Direction<br>0 – Read operation<br>1 – Write operation |
| 9 | R/W | 0 | DATA_TRANS<br>Data Transfer<br>0 – without data transfer<br>1 – with data transfer |
| 8 | R/W | 0 | CHK_RESP_CRC<br>Check Response CRC<br>0 – Do not check response CRC<br>1 – Check response CRC |
| 7 | R/W | 0 | LONG_RESP<br>Response Type<br>0 –Short Response (48 bits)<br>1 –Long Response (136 bits) |
| 6 | R/W | 0 | RESP_RCV<br>Response Receive<br>0 – Command without Response<br>1 – Command with Response |
| 5:0 | R/W | 0 | CMD_IDX<br>CMD Index<br>Command index value |

### 4.3.7.8. SD Command Argument Register (Default Value: 0x00000000)

| Offset: 0x001c | | | Register Name: **SD_CMDARG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R/W | 0 | CMD_ARG<br>Command argument |

### 4.3.7.9. SD Response 0 Register (Default Value: 0x00000000)

| Offset: 0x0020 | | | Register Name: **SD_RESP0** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R | 0 | CMD_RESP0<br>response 0 |

Bit[31:0] of response

#### 4.3.7.10. SD Response 1 Register (Default Value: 0x00000000)

| Offset: 0x0024 | | | Register Name: **SD_RESP1** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R | 0 | CMD_RESP1 response 1 Bit[63:31] of response |

#### 4.3.7.11. SD Response 2 Register (Default Value: 0x00000000)

| Offset: 0x0028 | | | Register Name: **SD_RESP2** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R | 0 | CMD_RESP2 response 2 Bit[95:64] of response |

#### 4.3.7.12. SD Response 3 Register (Default Value: 0x00000000)

| Offset: 0x002C | | | Register Name: **SD_RESP3** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R | 0 | CMD_RESP3 response 3 Bit[127:96] of response |

#### 4.3.7.13. SD Interrupt Mask Register (Default Value: 0x00000000)

| Offset: 0x0030 | | | Register Name: **SD_INTMASK** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R/W | 0 | INT_MASK 0 – interrupt masked 1 – interrupt enabled<br><br>Bit field defined as following:<br>bit 31– card removed<br>bit 30 – card inserted<br>bit 17~29 - reserved<br>bit 16 – SDIO interrupt |

| | | | bit 15 – Data End-bit error |
|---|---|---|---|
| | | | bit 14 – Auto Stop Command done |
| | | | bit 13 – Data Start Error |
| | | | bit 12 – Command Busy and illegal write |
| | | | bit 11 – FIFO under run/overflow |
| | | | bit 10 – Data starvation timeout /V1.8 Switch Done |
| | | | bit 9 – Data timeout/Boot data start |
| | | | bit 8 – Response timeout/Boot ACK received |
| | | | bit 7 – Data CRC error |
| | | | bit 6 – Response CRC error |
| | | | bit 5 – Data Receive Request |
| | | | bit 4 –Data Transmit Request |
| | | | bit 3 – Data Transfer Complete |
| | | | bit 2 – Command Complete |
| | | | bit 1 – Response Error (no response or response CRC error) |
| | | | bit 0 – Reserved |

### 4.3.7.14. SD Masked Interrupt Status Register (Default Value: 0x00000000)

| Offset: 0x0034 | | | Register Name: **SD_MINTSTS** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R | 0 | MSKD_ISTA<br>Interrupt status. Enabled only if corresponding bit in mask register is set.<br><br>Bit field defined as following:<br>bit 31 – card removed<br>bit 30 – card inserted<br>bit 17~29 - reserved<br>bit 16 – SDIO interrupt<br>bit 15 – Data End-bit error<br>bit 14 – Auto command done<br>bit 13 – Data Start Error<br>bit 12 – Command Busy and illegal write<br>bit 11 – FIFO under run/overflow<br>bit 10 – Data starvation timeout (HTO)/V1.8 Switch Done<br>bit 9 – Data timeout/Boot data start<br>bit 8 – Response timeout/Boot ACK received<br>bit 7 – Data CRC error<br>bit 6 – Response CRC error<br>bit 5 – Data Receive Request<br>bit 4 –Data Transmit Request<br>bit 3 – Data Transfer Complete<br>bit 2 – Command Complete<br>bit 1 – Response Error (no response or response CRC error) |

| | | | bit 0 – Reserved |
|---|---|---|---|

#### 4.3.7.15. SD Raw Interrupt Status Register (Default Value: 0x00000000)

| Offset: 0x0038 | | | Register Name: **SD_RINTSTS** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R/W | 0 | RAW_ISTA<br>Raw Interrupt Status.<br>*This is write-1-to-clear bits.*<br><br>Bit field defined as following:<br>bit 31 – card removed<br>bit 30 – card inserted<br>bit 17~29 - reserved<br>bit 16 – SDIO interrupt<br>bit 15 – Data End-bit error<br>bit 14 – Auto command done<br>bit 13 – Data Start Error<br>bit 12 – Command Busy and illegal write<br>bit 11 – FIFO under run/overflow<br>bit 10 – Data starvation timeout (HTO)/V1.8 Switch Done<br>bit 9 – Data timeout/Boot data start<br>bit 8 – Response timeout/Boot ACK received<br>bit 7 – Data CRC error<br>bit 6 – Response CRC error<br>bit 5 – Data Receive Request<br>bit 4 –Data Transmit Request<br>bit 3 – Data Transfer Complete<br>bit 2 – Command Complete<br>bit 1 – Response Error (no response or response CRC error)<br>bit 0 – Reserved |

#### 4.3.7.16. SD Status Register (Default Value: 0x00000006)

| Offset: 0x003C | | | Register Name: **SD_STATUS** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R | 0 | DMA_REQ<br>dma_req<br>DMA request signal state |
| 30:22 | / | / | / |
| 21:17 | R | 0 | FIFO_LEVEL<br>FIFO Level |

| | | | | Number of filled locations in FIFO |
|---|---|---|---|---|
| 16:11 | R | 0 | | RESP_IDX<br>Response Index<br>Index of previous response, including any auto-stop sent by controller |
| 10 | R | 0 | | FSM_BUSY<br>Data FSM Busy<br>Data transmit or receive state-machine is busy |
| 9 | R | 0 | | CARD_BUSY<br>Card data busy<br>Inverted version of DATA[0]<br>0 – card data not busy<br>1 – card data busy |
| 8 | R | 0 | | CARD_PRESENT<br>Data[3] status<br>level of DATA[3]; checks whether card is present<br>0 – card not present<br>1 – card present |
| 7:4 | R | 0 | | FSM_STA<br>Command FSM states:<br>0 – Idle<br>1 – Send init sequence<br>2 – Tx cmd start bit<br>3 – Tx cmd tx bit<br>4 – Tx cmd index + arg<br>5 – Tx cmd crc7<br>6 – Tx cmd end bit<br>7 – Rx resp start bit<br>8 – Rx resp IRQ response<br>9 – Rx resp tx bit<br>10 – Rx resp cmd idx<br>11 – Rx resp data<br>12 – Rx resp crc7<br>13 – Rx resp end bit<br>14 – Cmd path wait NCC<br>15 – Wait; CMD-to-response turnaround |
| 3 | R | 0 | | FIFO_FULL<br>FIFO full<br>1 – FIFO full<br>0 – FIFO not full |
| 2 | R | 1 | | FIFO_EMPTY<br>FIFO Empty<br>1 - FIFO Empty<br>0 - FIFO not Empty |
| 1 | R | 1 | | FIFO_TX_LEVEL<br>FIFO TX Water Level flag |

| | | | 0 – FIFO didn't reach transmit trigger level |
|---|---|---|---|
| | | | 1 - FIFO reached transmit trigger level |
| 0 | R | 0 | FIFO_RX_LEVEL |
| | | | FIFO TX Water Level flag |
| | | | 0 – FIFO didn't reach receive trigger level |
| | | | 1 - FIFO reached receive trigger level |

### 4.3.7.17. SD FIFO Water Level Register (Default Value: 0x000F0000)

| Offset: 0x0040 | | | Register Name: **SD_FIFOTH** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | / | / | / |
| 30:28 | R/W | 0 | BSIZE_OF_TRANS<br>Burst size of multiple transaction<br>000 – 1 transfers<br>001 – 4<br>010 – 8<br>011 – 16<br>100 – 32<br>101 – 64<br>110 – 128<br>111 – 256<br>Should be programmed same as DMA controller multiple transaction size. The units for transfers are the DWORD. A single transfer would be signaled based on this value. Value should be sub-multiple of (RX_TL + 1) and (FIFO_DEPTH - TX_TL)<br>Recommended:<br>MSize = 8, TX_TL = 16, RX_TL = 15 |
| 27:21 | R | 0 | / |
| 20:16 | R/W | 0xF | RX_TL<br>Rx Trigger Level<br>0x0~0x1e – RX Trigger Level is 0~30<br>0x1f – reserved<br>FIFO threshold when FIFO request host to receive data from FIFO. When FIFO data level is greater than this value, DMA is request is raised if DMA enabled, or RX interrupt bit is set if interrupt enabled. At the end of packet, if the last transfer is less than this level, the value is ignored and relative request will be raised as usual.<br>Recommended: 15 (means greater than 15) |
| 15:5 | R | 0 | / |
| 4:0 | R/W | 0 | TX_TL<br>TX Trigger Level<br>0x1~0xf – TX Trigger Level is 1~31<br>0x0 – no trigger |

| | | | FIFO threshold when FIFO requests host to transmit data to FIFO. When FIFO data level is less than or equal to this value, DMA TX request is raised if DMA enabled, or TX request interrupt bit is set if interrupt enabled. At the end of packet, if the last transfer is less than this level, the value is ignored and relative request will be raised as usual. |
|---|---|---|---|
| | | | Recommended: 16 (means less than or equal to 16) |

### 4.3.7.18. SD Function Select Register (Default Value: 0x00000000)

| Offset: 0x0044 | | | Register Name: **SD_CTRL** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:16 | R/W | 0 | CEATA_EN<br>CEATA Support ON/OFF<br>0xceaa – CEATA support on. All hidden CEATA relative bits are accessible normally and these 16 bits return value of 0x1 when be read.<br>Other value – CEATA support off. All hidden CEATA relative bits cannot be access and these 16 bits return value of 0 when be read. |
| 15:11 | - | 0 | / |
| 10 | R/W | 0 | CEATA_ISTA<br>ceata device interrupt status<br>0 – Interrupts not enabled in CE-ATA device (nIEN = 1 in ATA control register)<br>1 – Interrupts are enabled in CE-ATA device (nIEN = 0 in ATA control register)<br>Software should appropriately write to this bit after power-on reset or any other reset to CE-ATA device. After reset, usually CE-ATA device interrupt is disabled (nIEN = 1). If the host enables CE-ATA device interrupt, then software should set this bit. |
| 9 | R/W | 0 | SEND_AUTO_STOP_CCSD<br>send auto stop ccsd<br>0 – Clear bit if SD/MMC does not reset the bit.<br>1 – Send internally generated STOP after sending CCSD to CEATA device.<br>When set, SD/MMC automatically sends internally generated STOP command (CMD12) to CE-ATA device. After sending internally-generated STOP command, Auto Command Done (ACD) bit in RINTSTS is set and generates interrupt to host if Auto Command Done interrupt is not masked. After sending the CCSD, SD/MMC automatically clears send_auto_stop_ccsd bit. |
| 8 | R/W | 0 | SEND_CCSD<br>send ccsd<br>0 – Clear bit if SD/MMC does not reset the bit.<br>1 – Send Command Completion Signal Disable (CCSD) to CE-ATA device<br>When set, SD/MMC sends CCSD to CE-ATA device. Software sets this bit only if current command is expecting CCS (that is, RW_BLK) and interrupts are enabled in CE-ATA device. Once the CCSD pattern is sent to device, SD/MMC automatically clears send_ccsd bit. It also sets Command Done (CD) bit in |

| | | | |
|---|---|---|---|
| | | | RINTSTS register and generates interrupt to host if Command Done interrupt is not masked. |
| 7:3 | - | - | / |
| 2 | R/W | 0 | ABT_RDATA<br>Abort Read Data<br>0 – Ignored<br>1 –After suspend command is issued during read-transfer, software polls card to find when suspend happened. Once suspend occurs, software sets bit to reset data state-machine, which is waiting for next block of data.<br>Used in SDIO card suspends sequence.<br>This bit is auto-cleared once controller reset to idle state. |
| 1 | R/W | 0 | READ_WAIT<br>Read Wait<br>0 – Clear SDIO read wait<br>1 – Assert SDIO read wait |
| 0 | R/W | 0 | HOST_SEND_MMC_IRQRESQ<br>Host Send MMC IRQ Response<br>0 – Ignored<br>1 – Send auto IRQ response<br>When host is waiting MMC card interrupt response, setting this bit will make controller cancel wait state and return to idle state, at which time, controller will receive IRQ response sent by itself.<br>This bit is auto-cleared after response is sent. |

## 4.3.7.19. SD Auto Command 12 Register (Default Value: 0x0000FFFF)

| Offset: 0x0058 | | | Register Name: **SD_A12A** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:16 | - | - | / |
| 0:15 | R/W | 0xffff | SD_A12A.<br>SD_A12A set the argument of command 12 automatically send by controller |

## 4.3.7.20. SD NewTiming Set Register (Default Value: 0x00000001,only used in SDC2)

| Offset: 0x005C | | | Register Name: **SD_NTSR_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0 | MODE_SELEC<br>0 - Old mode of Sample/Output Timing ;<br>1 - New mode of Sample/Output Timing;<br>Default value : 0; |
| 30:6 | R/W | 0x00 | SAMPLE_TIMING_PHASE(RX)<br>00 - Sample timing phase offset 90。; |

| | | | 01 - Sample timing phase offset 180。; |
|---|---|---|---|
| | | | 10 - Sample timing phase offset 270。; |
| | | | 11 - Ignore; |
| | | | Default value : 00; |
| 3:2 | / | / | / |
| 1:0 | R/W | 0x01 | OUTPUT_TIMING_PHASE(TX)<br>00 - Output timing phase offset 90。;<br>01 - Output timing phase offset 180。;<br>10 - Output timing phase offset 270。;<br>11 - Ignore;<br>Default value : 01; |

### 4.3.7.21. SD Hardware Reset Register (Default Value: 0x00000001)

| Offset: 0x0078 | | | Register Name: **SD_HWRST** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:1 | / | / | / |
| 0 | R/W | 1 | HW_RESET.<br>1 – Active mode<br>0 – Reset<br>These bits cause the cards to enter pre-idle state, which requires them to be re-initialized. |

### 4.3.7.22. SD DMAC Control Register (Default Value: 0x0000_0000)

| Offset: 0x0080 | | | Register Name: **SD_BUS_MODE** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | W | 0 | DES_LOAD_CTRL<br>When DMAC fetches a descriptor, if the valid bit of a descriptor is not set, DMAC FSM will go to the suspend state. Setting this bit will make DMAC re-fetch descriptor again and do the transfer normally. |
| | | | |
| 10:8 | R | 0 | PRG_BURST_LEN<br>Programmable Burst Length.<br>These bits indicate the maximum number of beats to be performed in one IDMAC transaction. The IDMAC will always attempt to burst as specified in PBL each time it starts a Burst transfer on the host bus. The permissible values are 1, 4, 8, 16, 32, 64, 128 and 256. This value is the mirror of MSIZE of FIFOTH register. In order to change this value, write the required value to FIFOTH register. This is an encode value as follows.<br>000 – 1 transfers<br>001 – 4 transfers |

| | | | 010 – 8 transfers |
|---|---|---|---|
| | | | 011 – 16 transfers |
| | | | 100 – 32 transfers |
| | | | 101 – 64 transfers |
| | | | 110 – 128 transfers |
| | | | 111 – 256 transfers |
| | | | Transfer unit is either 16, 32, or 64 bits, based on HDATA_WIDTH. PBL is a read-only value. |
| 7 | R/W | 0 | IDMAC_ENB<br>IDMAC Enable.<br>When set, the IDMAC is enabled. DE is read/write. |
| 6:2 | R/W | 0 | DES_SKIP_LEN<br>Descriptor Skip Length.<br>Specifies the number of Word to skip between two unchained descriptors. This is applicable only for dual buffer structure.<br>Default value is set to 4 DWORD. |
| 1 | R/W | 0 | FIX_BUST_CTRL<br>Fixed Burst.<br>Controls whether the AHB Master interface performs fixed burst transfers or not. When set, the AHB will use only SINGLE, INCR4, INCR8 during start of normal burst transfers. When reset, the AHB will use SINGLE and INCR burst transfer operations. |
| 0 | R/W | 0 | IDMAC_RST<br>DMA Reset.<br>When set, the DMA Controller resets all its internal registers. SWR is read/write. It is automatically cleared after 1 clock cycle. |

### 4.3.7.23. SD Descriptor List Base Address Register (Default Value: 0x00000000)

| Offset: 0x0084 | | | Register Name: **SD_DLBA** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R/W | 0 | DES_BASE_ADDR<br>Start of Descriptor List.<br>Contains the base address of the First Descriptor. The LSB bits [1:0] are ignored and taken as all-zero by the IDMAC internally. Hence these LSB bits are read-only. |

### 4.3.7.24. SD DMAC Status Register (Default Value: 0x00000000)

| Offset: 0x0088 | | | Register Name: **SD_DSR** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:17 | / | / | / |

| 16:13 | R | 0 | DMAC_FSM_STA<br>DMAC FSM present state.<br>0 – DMA_IDLE<br>1 – DMA_SUSPEND<br>2 – DESC_RD<br>3 – DESC_CHK<br>4 – DMA_RD_REQ_WAIT<br>5 – DMA_WR_REQ_WAIT<br>6 – DMA_RD<br>7 – DMA_WR<br>8 – DESC_CLOSE<br>This bit is read-only. |
|---|---|---|---|
| 12:10 | R | 0 | DMAC_ERR_STA<br>Error Bits.<br>Indicates the type of error that caused a Bus Error. Valid only with Fatal Bus Error bit (IDSTS[2]) set. This field does not generate an interrupt.<br>3'b001 – Host Abort received during transmission<br>3'b010 – Host Abort received during reception<br>Others: Reserved EB is read-only. |
| 9 | R/W | 0 | ABN_INT_SUM<br>Abnormal Interrupt Summary.<br>Logical OR of the following:<br>IDSTS[2] – Fatal Bus Interrupt<br>IDSTS[4] – DU bit Interrupt<br>IDSTS[5] – Card Error Summary Interrupt<br>Only unmasked bits affect this bit.<br>This is a sticky bit and must be cleared each time a corresponding bit that causes AIS to be set is cleared. Writing a 1 clears this bit. |
| 8 | R/W | 0 | NOR_INT_SUM<br>Normal Interrupt Summary.<br>Logical OR of the following:<br>IDSTS[0] – Transmit Interrupt<br>IDSTS[1] – Receive Interrupt<br>Only unmasked bits affect this bit.<br>This is a sticky bit and must be cleared each time a corresponding bit that causes NIS to be set is cleared. Writing a 1 clears this bit. |
| 7:6 | / | / | / |
| 5 | R/W | 0 | ERR_FLAG_SUM<br>Card Error Summary.<br>Indicates the status of the transaction to/from the card; also present in RINTSTS. Indicates the logical OR of the following bits:<br>EBE – End Bit Error<br>RTO – Response Timeout/Boot Ack Timeout<br>RCRC – Response CRC<br>SBE – Start Bit Error |

| | | | DRTO – Data Read Timeout/BDS timeout |
|---|---|---|---|
| | | | DCRC – Data CRC for Receive |
| | | | RE – Response Error |
| | | | Writing a 1 clears this bit. |
| 4 | R/W | 0 | DES_UNAVL_INT |
| | | | Descriptor Unavailable Interrupt. |
| | | | This bit is set when the descriptor is unavailable due to OWN bit = 0 (DES0[31] =0). Writing a 1 clears this bit. |
| 3 | / | / | / |
| 2 | R/W | 0 | FATAL_BERR_INT |
| | | | Fatal Bus Error Interrupt. |
| | | | Indicates that a Bus Error occurred (IDSTS[12:10]). When this bit is set, the DMA disables all its bus accesses. Writing a 1 clears this bit. |
| 1 | R/W | 0 | RX_INT |
| | | | Receive Interrupt. |
| | | | Indicates the completion of data reception for a descriptor. Writing a 1 clears this bit. |
| 0 | R/W | 0 | TX_INT |
| | | | Transmit Interrupt. |
| | | | Indicates that data transmission is finished for a descriptor. Writing a '1' clears this bit. |

### 4.3.7.25. SD DMAC Interrupt Enable Register (Default Value: 0x00000000)

| Offset: 0x008C | | | Register Name: **SD_IDIE_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:10 | / | / | / |
| 9 | R/W | 0 | ABN_INT_ENB |
| | | | Abnormal Interrupt Summary Enable. |
| | | | When set, an abnormal interrupt is enabled. This bit enables the following bits: |
| | | | IDINTEN[2] – Fatal Bus Error Interrupt |
| | | | IDINTEN[4] – DU Interrupt |
| | | | IDINTEN[5] – Card Error Summary Interrupt |
| 8 | R/W | 0 | NOR_INT_ENB |
| | | | Normal Interrupt Summary Enable. |
| | | | When set, a normal interrupt is enabled. When reset, a normal interrupt is disabled. This bit enables the following bits: |
| | | | IDINTEN[0] – Transmit Interrupt |
| | | | IDINTEN[1] – Receive Interrupt |
| 7:6 | / | / | / |
| 5 | R/W | 0 | ERR_SUM_INT_ENB |
| | | | Card Error summary Interrupt Enable. |
| | | | When set, it enables the Card Interrupt summary. |

| Bit | R/W | Default/Hex | Description |
|---|---|---|---|
| 4 | R/W | 0 | DES_UNAVL_INT_ENB<br>Descriptor Unavailable Interrupt.<br>When set along with Abnormal Interrupt Summary Enable, the DU interrupt is enabled. |
| 3 | / | / | / |
| 2 | R/W | 0 | FERR_INT_ENB<br>Fatal Bus Error Enable.<br>When set with Abnormal Interrupt Summary Enable, the Fatal Bus Error Interrupt is enabled. When reset, Fatal Bus Error Enable Interrupt is disabled. |
| 1 | R/W | 0 | RX_INT_ENB<br>Receive Interrupt Enable.<br>When set with Normal Interrupt Summary Enable, Receive Interrupt is enabled. When reset, Receive Interrupt is disabled. |
| 0 | R/W | 0 | TX_INT_ENB<br>Transmit Interrupt Enable.<br>When set with Normal Interrupt Summary Enable, Transmit Interrupt is enabled. When reset, Transmit Interrupt is disabled. |

### 4.3.7.26. Card Threshold Control Register (Default Value: 0x00000000)

| Offset: 0x0100 | | | Register Name: **SD_THLD_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:28 | / | / | / |
| 27:16 | R/W | 0 | CARD_RD_THLD<br>Card Read Threshold Size |
| 15:1 | / | / | / |
| 0 | R/W | 0 | CARD_RD_THLD_ENB<br>Card Read Threshold Enable<br>0 – Card Read Threshold Disable<br>1 - Card Read Threshold Enable<br>Host controller initiates Read Transfer only if CARD_RD_THLD amount of space is available in receive FIFO |

### 4.3.7.27. eMMC4.41 DDR Start Bit Detection Control Register (Default Value: 0x00000000)

| Offset: 0x010C | | | Register Name: **EMMC_DDR_SBIT_DET _REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:1 | / | / | / |
| 0 | R/W | 0 | HALF_START_BIT<br>Control for start bit detection mechanism inside mstorage based on duration of start bit. |

| | | | For eMMC 4.41, start bit can be: |
|---|---|---|---|
| | | | 0 - Full cycle |
| | | | 1 - Less than one full cycle |
| | | | Set HALF_START_BIT=1 for eMMC 4.41 and above; set to 0 for SD applications. |

### 4.3.7.28. SD FIFO Register (Default Value: 0x00000000)

| Offset: 0x0200 | | | Register Name: **SD_FIFO_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R/W | 0 | TX/RX_FIFO |
| | | | Data FIFO |

# Chapter 5  Image

This section describes the image of A83T:

- MIPI CSI
- CSI

## 5.1.   MIPI CSI

### 5.1.1.   Overview

The MIPI CSI includes the following feature:

- Compliant with MIPI-CSI2 V1.00 and MIPI DPHY V1.00
- 1/2/3/4 Data Lanes Configuration and up to 1Gbps per Lane in HS Transmission
- Maximum to 8M@30fps with 4 data lane
- Supports format: YUV422-8bit/10bit,YUV420-8bit/10bit,RAW-8,RAW-10, RAW-12,RGB888,RGB565

## 5.2.   CSI

### 5.2.1.   Overview

The CSI includes the following feature:

- Support 10bit yuv422 /raw CMOS sensor interface
- Support CCIR656 protocol for NTSC and PAL
- Support parallel interface still capture resolution up to 5M,video capture resolution up to 720p@30fps
- Support MIPI interface still capture resolution up to 8M,video capture resolution up to 1080p@60fps

## 5.2.2.  Functionalities Description

### 5.2.2.1.  Block Diagram
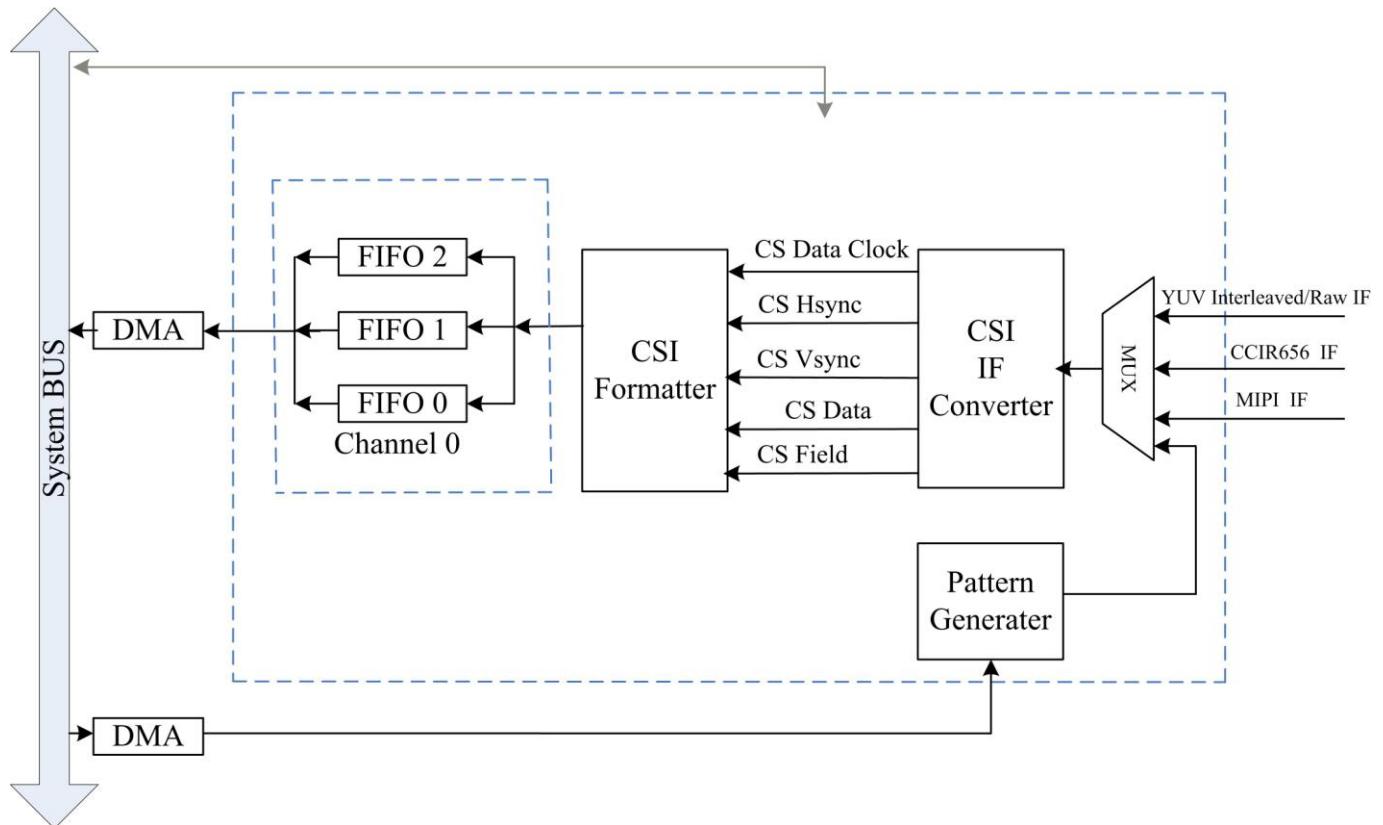


Figure 5-1. CSI Block Diagram

### 5.2.2.2.  Timing
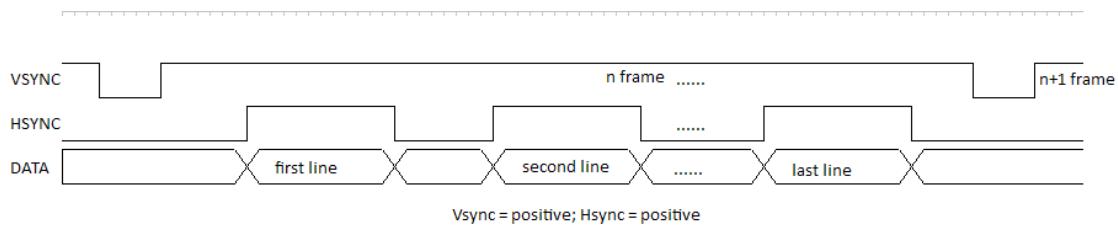


Figure 5-2. Vref= positive; Href= positive
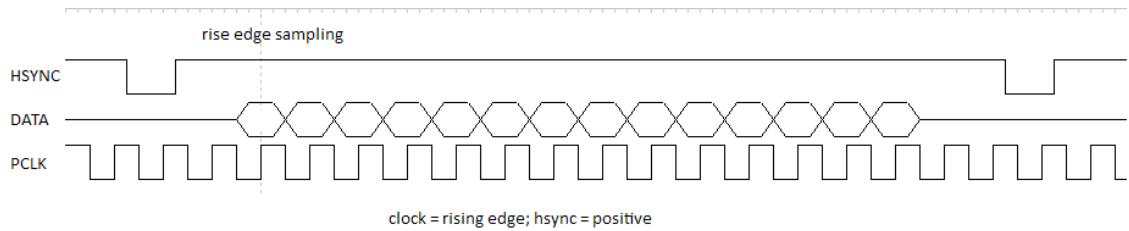
Figure 5-3. horizontal size setting and pixel clock timing(Href= positive)

### 5.2.2.3. Bit Definition

| Data Bit | First Word(0xFF) | Second Word(0x00) | Third Word(0x00) | Fourth Word |
|---|---|---|---|---|
| CS D[9] (MSB) | 1 | 0 | 0 | 1 |
| CS D[8] | 1 | 0 | 0 | F |
| CS D[7] | 1 | 0 | 0 | V |
| CS D[6] | 1 | 0 | 0 | H |
| CS D[5] | 1 | 0 | 0 | P3 |
| CS D[4] | 1 | 0 | 0 | P2 |
| CS D[3] | 1 | 0 | 0 | P1 |
| CS D[2] | 1 | 0 | 0 | P0 |
| CS D[1] | x | x | x | x |
| CS D[0] | x | x | x | x |

**Note:**

For compatibility with 8-bit interface, CS D[1] and CS D[0] are not defined.

| Decode | F | V | H | P3 | P2 | P1 | P0 |
|---|---|---|---|---|---|---|---|
| Field 1 start of active video (SAV) | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Field 1 end of active video (EAV) | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| Field 1 SAV (digital blanking) | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| Field 1 EAV (digital blanking) | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| Field 2 SAV | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| Field 2 EAV | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| Field 2 SAV (digital blanking) | 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| Field 2 EAV (digital blanking) | 1 | 1 | 1 | 0 | 0 | 0 | 1 |

## 5.2.3. Register list

| Module Name | Base Address |
|---|---|
| CSI0 | 0x01CB0000 |

| Register Name | Offset | Register name |
|---|---|---|
| CSI0_EN_REG | 0X0000 | CSI Enable register |
| CSI0_IF_CFG_REG | 0X0004 | CSI Interface Configuration Register |
| CSI0_CAP_REG | 0X0008 | CSI Capture Register |
| CSI0_SYNC_CNT_REG | 0X000C | CSI Synchronization Counter Register |
| CSI0_FIFO_THRS_REG | 0X0010 | CSI FIFO Threshold Register |
| CSI0_PTN_LEN_REG | 0X0030 | CSI Pattern Generation Length register |
| CSI0_PTN_ADDR_REG | 0X0034 | CSI Pattern Generation Address register |
| CSI0_VER_REG | 0X003C | CSI Version Register |
| CSI0_C0_CFG_REG | 0X0044 | CSI Channel_0 configuration register |
| CSI0_C0_SCALE_REG | 0X004C | CSI Channel_0 scale register |
| CSI0_C0_F0_BUFA_REG | 0X0050 | CSI Channel_0 FIFO 0 output buffer-A address register |
| CSI0_C0_F1_BUFA_REG | 0X0058 | CSI Channel_0 FIFO 1 output buffer-A address register |
| CSI0_C0_F2_BUFA_REG | 0X0060 | CSI Channel_0 FIFO 2 output buffer-A address register |
| CSI0_C0_CAP_STA_REG | 0X006C | CSI Channel_0 status register |
| CSI0_C0_INT_EN_REG | 0X0070 | CSI Channel_0 interrupt enable register |
| CSI0_C0_INT_STA_REG | 0X0074 | CSI Channel_0 interrupt status register |
| CSI0_C0_HSIZE_REG | 0X0080 | CSI Channel_0 horizontal size register |
| CSI0_C0_VSIZE_REG | 0X0084 | CSI Channel_0 vertical size register |
| CSI0_C0_BUF_LEN_REG | 0X0088 | CSI Channel_0 line buffer length register |
| CSI0_C0_FLIP_SIZE_REG | 0X008C | CSI Channel_0 flip size register |
| CSI0_C0_FRM_CLK_CNT_REG | 0X0090 | CSI Channel_0 frame clock counter register |
| CSI0_C0_ACC_ITNL_CLK_CNT_REG | 0X0094 | CSI Channel_0 accumulated and internal clock counter register |
| CSI0_C0_FIFO_STAT_REG | 0X0098 | CSI Channel_0 FIFO Statistic Register |
| CSI0_C0_PCLK_STAT_REG | 0X009C | CSI Channel_0 PCLK Statistic Register |
| CCI_CTRL | 0x3000 | CCI control register |
| CCI_CFG | 0x3004 | CCI transmission config register |
| CCI_FMT | 0x3008 | CCI packet format register |
| CCI_BUS_CTRL | 0x300C | CCI bus control register |
| CCI_INT_CTRL | 0x3014 | CCI interrupt control register |
| CCI_LC_TRIG | 0x3018 | CCI line counter trigger register |
| CCI_FIFO_ACC | 0x3100 | CCI FIFO access register |
| CCI_RSV_REG | 0x3200 | CCI reserved register |

## 5.2.4. Register Description

### 5.2.4.1. CSI Enable Register (Default Value: 0x00000000)

| Offset: 0x0000 | | | Register Name: **CSI0_EN_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | / | / | / |
| 30 | R/W | 0x0 | VER_EN<br>CSI Version Register Read Enable:<br>0: Disable<br>1: Enable |
| 29:24 | / | / | / |
| 23:16 | R/W | 0x00 | PTN_CYCLE<br>Pattern generating cycle counter.<br>The pattern in dram will be generated in cycles of PTN_CYCLE+1. |
| 15:9 | / | / | / |
| 8 | R/W | 0x0 | SRAM_PWDN<br>0: SRAM in normal<br>1: SRAM in power down |
| 7:5 | / | / | / |
| 4 | R/W | 0x0 | PTN_START<br>CSI Pattern Generating Start<br>0: Finish<br>other: Start<br>Software write this bit to"1" to start pattern generating from DRAM. When finished, the hardware will clear this bit to "0" automatically. Generating cycles depends on PTN_CYCLE. |
| 3 | R/W | 0 | CLK_CNT_SPL<br>Sampling time for clk counter per frame<br>0: Sampling clock counter every frame done<br>1: Sampling clock counter every vsync |
| 2 | R/W | 0 | CLK_CNT_EN<br>clk count per frame enable |
| 1 | R/W | 0 | PTN_GEN_EN<br>Pattern Generation Enable |
| 0 | R/W | 0 | CSI_EN<br>Enable<br>0: Reset and disable the CSI module<br>1: Enable the CSI module |

### 5.2.4.2. CSI Interface Configuration Register (Default Value: 0x00000000)

| Offset: 0x0004 | | | Register Name: **CSI0_IF_CFG_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:22 | / | / | / |
| 21 | R/W | 0 | SRC_TYPE<br>Source type<br>0: Progressed<br>1: Interlaced |
| 20 | R/W | 0 | FPS_DS<br>Fps down sample<br>0: no down sample<br>1: 1/2 fps, only receives the first frame every 2 frames |
| 19 | R/W | 0 | FIELD<br>For YUV HV timing, Field polarity<br>0: negative(field=0 indicate odd, field=1 indicate even )<br>1: positive(field=1 indicate odd, field=0 indicate even )<br>For BT656 timing, Field sequence<br>0: Normal sequence (field 0 first)<br>1: Inverse sequence (field 1 first) |
| 18 | R/W | 1 | VREF_POL<br>Vref polarity<br>0: negative<br>1: positive<br>This register is not apply to CCIR656 interface. |
| 17 | R/W | 0 | HERF_POL<br>Href polarity<br>0: negative<br>1: positive<br>This register is not apply to CCIR656 interface. |
| 16 | R/W | 1 | CLK_POL<br>Data clock type<br>0: active in rising edge<br>1: active in falling edge |
| 15:12 | / | / | / |
| 11:10 | R/W | 0 | SEQ_8PLUS2<br>When select IF_DATA_WIDTH to be 8+2bit, odd/even pixel byte at CSI-D[11:4] will be rearranged to D[11:2]+2'b0 at the actual csi data bus according to these sequences:<br>00: 6'bx+D[9:8], D[7:0]<br>01: D[9:2], 6'bx+D[1:0]<br>10: D[7:0], D[9:8]+6'bx<br>11: D[7:0], 6'bx+D[9:8] |
| 9:8 | R/W | 0 | IF_DATA_WIDTH<br>00: 8 bit data bus |

| | | | 01: 10 bit data bus |
|---|---|---|---|
| | | | 10: 12 bit data bus |
| | | | 11: 8+2bit data bus |
| 7 | R/W | 0 | MIPI_IF |
| | | | MIPI Interface Enable: |
| | | | 0: CSI |
| | | | 1: MIPI |
| 6:5 | / | / | / |
| 4:0 | R/W | 0 | CSI_IF |
| | | | YUV: |
| | | | 00000: YUYV422 Interleaved or RAW (All data in one data bus) |
| | | | CCIR656: |
| | | | 00100: YUYV422 Interleaved or RAW (All data in one data bus) |
| | | | Others: Reserved |

### 5.2.4.3. CSI Capture Register (Default Value: 0x00000000)

| Offset: 0x0008 | | | Register Name: **CSI0_CAP_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:6 | / | / | / |
| 5:2 | R/W | 0x0 | CH0_CAP_MASK |
| | | | Vsync number masked before capture. |
| 1 | R/W | 0x0 | CH0_VCAP_ON |
| | | | Video capture control: Capture the video image data stream on channel 0. |
| | | | 0: Disable video capture |
| | | | If video capture is in progress, the CSI stops capturing image data at the end of the current frame, and all of the current frame data is wrote to output FIFO. |
| | | | 1: Enable video capture |
| | | | The CSI starts capturing image data at the start of the next frame. |
| 0 | R/W | 0x0 | CH0_SCAP_ON |
| | | | Still capture control: Capture a single still image frame on channel 0. |
| | | | 0: Disable still capture. |
| | | | 1: Enable still capture |
| | | | The CSI module starts capturing image data at the start of the next frame. The CSI module captures only one frame of image data. This bit is self clearing and always reads as a 0. |

### 5.2.4.4. CSI Synchronization Counter Register (Default Value: 0x00000000)

| Offset: 0x000C | | | Register Name: **CSI0_SYNC_CNT_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |

| 31:24 | / | / | / |
|---|---|---|---|
| 23:0 | R | 0 | SYNC_CNT<br>The counter value between vsync of CSI0 channel 0 and vsync of CSI1 channel 0 , using 24MHz. |

### 5.2.4.5. CSI FIFO Threshold Register (Default Value: 0x040f0400)

| Offset: 0x0010 | | | Register Name: **CSI0_FIFO_THRS_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:29 | / | / | / |
| 28:26 | R/W | 0x1 | FIFO_NEARLY_FULL_TH<br>The threshold of FIFO being nearly full. Indicates that the ISP should stop writing. Only valid when ISP is enabled.<br>0~7:<br>The smaller the value, the flag of FIFO being nearly full is easier to reach. |
| 25:24 | R/W | 0x0 | PTN_GEN_CLK_DIV<br>Packet generator clock divider |
| 23:16 | R/W | 0x0f | PTN_GEN_DLY<br>Clocks delayed before pattern generating start. |
| 15:12 | / | / | / |
| 11:00 | R/W | 0x400 | FIFO_THRS<br>When CSI0 FIFO occupied memory exceed the threshold, dram frequency can not change. |

### 5.2.4.6. CSI Pattern Generation Length Register (Default Value: 0x00000000)

| Offset: 0x0030 | | | Register Name: **CSI0_PTN_LEN_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R/W | 0x0 | PTN_LEN<br>The pattern length in byte when generating pattern. |

### 5.2.4.7. CSI Pattern Generation Address Register (Default Value: 0x00000000)

| Offset: 0x0034 | | | Register Name: **CSI0_PTN_ADDR_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R/W | 0x0 | PTN_ADDR<br>The pattern DRAM address when generating pattern. |

#### 5.2.4.8. CSI Version Register (Default Value: 0x00000000)

| Offset: 0x003C | | | Register Name: **CSI0_VER_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R | 0x0 | VER<br>Version of hardware circuit. Only can be read when version register read enable is on. |

#### 5.2.4.9. CSI Channel_0 configuration Register (Default Value: 0x00300200)

| Offset: 0x0044 | | | Register Name: **CSI0_C0_CFG_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:24 | R/W | 0 | PAD_VAL<br>Padding value when OUTPUT_FMT is prgb888<br>0x00~0xff |
| 23:20 | R/W | 3 | INPUT_FMT<br>Input data format<br>0000: RAW stream<br>0001: reserved<br>0010: reserved<br>0011: YUV422<br>0100: YUV420<br>Others: reserved |
| 19:16 | R/W | 0 | OUTPUT_FMT<br>Output data format<br>When the input format is set RAW stream<br>0000: field-raw-8<br>0001: field-raw-10<br>0010: field-raw-12<br>0011: reserved<br>0100: field-rgb565<br>0101: field-rgb888<br>0110: field-prgb888<br>0111: field-uv-combined<br>1000: frame-raw-8<br>1001: frame-raw-10<br>1010: frame-raw-12<br>1011: reserved<br>1100: frame-rgb565<br>1101: frame-rgb888<br>1110: frame-prgb888<br>1111: frame-uv-combined |

| | | | |
|---|---|---|---|
| | | | When the input format is set YUV422 |
| | | | 0000: field planar YCbCr 422 |
| | | | 0001: field planar YCbCr 420 |
| | | | 0010: frame planar YCbCr 420 |
| | | | 0011: frame planar YCbCr 422 |
| | | | 0100: field planar YCbCr 422 UV combined |
| | | | 0101: field planar YCbCr 420 UV combined |
| | | | 0110: frame planar YCbCr 420 UV combined |
| | | | 0111: frame planar YCbCr 422 UV combined |
| | | | 1000: field MB YCbCr 422 |
| | | | 1001: field MB YCbCr 420 |
| | | | 1010: frame MB YCbCr 420 |
| | | | 1011: frame MB YCbCr 422 |
| | | | 1100: field planar YCbCr 422 10bit UV combined |
| | | | 1101: field planar YCbCr 420 10bit UV combined |
| | | | 1110: Reserved |
| | | | 1111: Reserved |
| | | | |
| | | | When the input format is set YUV420 |
| | | | 0000: Reserved |
| | | | 0001: field planar YCbCr 420 |
| | | | 0010: frame planar YCbCr 420 |
| | | | 0011: Reserved |
| | | | 0100: Reserved |
| | | | 0101: field planar YCbCr 420 UV combined |
| | | | 0110: frame planar YCbCr 420 UV combined |
| | | | 0111: Reserved |
| | | | 1000: Reserved |
| | | | 1001: field MB YCbCr 420 |
| | | | 1010: frame MB YCbCr 420 |
| | | | 1011: Reserved |
| | | | 1100: Reserved |
| | | | 1101: field planar YCbCr 420 10bit UV combined |
| | | | 1110: Reserved |
| | | | 1111: Reserved |
| | | | |
| | | | Others: reserved |
| 15:14 | / | / | / |
| 13 | R/W | 0 | VFLIP_EN<br>Vertical flip enable<br>When enabled, the received data will be arranged in vertical flip.<br>0:Disable<br>1:Enable |
| 12 | R/W | 0 | HFLIP_EN<br>Horizontal flip enable |

| | | | When enabled, the received data will be arranged in horizontal flip.<br>0:Disable<br>1:Enable |
|---|---|---|---|
| 11:10 | R/W | 0 | FIELD_SEL<br>Field selection.<br>00: capturing with field 1.<br>01: capturing with field 2.<br>10: capturing with either field.<br>11: reserved |
| 09:08 | R/W | 2 | INPUT_SEQ<br>Input data sequence, only valid for YUV422 and YUV420 input format.<br>All data interleaved in one channel:<br>00: YUYV<br>01: YVYU<br>10: UYVY<br>11: VYUY<br>Y and UV in separated channel:<br>x0: UV<br>x1: VU |
| 07:02 | / | / | / |
| 01:00 | R/W | 0 | MIN_SDR_WR_SIZE<br>Minimum size of SDRAM block write<br>0: 256 bytes (if hflip is enable, always select 256 bytes)<br>1: 512 bytes<br>2: 1k bytes<br>3: 2k bytes |

### 5.2.4.10. CSI Channel_0 scale Register (Default Value: 0x00000000)

| Offset: 0x004C | | | Register Name: **CSI0_C0_SCALE_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:01 | / | / | / |
| 00 | R/W | 0 | QUART_EN<br>When this bit is set to 1, input image will be decimated to quarter size. All input format are supported. |

### 5.2.4.11. CSI Channel_0 FIFO 0 output buffer-A address Register (Default Value: 0x00000000)

| Offset: 0x0050 | | | Register Name: **CSI0_C0_F0_BUFA_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:00 | R/W | 0 | C0F0_BUFA<br>FIFO 0 output buffer-A address |

### 5.2.4.12. CSI Channel_0 FIFO 1 output buffer-A address Register (Default Value: 0x00000000)

| Offset: 0x0058 | | | Register Name: **CSI0_C0_F1_BUFA_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:00 | R/W | 0 | C0F1_BUFA<br>FIFO 1 output buffer-A address |

### 5.2.4.13. CSI Channel_0 FIFO 2 output buffer-A address Register (Default Value: 0x00000000)

| Offset: 0x0060 | | | Register Name: **CSI0_C0_F2_BUFA_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:00 | R/W | 0 | C0F2_BUFA<br>FIFO 2 output buffer-A address |

### 5.2.4.14. CSI Channel_0 status Register (Default Value: 0x00000000)

| Offset: 0x006C | | | Register Name: **CSI0_C0_CAP_STA_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:03 | / | / | / |
| 02 | R | 0 | FIELD_STA<br>The status of the received field<br>0: Field 0<br>1: Field 1 |
| 01 | R | 0 | VCAP_STA<br>Video capture in progress<br>Indicates the CSI is capturing video image data (multiple frames). The bit is set at the start of the first frame after enabling video capture. When software disables video capture, it clears itself after the last pixel of the current frame is captured. |
| 00 | R | 0 | SCAP_STA<br>Still capture in progress<br>Indicates the CSI is capturing still image data (single frame). The bit is set at the start of the first frame after enabling still frame capture. It clears itself after the last pixel of the first frame is captured.<br>For CCIR656 interface, if the output format is frame planar YCbCr 420 mode, the frame end means the field2 end, the other frame end means filed end. |

### 5.2.4.15. CSI Channel_0 interrupt enable Register (Default Value: 0x00000000)

| Offset: 0x0070 | | | Register Name: **CSI0_C0_INT_EN_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:08 | / | / | / |
| 07 | R/W | 0 | VS_INT_EN<br>vsync flag<br>The bit is set when vsync come. And at this time load the buffer address for the coming frame. So after this irq come, change the buffer address could only effect next frame |
| 06 | R/W | 0 | HB_OF_INT_EN<br>Hblank FIFO overflow<br>The bit is set when 3 FIFOs still overflow after the hblank. |
| 05 | R/W | 0 | MUL_ERR_INT_EN<br>Multi-channel writing error<br>Indicates error has been detected for writing data to a wrong channel. |
| 04 | R/W | 0 | FIFO2_OF_INT_EN<br>FIFO 2 overflow<br>The bit is set when the FIFO 2 become overflow. |
| 03 | R/W | 0 | FIFO1_OF_INT_EN<br>FIFO 1 overflow<br>The bit is set when the FIFO 1 become overflow. |
| 02 | R/W | 0 | FIFO0_OF_INT_EN<br>FIFO 0 overflow<br>The bit is set when the FIFO 0 become overflow. |
| 01 | R/W | 0 | FD_INT_EN<br>Frame done<br>Indicates the CSI has finished capturing an image frame. Applies to video capture mode. The bit is set after each completed frame capturing data is wrote to buffer as long as video capture remains enabled. |
| 00 | R/W | 0 | CD_INT_EN<br>Capture done<br>Indicates the CSI has completed capturing the image data.<br>For still capture, the bit is set when one frame data has been wrote to buffer.<br>For video capture, the bit is set when the last frame has been wrote to buffer after video capture has been disabled.<br>For CCIR656 interface, if the output format is frame planar YCbCr 420 mode, the frame end means the field2 end, the other frame end means field end. |

### 5.2.4.16. CSI Channel_0 interrupt status Register (Default Value: 0x00000000)

| Offset: 0x0074 | Register Name: **CSI0_C0_INT_STA_REG** |
|---|---|

| Bit | R/W | Default/Hex | Description |
|-----|-----|-------------|-------------|
| 31:08 | / | / | / |
| 07 | R/W | 0 | VS_PD<br>vsync flag |
| 06 | R/W | 0 | HB_OF_PD<br>Hblank FIFO overflow |
| 05 | R/W | 0 | MUL_ ERR_PD<br>Multi-channel writing error |
| 04 | R/W | 0 | FIFO2_OF_PD<br>FIFO 2 overflow |
| 03 | R/W | 0 | FIFO1_OF_PD<br>FIFO 1 overflow |
| 02 | R/W | 0 | FIFO0_OF_PD<br>FIFO 0 overflow |
| 01 | R/W | 0 | FD_PD<br>Frame done |
| 00 | R/W | 0 | CD_PD<br>Capture done |

### 5.2.4.17. CSI Channel_0 horizontal size Register (Default Value: 0x05000000)

| Offset: 0x0080 | | | Register Name: **CSI0_C0_HSIZE_REG** |
|-----|-----|-------------|-------------|
| Bit | R/W | Default/Hex | Description |
| 31:29 | / | / | / |
| 28:16 | R/W | 500 | HOR_LEN<br>Horizontal pixel unit length. Valid pixel of a line. |
| 15:13 | / | / | / |
| 12:00 | R/W | 0 | HOR_START<br>Horizontal pixel unit start. Pixel is valid from this pixel. |

### 5.2.4.18. CSI Channel_0 vertical size Register (Default Value: 0x01E00000)

| Offset: 0x0084 | | | Register Name: **CSI0_C0_VSIZE_REG** |
|-----|-----|-------------|-------------|
| Bit | R/W | Default/Hex | Description |
| 31:29 | / | / | / |
| 28:16 | R/W | 1E0 | VER_LEN<br>Vertical line length. Valid line number of a frame. |
| 15:13 | / | / | / |
| 12:00 | R/W | 0 | VER_START<br>Vertical line start. data is valid from this line. |

### 5.2.4.19. CSI Channel_0 buffer length Register (Default Value: 0x01400280)

| Offset: 0x0088 | | | Register Name: **CSI0_C0_BUF_LEN_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:30 | / | / | / |
| 29:16 | R/W | 140 | BUF_LEN_C<br>Buffer length of chroma C in a line. Unit is byte. |
| 15:14 | / | / | / |
| 13:00 | R/W | 280 | BUF_LEN<br>Buffer length of luminance Y in a line. Unit is byte. |

### 5.2.4.20. CSI Channel_0 flip size Register (Default Value: 0x01E00280)

| Offset: 0x008C | | | Register Name: **CSI0_C0_FLIP_SIZE_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:29 | / | / | / |
| 28:16 | R/W | 1E0 | VER_LEN<br>Vertical line number when in vflip mode. |
| 15:13 | / | / | / |
| 12:00 | R/W | 280 | VALID_LEN<br>Valid components of a line when in flip mode. |

### 5.2.4.21. CSI Channel_0 frame clock counter Register (Default Value: 0x00000000)

| Offset: 0x0090 | | | Register Name: **CSI0_C0_FRM_CLK_CNT_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:24 | / | / | / |
| 23:00 | R | 0 | FRM_CLK_CNT<br>Counter value between every frame. For instant hardware frame rate statics.<br>The internal counter is added by one every 24MHz clock cycle. When frame done or vsync comes, the internal counter value is sampled to FRM_CLK_CNT, and cleared to 0. |

### 5.2.4.22. CSI Channel_0 accumulated and internal clock counter Register (Default Value: 0x00000000)

| Offset: 0x0094 | | | Register Name: **CSI0_C0_ACC_ITNL_CLK_CNT_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:24 | R | 0 | ACC_CLK_CNT<br>The accumulated value of FRM_CLK_CNT for software frame rate statics. |

| | | | Every interrupt of frame done, the software check this accumulated value and clear it to 0. If the ACC_CLK_CNT is larger than 1, the software has lost frame. |
| | | | When frame done or vsync comes, ACC_CLK_CNT = ACC_CLK_CNT + 1, and cleared to 0 when writing 0 to this register. |
| 23:00 | R | 0 | ITNL_CLK_CNT |
| | | | The instant value of internal frame clock counter. |
| | | | When frame done interrupt comes, the software can query this counter for judging whether it is the time for updating the double buffer address registers. |

### 5.2.4.23. CSI Channel_0 FIFO Statistic Register (Default Value: 0x00000000)

| Offset: 0x0098 | | | Register Name: **CSI0_C0_FIFO_STAT_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:12 | / | / | / |
| 11:00 | R | 0 | FIFO_FRM_MAX |
| | | | Indicates the maximum depth of FIFO being occupied for whole frame. |
| | | | Update at every vsync or framedone. |

### 5.2.4.24. CSI Channel_0 PCLK Statistic Register (Default Value: 0x00007FFF)

| Offset: 0x009C | | | Register Name: **CSI0_C0_PCLK_STAT_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | / | / | / |
| 30:16 | R | 0 | PCLK_CNT_LINE_MAX |
| | | | Indicates maximum pixel clock counter value for each line. |
| | | | Update at every vsync or framedone. |
| 15 | / | / | / |
| 14:00 | R | 0x7fff | PCLK_CNT_LINE_MIN |
| | | | Indicates minimum pixel clock counter value for each line. |
| | | | Update at every vsync or framedone. |

### 5.2.4.25. CCI Control Register (Default Value: 0x00000000)

| Offset: 0x3000 | | | Register Name: **CCI_CTRL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0 | SINGLE_TRAN |
| | | | 0: Transmission idle |
| | | | 1: Start single transmission |
| | | | Automatically cleared to '0' when finished. Abort current transmission |

| | | | |
|---|---|---|---|
| | | | immediately if changing from '1' to '0'. If slave not respond for the expected status over the time defined by TIMEOUT, current transmission will stop. PACKET_CNT will return the sequence number when transmission fail. All format setting and data will be loaded from registers and FIFO when transmission start. |
| 30 | R/W | 0 | REPEAT_TRAN<br>0: transmission idle<br>1: repeated transmission<br>When this bit is set to 1, transmission repeats when trigger signal (such as VSYNC/ VCAP done ) repeats.<br>If changing this bit from '1' to '0' during transmission, the current transmission will be guaranteed then stop. |
| 29 | R/W | 0 | RESTART_MODE<br>0: RESTART<br>1: STOP+START<br>Define the CCI action after sending register address. |
| 28 | R/W | 0 | READ_TRAN_MODE<br>0: send slave_id+W<br>1: do not send slave_id+W<br>**Setting this bit to 1 if reading from a slave which register width is equal to 0.** |
| 27:24 | R | 0 | TRAN_RESULT<br>000: OK<br>001: FAIL<br>Other: Reserved |
| 23:16 | R | / | CCI_STA<br>0x00: bus error<br>0x08: START condition transmitted<br>0x10: Repeated START condition transmitted<br>0x18: Address + Write bit transmitted, ACK received<br>0x20: Address + Write bit transmitted, ACK not received<br>0x28: Data byte transmitted in master mode, ACK received<br>0x30: Data byte transmitted in master mode, ACK not received<br>0x38: Arbitration lost in address or data byte<br>0x40: Address + Read bit transmitted, ACK received<br>0x48: Address + Read bit transmitted, ACK not received<br>0x50: Data byte received in master mode, ACK received<br>0x58: Data byte received in master mode, ACK not received<br>0x01: Timeout when sending 9th SCL clk<br>Other: Reserved |
| 15:2 | / | / | / |
| 1 | R/W | 0 | SOFT_RESET<br>0: normal<br>1: reset |
| 0 | R/W | 0 | CCI_EN |

| | | | 0: Module disable |
| | | | 1: Module enable |

### 5.2.4.26. CCI Transmission Configuration Register (Default Value: 0x10000000)

| Offset: 0x3004 | | | Register Name: **CCI_CFG_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:24 | R/W | 0x10 | TIMEOUT_N<br>When sending the 9th clock, assert fail signal when slave device did not response after N*FSCL cycles. And software must do a reset to CCI module and send a stop condition to slave. |
| 23:16 | R/W | 0x00 | INTERVAL<br>Define the interval between each packet in 40*FSCL cycles. 0~255 |
| 15 | R/W | 0 | PACKET_MODE<br>Select where to load slave id / data width<br>0: Compact mode<br>1: Complete mode<br>In compact mode, slave id/register width / data width will be loaded from CCI_FMT register, only address and data read from memory.<br>In complete mode, they will be loaded from packet memory. |
| 14:7 | / | / | / |
| 6:4 | R/W | 0 | TRIG_MODE<br>Transmit mode:<br>000: Immediately, no trigger<br>001: Reserved<br>010: CSI0 int trigger<br>011: CSI1 int trigger |
| 3:0 | R/W | 0 | CSI_TRIG<br>CSI Int trig signal select:<br>0000: First HREF start<br>0001: Last HREF done<br>0010: Line counter trigger<br>other: Reserved |

### 5.2.4.27. CCI Packet Format Register (Default Value: 0x00110001)

| Offset: 0x3008 | | | Register Name: **CCI_FMT_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:25 | R/W | 0 | SLV_ID<br>7bit address |
| 24 | R/W | 0 | CMD<br>0: write |

| | | | 1: read |
|---|---|---|---|
| 23:20 | R/W | 1 | ADDR_BYTE<br>How many bytes be sent as address<br>0~15 |
| 19:16 | R/W | 1 | DATA_BYTE<br>How many bytes be sent/received as data<br>1~15<br>Normally use ADDR_DATA with 0_2, 1_1, 1_2, 2_1, 2_2 access mode. If DATA bytes is 0, transmission will not start. In complete mode, the ADDR_BYTE and DATA_BYTE is defined in a byte's high/low 4bit. |
| 15:0 | R/W | 1 | PACKET_CNT<br>FIFO data be transmitted as PACKET_CNT packets in current format.<br>Total bytes not exceed 32bytes. |

### 5.2.4.28. CCI Bus Control Register (Default Value: 0x00002500)

| Offset: 0x300C | | | Register Name: **CCI_BUS_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:16 | R/W | 0 | DLY_CYC<br>0~65535 FSCL cycles between each transmission |
| 15 | R/W | 0 | DLY_TRIG<br>0: disable<br>1: execute transmission after internal counter delay when triggered |
| 14:12 | R/W | 0x2 | CLK_N<br>CCI bus sampling clock F0=24MHz/2^CLK_N |
| 11:8 | R/W | 0x5 | CLK_M<br>CCI output SCL frequency is FSCL=F1/10=(F0/(CLK_M+1))/10 |
| 7 | R | / | SCL_STA<br>SCL current status |
| 6 | R | / | SDA_STA<br>SDA current status |
| 5 | R/W | 0 | SCL_PEN<br>SCL PAD enable |
| 4 | R/W | 0 | SDA_PEN<br>SDA PAD enable |
| 3 | R/W | 0 | SCL_MOV<br>SCL manual output value |
| 2 | R/W | 0 | SDA_MOV<br>SDA manual output value |
| 1 | R/W | 0 | SCL_MOE<br>SCL manual output en |
| 0 | R/W | 0 | SDA_MOE<br>SDA manual output en |

### 5.2.4.29. CCI Interrupt Control Register (Default Value: 0x00000000)

| Offset: 0x3014 | | | Register Name: **CCI_INT_CTRL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:18 | / | / | / |
| 17 | R/W | 0 | S_TRAN_ERR_INT_EN |
| 16 | R/W | 0 | S_TRAN_COM_INT_EN |
| 15:2 | / | / | / |
| 1 | R/W | 0 | S_TRAN_ERR_PD |
| 0 | R/W | 0 | S_TRAN_COM_PD |

### 5.2.4.30. CCI Line Counter Trigger Control Register (Default Value: 0x00000000)

| Offset: 0x3018 | | | Register Name: **CCI_LC_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:13 | / | / | / |
| 12:0 | R/W | 0 | LN_CNT<br>0~8191: line counter send trigger when 1st~8192th line is received. |

### 5.2.4.31. CCI FIFO Acess Register (Default Value: 0x00000000)

| Offset: 0x3100~0x313f | | | Register Name: **CCI_FIFO_ACC_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R/W | 0 | DATA_FIFO<br>From 0x100 to 0x13f, CCI data fifo is 64bytes, used in fifo input mode. CCI transmission read/write data from/to fifo in byte. |

# Chapter 6  Display

This chapter describes the A83T display system from following perspectives:

- Display Engine2.0
- TCON

## 6.1.  Display System

Figure 6-1 shows the block diagram of display system.
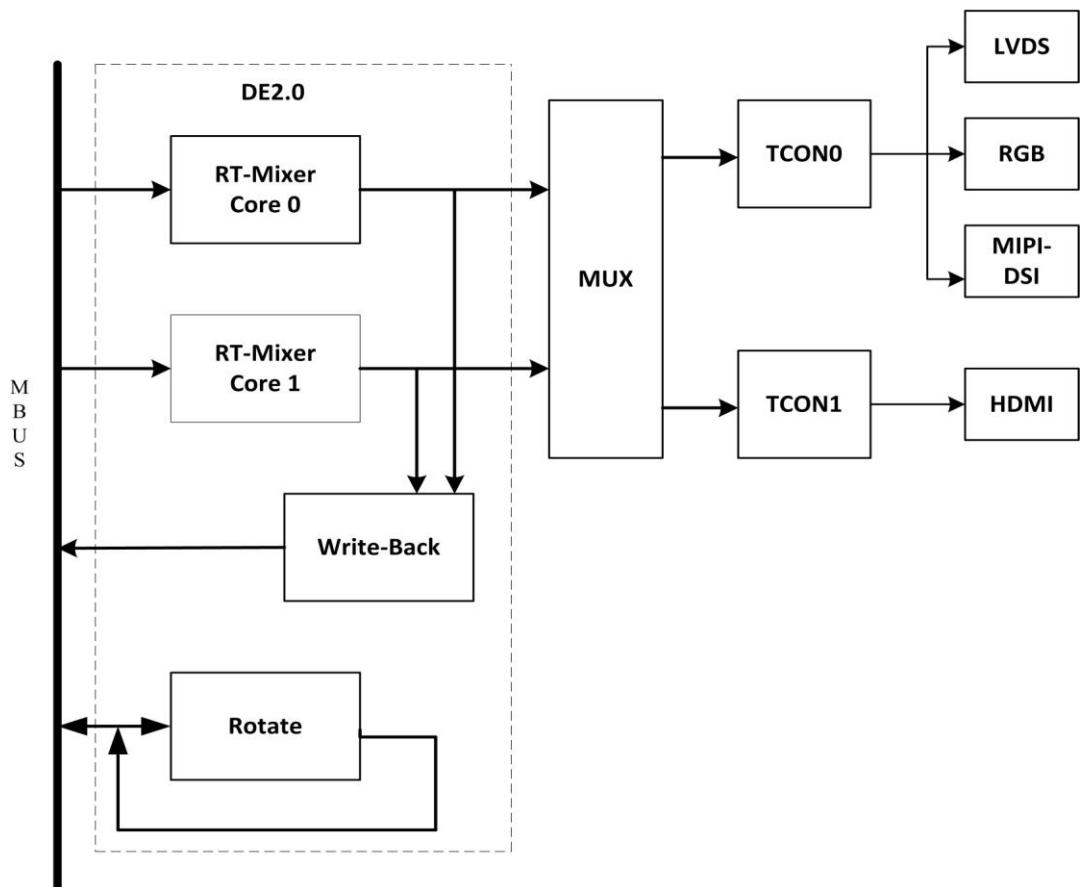


Figure 6-1. Dispalay System Block Diagram

## 6.2.    Display Engine2.0

### 6.2.1.    Overview

- Support input layer size up to 2048x2048, and output size up to 2048x2048.

- Support four alpha blending channel for main display,two channel for aux display.

- Support four overlay layer in each channel, and has a independent scaler.

- Support potter-duff compatible blending operation.

- Support input format YUV422/YUV420/YUV411/ARGB8888/XRGB8888/RGB888/ARGB4444/ARGB1555 and RGB565

- Support Frame Packing/Top-and-Bottom/Side-by-Side Full/Side-by-Side Half 3D format data

- Support display enhancement 2.0 for excellent display experience

  - Adaptive edge sharping
  - Adaptive color enhancement
  - Adaptive contrast enhancement and fresh tone rectify

- Support write back & rotation for high efficient dual display and miracast.

## 6.3.    TCON

### 6.3.1.    Overview

The TCON0 module is used for LCD panel, and TCON1 module is used for HDTV.

- Support LVDS interface with single link, up to 1280x800@60fps
- Support RGB interface with DE/SYNC mode, up to 1920x1200@60fps
- Support serial RGB/dummy RGB/CCIR656 interface, up to 1280x800@60fps
- Support i80 interface with 18/16/9/8 bit, TE supported, up to 1280x800@60fps
- Support pixel format: RGB888, RGB666 and RGB565
- Dither function from RGB666/RGB565 to RGB888
- Gamma correction with R/G/B channel independence
- 4 interrupts for programmer single LCD output
- Support 3D function

## 6.3.2. Block Diagram
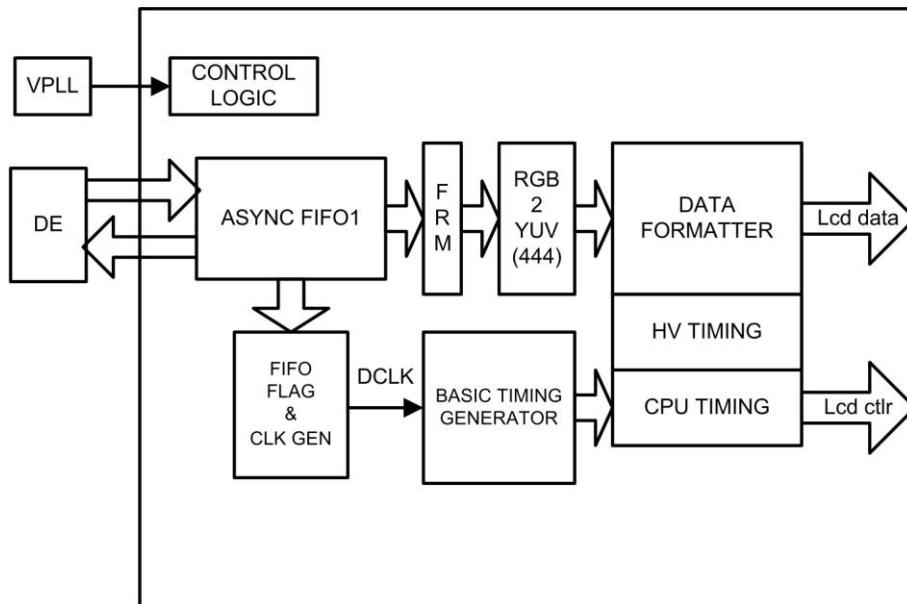


Figure 6-2. TCON0 Block Diagram
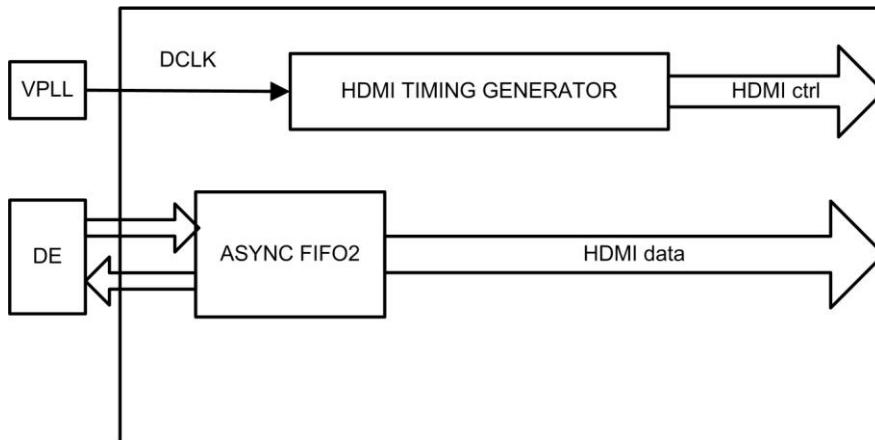


Figure 6-3. TCON1 Block Diagram

## 6.3.3. Functionalities Description

### 6.3.3.1. Panel Interface

**HV_I/F(Sync+DE mode)**

HV I/F is also known as Sync + DE mode, which is widely used in TFT LCD module for PMP/MP4 applications.
Its signals are define as:

| Main Signal | I/O type | Definition And Description |
|---|---|---|
| Vsync | O | Vertical sync, indicates one new frame |
| Hsync | O | Horizontal sync, indicate one new scan line |
| DCLK | O | Dot clock, pixel data are sync by this clock |
| LDE | O | LCD data enable |
| LD[23..0] | O | 24Bit RGB/YUV output from input FIFO for panel |

HV control signals are active low.
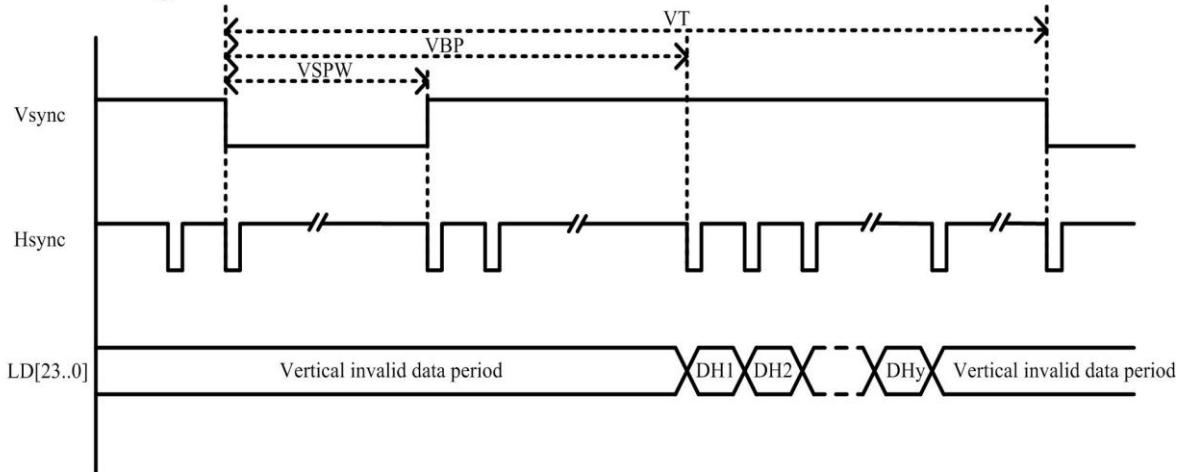


Figure 6-4. Panel Interface Odd/Even Field Timing
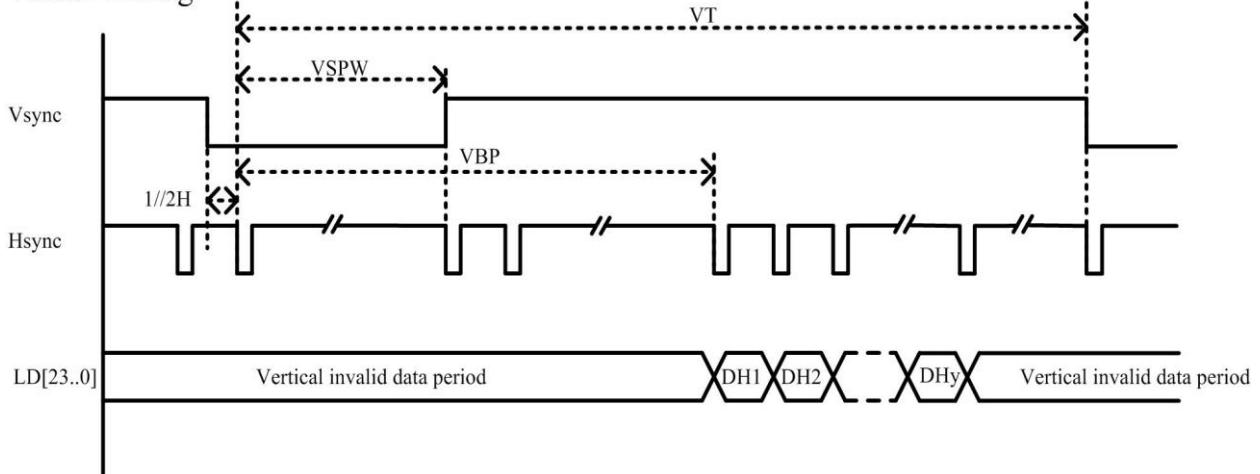


Figure 6-5. Panel Interface Even Filed Timing
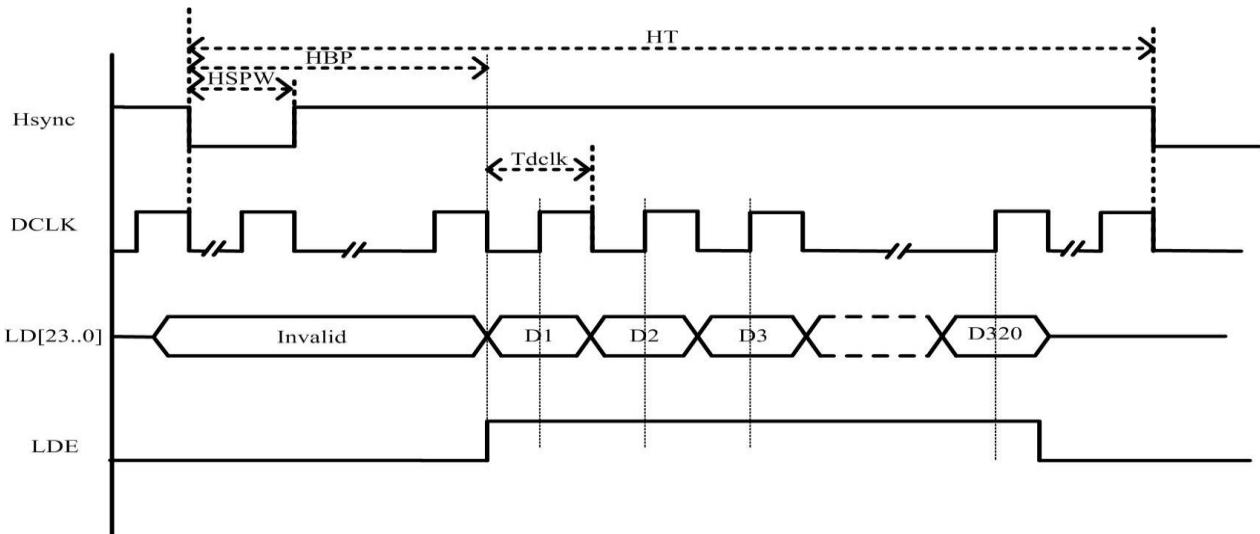
Figure 6-6. Parallel Mode Horizontal Timing
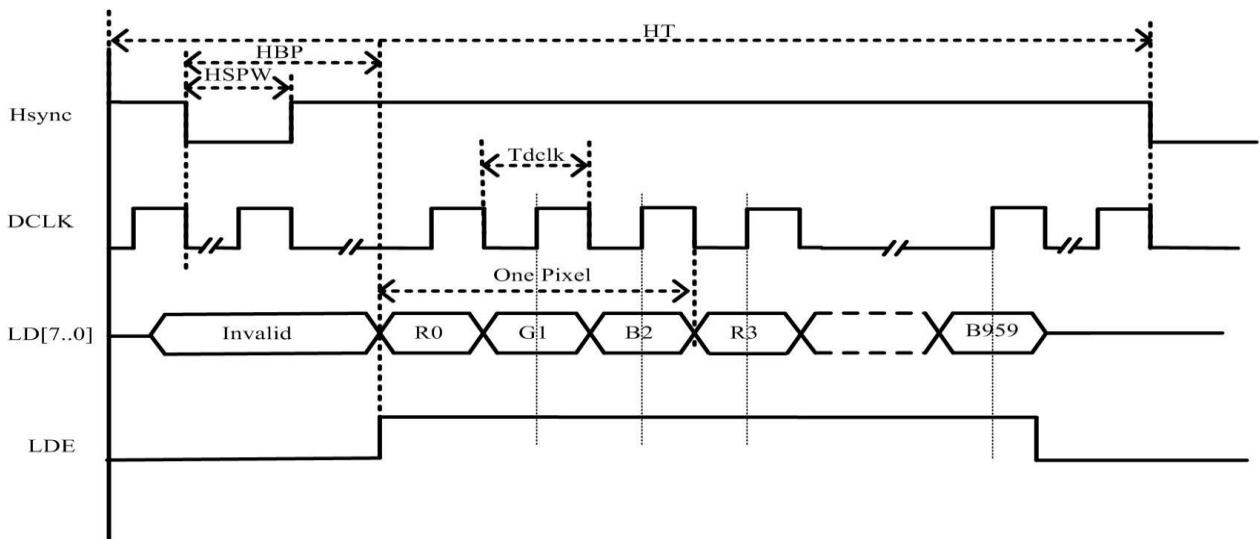


Figure 6-7. Serial Mode Horizontal Timing

**CCIR output SAV/EAV sync signal**

When in HV serial YUV output mode, it's timing is CCIR656/601 compatible. SAV add right before active area every line; EAV add right after active area every line.

It logic are:

F = "0" for Field 1 F = "1" for Field 2

V = "1" during vertical blanking

H = "0" at SAV H = "1" at EAV

P3–P0 = protection bits

P3 = V $\oplus$ H

P2 = F $\oplus$ H

P1 = F $\oplus$ V

P0 = F $\oplus$ V $\oplus$ H

Where $\oplus$ represents the exclusive-OR function

The 4 byte SAV/EAV sequences are:

| | 8-bit Data | | | | | | | | 10-bit Data | |
|---|---|---|---|---|---|---|---|---|---|---|
| | D9 (MSB) | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| preamble | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Status word | 1 | F | V | H | P3 | P2 | P1 | P0 | 0 | 0 |

**CPU_I/F**

CPU I/F LCD panel is most common interface for small size, low resolution LCD panels.

CPU control signals are active low.

| Main Signal | I/O type | Definition And Description |
|---|---|---|
| CS | O | Chip select, active low |
| WR | O | Write strobe, active low |
| RD | O | Read strobe, active low |
| A1 | O | Address bit, controlled by "LCD_CPUI/F" BIT26/25 |
| D[23..0] | I/O | Digital RGB output signal |

The following figure relationship between basic timing and CPU timing.

WR is 180 degree delay of DCLK; CS is active when pixel data are valid;

RD is always set to 1; A1 are set by "**Lcd_CPUI/F**".
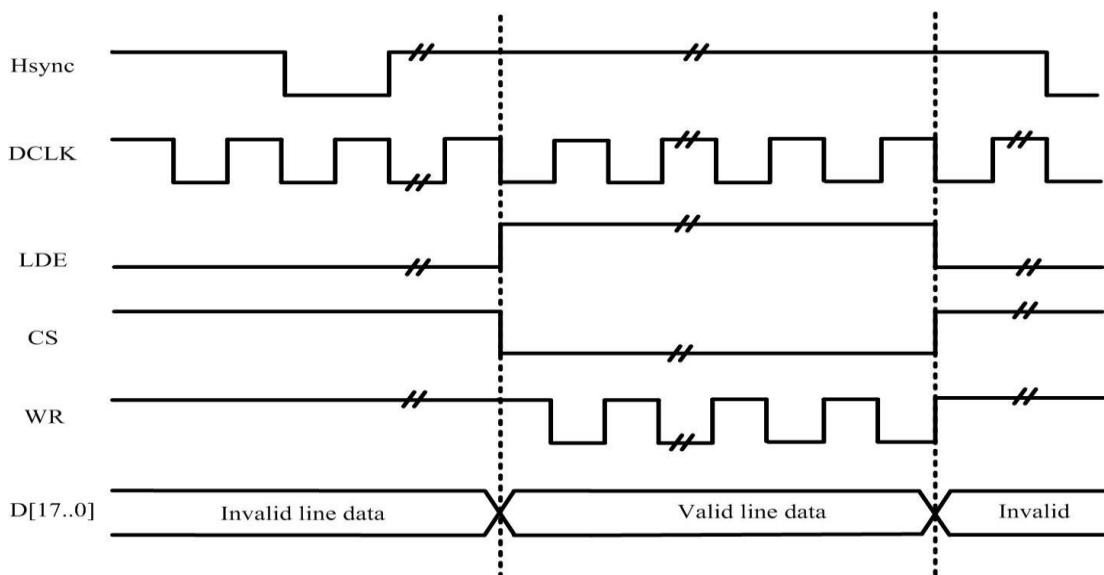


Figure 6-8. Lcd_CPUI/F Timing

When CPU I/F is in IDLE state, it can generate WR/RD timing by setting "**Lcd_CPUI/F**". CS strobe is one DCLK width, WR/RD strobe is half DCLK width.

**LVDS_IF**
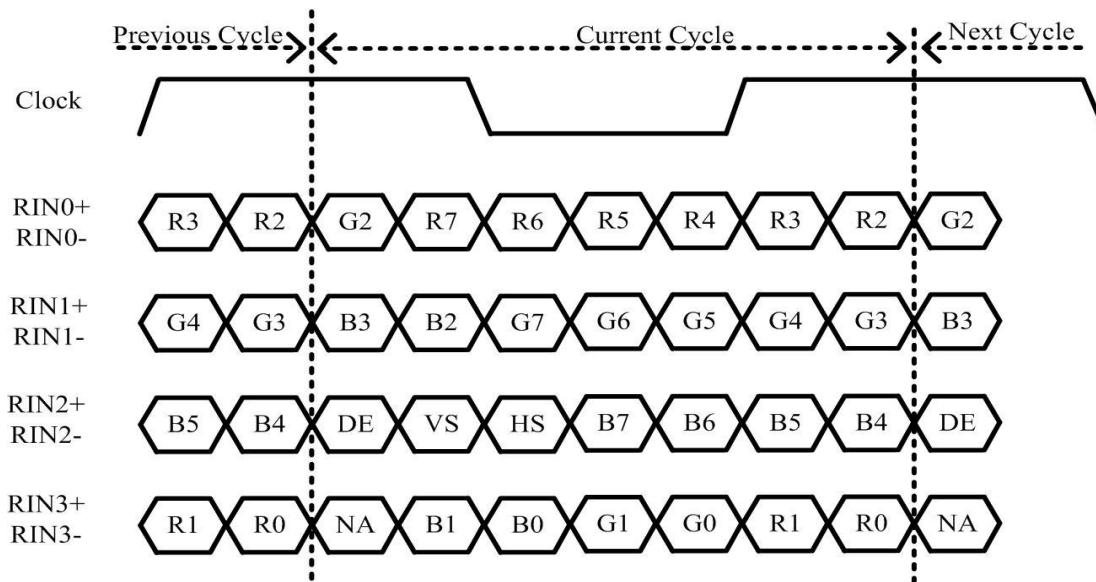
Figure 6-9. JEDIA Mode Timing
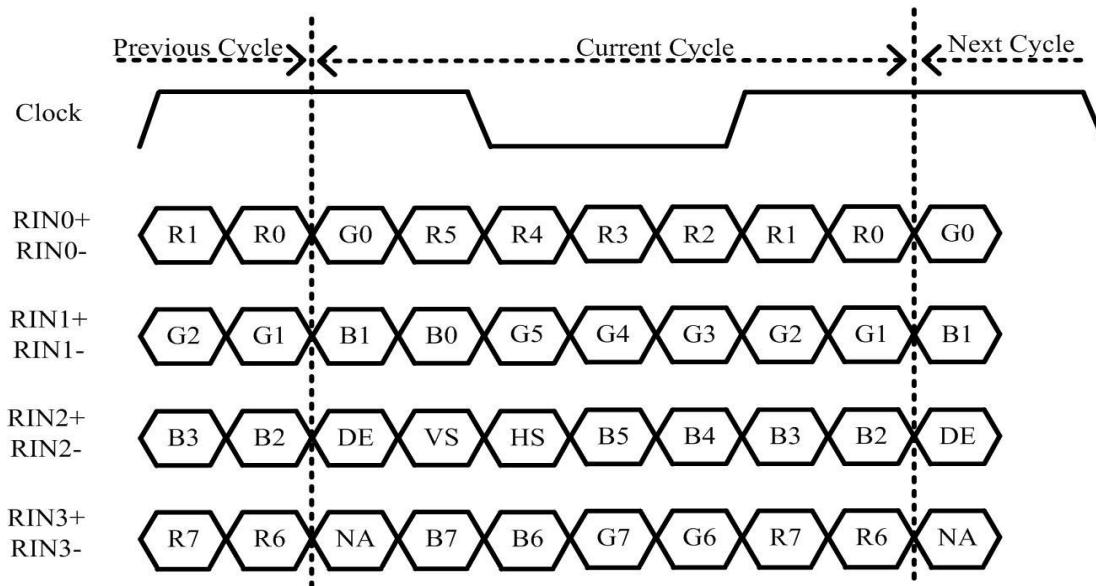


Figure 6-10. NS Mode Timing

### 6.3.3.2. RGB　gamma correction

Function: This module correct the RGB input data of DE0 .

A 256*8*3 Byte register file is used to store the gamma table. The following is the layout:

| Offset | Value |
|---|---|
| 0x400, 0x401, 0x402 | { B0[7:0], G0[7:0], R0[7:0] } |
| 0x404, | { B1[7:0], G1[7:0], R1[7:0] } |
| ...... | ...... |
| 0x4FC | { B255[7:0], G255[7:0], R255[7:0] } |

### 6.3.3.3. CEU module

Function: This module enhance color data from DE0 .

R' = Rr*R + Rg*G + Rb*B + Rc

G' = Gr*R + Gg*G + Gb*B +Gc

B' = Br*R + Bg*G + Bb*B + Bc

Note:

Rr, Rg, Rb, ,Gr, Gg, Gb, Br, Bg, Bb       s13 (-16,16)

Rc, Gc, Bc                                                s19 (-16384, 16384)

R, G, B                                                     u8   [0-255]

R' have the range of    [Rmin ,Rmax]
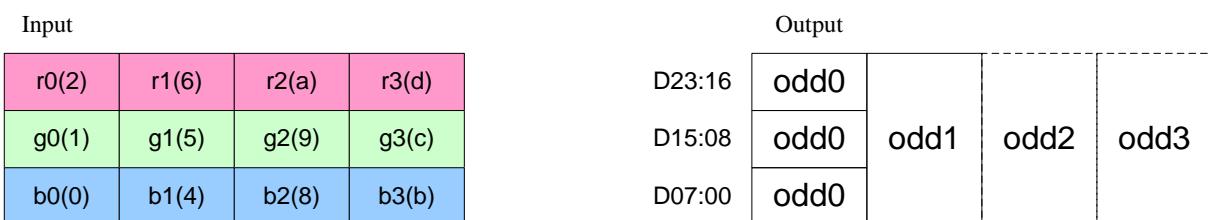
G' have the range of    [Rmin ,Rmax]

B' have the range of    [Rmin ,Rmax]

### 6.3.3.4. CMAP module
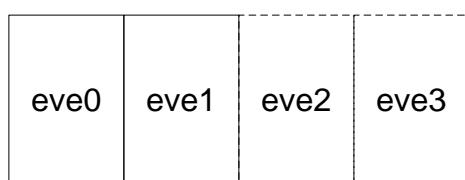
Function: This module map color data from DE0

Every 4 input pixels as an unit. an unit is divided into 12 bytes. Output byte can select one of those 12 bytes.Note that even line and odd line can be different, and output can be 12 bytes(4 pixels) or reduce to 6 bytes(2 pixels).

Input

| r0(2) | r1(6) | r2(a) | r3(d) |
| g0(1) | g1(5) | g2(9) | g3(c) |
| b0(0) | b1(4) | b2(8) | b3(b) |

Output

| | | | |
|---|---|---|---|
| D23:16 odd0 | | | |
| D15:08 odd0 | odd1 | odd2 | odd3 |
| D07:00 odd0 | | | |

| eve0 | eve1 | eve2 | eve3 |

In mode:    4 pixels
Out mode: 4 pixels/2 pixels

## 6.3.4.   TCON0 Module Register List

| Module Name | Base Address |
|---|---|
| TCON0 | 0x01C0C000 |

| Register Name | Offset | Description |
|---|---|---|
| TCON_GCTL_REG | 0x000 | TCON global control register |
| TCON_GINT0_REG | 0x004 | TCON global interrupt register0 |

| TCON_GINT1_REG | 0x008 | TCON global interrupt register1 |
|---|---|---|
| TCON0_FRM_CTL_REG | 0x010 | TCON FRM control register |
| TCON0_FRM_SEED_REG | 0x014+N*0x04 | TCON FRM seed register<br><br>(N=0,1,2,3,4,5) |
| TCON0_FRM_TAB_REG | 0x02C+N*0x04 | TCON FRM table register<br><br>(N=0,1,2,3) |
| TCON0_3D_FIFO_REG | 0x03C | TCON0 3D fifo register |
| TCON0_CTL_REG | 0x040 | TCON0 control register |
| TCON0_DCLK_REG | 0x044 | TCON0 data clock register |
| TCON0_BASIC0_REG | 0x048 | TCON0 basic timing register0 |
| TCON0_BASIC1_REG | 0x04C | TCON0 basic timing register1 |
| TCON0_BASIC2_REG | 0x050 | TCON0 basic timing register2 |
| TCON0_BASIC3_REG | 0x054 | TCON0 basic timing register3 |
| TCON0_HV_IF_REG | 0x058 | TCON0 hv panel interface register |
| TCON0_CPU_IF_REG | 0x060 | TCON0 cpu panel interface register |
| TCON0_CPU_WR_REG | 0x064 | TCON0 cpu panel write data register |
| TCON0_CPU_RD0_REG | 0x068 | TCON0 cpu panel read data register0 |
| TCON0_CPU_RD1_REG | 0x06C | TCON0 cpu panel read data register1 |
| TCON0_LVDS_IF_REG | 0x084 | TCON0 lvds panel interface register |
| TCON0_IO_POL_REG | 0x088 | TCON0 IO polarity register |
| TCON0_IO_TRI_REG | 0x08C | TCON0 IO control register |
| TCON_DEBUG_REG | 0x0FC | TCON debug register |
| TCON_CEU_CTL_REG | 0x100 | TCON CEU control register |
| TCON_CEU_COEF_MUL_REG | 0x110+N*0x04 | TCON CEU coefficient register0<br><br>(N=0,1,2,4,5,6,8,9,10) |
| TCON_CEU_COEF_ADD_REG | 0x11C+N*0x10 | TCON CEU coefficient register1<br><br>(N=0,1,2) |
| TCON_CEU_COEF_RANG_REG | 0x140+N*0x04 | TCON CEU coefficient register2<br><br>(N=0,1,2) |
| TCON0_CPU_TRI0_REG | 0x160 | TCON0 cpu panel trigger register0 |
| TCON0_CPU_TRI1_REG | 0x164 | TCON0 cpu panel trigger register1 |
| TCON0_CPU_TRI2_REG | 0x168 | TCON0 cpu panel trigger register2 |
| TCON0_CPU_TRI3_REG | 0x16C | TCON0 cpu panel trigger register3 |
| TCON0_CPU_TRI4_REG | 0x170 | TCON0 cpu panel trigger register4 |
| TCON0_CPU_TRI5_REG | 0x174 | TCON0 cpu panel trigger register5 |
| TCON_CMAP_CTL_REG | 0x180 | TCON color map control register |
| TCON_CMAP_ODD0_REG | 0x190 | TCON color map odd line register0 |
| TCON_CMAP_ODD1_REG | 0x194 | TCON color map odd line register1 |
| TCON_CMAP_EVEN0_REG | 0x198 | TCON color map even line register0 |
| TCON_CMAP_EVEN1_REG | 0x19C | TCON color map even line register1 |

| TCON_SAFE_PERIOD_REG | 0x1F0 | TCON safe period register |
|---|---|---|
| TCON0_LVDS_ANA0_REG | 0x220 | TCON LVDS analog register0 |
| TCON_GAMMA_TABLE_REG | 0x400-0x7FF | |
| TCON0_3D_FIFO_BIST_REG | 0xFF4 | |
| TCON_TRI_FIFO_BIST_REG | 0xFF8 | |
| TCON_ECC_FIFO_BIST_REG | 0xFFC | |

## 6.3.5.  TCON0 Module Register Description

### 6.3.5.1.  TCON_GCTL_REG

| Offset: 0x0000 | | | Register Name: **TCON_GCTL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0 | TCON_En<br>0: disable<br>1: enable<br>When it's disabled, the module will be reset to idle state. |
| 30 | R/W | 0 | TCON_Gamma_En<br>0: disable<br>1: enable |
| 29:0 | / | / | / |

### 6.3.5.2.  TCON_ GINT0_REG

| Offset: 0x0004 | | | Register Name: **TCON_GINT0_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0 | TCON0_Vb_Int_En<br>0: disable<br>1: enable |
| 30 | / | / | / |
| 29 | R/W | 0 | TCON0_Line_Int_En<br>0: disable<br>1: enable |
| 28 | / | / | / |
| 27 | R/W | 0 | TCON0_Tri_Finish_Int_En<br>0: disable<br>1: enable |
| 26: | R/W | 0 | TCON0_Tri_Counter_Int_En |

| | | | 0: disable |
|---|---|---|---|
| | | | 1: enable |
| 25:16 | / | / | / |
| 15 | R/W | 0 | TCON0_Vb_Int_Flag |
| | | | Asserted during vertical no-display period every frame. |
| | | | Write 0 to clear it. |
| 14 | / | / | / |
| 13 | R/W | 0 | TCON0_Line_Int_Flag |
| | | | trigger when SY0 match the current TCON0 scan line |
| | | | Write 0 to clear it. |
| 12 | / | / | / |
| 11 | R/W | 0 | TCON0_Tri_Finish_Int_Flag |
| | | | trigger when cpu trigger mode finish |
| | | | Write 0 to clear it. |
| 10 | R/W | 0 | TCON0_Tri_Counter_Int_Flag |
| | | | trigger when tri counter reach this value |
| | | | Write 0 to clear it. |
| 9 | R/W | 0 | TCON0_Tri_Underflow_Flag |
| | | | only used in dsi video mode, tri when sync by dsi but not finish |
| | | | Write 0 to clear it. |
| 8:0 | / | / | / |

### 6.3.5.3.  TCON_ GINT1_REG

| Offset: 0x0008 | | | Register Name: **TCON_GINT1_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:28 | / | / | / |
| 27:16 | R/W | 0 | TCON0_Line_Int_Num |
| | | | scan line for TCON0 line trigger(including inactive lines) |
| | | | Setting it for the specified line for trigger0. |
| | | | Note: SY0 is writable only when LINE_TRG0 disable. |
| 15:12 | / | / | / |
| 11:0 | / | / | / |

### 6.3.5.4.  TCON_FRM_CTL_REG

| Offset: 0x0010 | | | Register Name: **TCON_FRM_CTL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0 | TCON0_Frm_En |
| | | | 0:disable |
| | | | 1:enable |
| 30:7 | / | / | / |

| 6 | R/W | 0 | TCON0_Frm_Mode_R |
|---|---|---|---|
| | | | 0: 6bit frm output |
| | | | 1: 5bit frm output |
| 5 | R/W | 0 | TCON0_Frm_Mode_G |
| | | | 0: 6bit frm output |
| | | | 1: 5bit frm output |
| 4 | R/W | 0 | TCON0_Frm_Mode_B |
| | | | 0: 6bit frm output |
| | | | 1: 5bit frm output |
| 3:2 | / | / | / |
| 1:0 | R/W | 0 | TCON0_Frm_Test |
| | | | 00: FRM |
| | | | 01: half 5/6bit, half FRM |
| | | | 10: half 8bit, half FRM |
| | | | 11: half 8bit, half 5/6bit |

#### 6.3.5.5. TCON0_FRM_SEED_REG

| Offset: 0x014+N*0x04(N=0,1,2,3,4,5) | | | Register Name: **TCON0_FRM_SEED_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:25 | / | / | / |
| 24:0 | R/W | 0 | Seed_Value |
| | | | N=0: Pixel_Seed_R |
| | | | N=1: Pixel_Seed_G |
| | | | N=2: Pixel_Seed_B |
| | | | N=3: Line_Seed_R |
| | | | N=4: Line_Seed_G |
| | | | N=5: Line_Seed_B |
| | | | Note: avoid set it to 0 |

#### 6.3.5.6. TCON0_FRM_TAB_REG

| Offset: 0x02C+N*0x04(N=0,1,2,3) | | | Register Name: **TCON0_FRM_TAB_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R/W | 0 | Frm_Table_Value |

#### 6.3.5.7. TCON0_3D_FIFO_REG

| Offset: 0x03C | | | Register Name: **TCON0_3D_FIFO_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0 | 3D_FIFO_BIST_EN |

| | | | 0: disable |
| | | | 1: enable |
| 30:14 | / | / | / |
| 13:4 | R/W | 0 | 3D_FIFO_HALF_LINE_SIZE <br> Note: <br> The number of data in half line=3D_FIFO_HALF_LINE_SIZE+1 <br> only valid when 3D_FIFO_SETTING set as 2 |
| 3:2 | / | / | / |
| 1:0 | R/W | 0 | 3D_FIFO_SETTING <br> 0: by pass <br> 1: used as normal FIFO <br> 2: used as 3D interlace FIFO <br> 3: reserved |

### 6.3.5.8.  TCON0_CTL_REG

| Offset: 0x040 | | | Register Name: **TCON0_CTL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0 | TCON0_En <br> 0: disable <br> 1: enable <br> Note: It executes at the beginning of the first blank line of TCON0 timing. |
| 30:29 | / | / | / |
| 28 | R/W | 0 | TCON0_Work_Mode <br> 0: normal <br> 1: dynamic freq |
| 27:26 | / | / | / |
| 25:24 | R/W | 0 | TCON0_IF <br> 00: HV(Sync+DE) <br> 01: 8080 I/F <br> 1x:reservd |
| 23 | R/W | 0 | TCON0_RB_Swap <br> 0: default <br> 1: swap RED and BLUE data at FIFO1 |
| 22 | / | / | / |
| 21 | R/W | 0 | TCON0_FIFO1_Rst <br> Write 1 and then 0 at this bit will reset FIFO 1 <br> Note: 1 holding time must more than 1 DCLK |
| 20:9 | / | / | / |
| 8:4 | R/W | 0 | TCON0_Start_Delay <br> STA delay <br> NOTE: valid only when TCON0_EN == 1 |
| 3 | / | / | / |

| 2:0 | R/W | 0 | TCON0_SRC_SEL: |
|-----|-----|---|----------------|
| | | | 000: DE0 |
| | | | 001: reservd |
| | | | 010: reservd |
| | | | 011: reservd |
| | | | 100: Test Data all 0 |
| | | | 101: Test Data all 1 |
| | | | 11x: reservd |

### 6.3.5.9. TCON0_DCLK_REG

| Offset: 0x044 | | | Register Name: **TCON0_DCLK_REG** |
|---------------|-----|--------------|-----------------------------------|
| Bit | R/W | Default/Hex | Description |
| 31:28 | R/W | 0 | TCON0_Dclk_En |
| | | | LCLK_EN[3:0] :TCON0 clock enable |
| | | | 4'h0, 'h4,4'h6,4'ha7:dclk_en=0;dclk1_en=0;dclk2_en=0;dclkm2_en=0; |
| | | | 4'h1: dclk_en = 1; dclk1_en = 0; dclk2_en = 0; dclkm2_en = 0; |
| | | | 4'h2: dclk_en = 1; dclk1_en = 0; dclk2_en = 0; dclkm2_en = 1; |
| | | | 4'h3: dclk_en = 1; dclk1_en = 1; dclk2_en = 0; dclkm2_en = 0; |
| | | | 4'h5: dclk_en = 1; dclk1_en = 0; dclk2_en = 1; dclkm2_en = 0; |
| | | | 4'h8,4'h9,4'ha,4'hb,4'hc,4'hd,4'he,4'hf: |
| | | | dclk_en    = 1; |
| | | | dclk1_en   = 1; |
| | | | dclk2_en   = 1; |
| | | | dclkm2_en = 1; |
| 27:7 | / | / | / |
| 6:0 | R/W | 0 | TCON0_Dclk_Div |
| | | | Tdclk = Tsclk * DCLKDIV |
| | | | Note: |
| | | | 1.if dclk1&dclk2 used，DCLKDIV >=6 |
| | | | 2.if dclk only，DCLKDIV >=1 |

### 6.3.5.10. TCON0_BASIC0_REG

| Offset: 0x048 | | | Register Name: **TCON0_BASIC0_REG** |
|---------------|-----|--------------|-------------------------------------|
| Bit | R/W | Default/Hex | Description |
| 31:28 | / | / | / |
| 27:16 | R/W | 0 | TCON0_X |
| | | | Panel width is X+1 |
| 15:12 | / | / | / |
| 11:0 | R/W | 0 | TCON0_Y |
| | | | Panel height is Y+1 |

### 6.3.5.11. TCON0_BASIC1_REG

| Offset: 0x04C | | | Register Name: **TCON0_BASIC1_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0 | Reservd |
| 30:29 | / | / | / |
| 28:16 | R/W | 0 | HT<br>Thcycle = (HT+1) * Tdclk<br>Computation<br>1) parallel:HT = X + BLANK<br>Limitation:<br>1) parallel :HT >=　(HBP +1) + (X+1) +2<br>2) serial 1: HT >=　(HBP +1) + (X+1) *3+2<br>3) serial 2: HT >=　(HBP +1) + (X+1) *3/2+2 |
| 15:12 | / | / | / |
| 11:0 | R/W | 0 | HBP<br>horizontal back porch (in dclk)<br>Thbp = (HBP +1) * Tdclk |

### 6.3.5.12. TCON0_BASIC2_REG

| Offset: 0x050 | | | Register Name: **TCON0_BASIC2_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:29 | / | / | / |
| 28:16 | R/W | 0 | VT<br>TVT = (VT)/2 * Thsync<br>Note: VT/2 >=　(VBP+1 ) + (Y+1) +2 |
| 15:12 | / | / | / |
| 11:0 | R/W | 0 | VBP<br>Tvbp = (VBP +1) * Thsync |

### 6.3.5.13. TCON0_BASIC3_REG

| Offset: 0x054 | | | Register Name: **TCON0_BASIC3_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:26 | / | / | / |
| 25:16 | R/W | 0 | HSPW<br>Thspw = (HSPW+1) * Tdclk<br>Note: HT>　(HSPW+1) |
| 15:10 | / | / | / |
| 9:0 | R/W | 0 | VSPW |

| | | | Tvspw = (VSPW+1) * Thsync |
|---|---|---|---|
| | | | Note: VT/2 >　(VSPW+1) |

### 6.3.5.14. TCON0_HV_IF_REG

| Offset: 0x058 | | | Register Name: **TCON0_HV_IF_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:28 | R/W | 0 | HV_Mode<br>0000: 24bit/1cycle parallel mode<br>1000: 8bit/3cycle RGB serial mode(RGB888)<br>1010: 8bit/4cycle Dummy RGB(DRGB)<br>1011: 8bit/4cycle RGB Dummy(RGBD)<br>1100: 8bit/2cycle YUV serial mode(CCIR656) |
| 27:26 | R/W | 0 | RGB888_SM0<br>serial RGB888 mode Output sequence at odd lines of the panel (line 1, 3, 5, 7…)<br>00: R→G→B<br>01: B→R→G<br>10: G→B→R<br>11: R→G→B |
| 25:24 | R/W | 0 | RGB888_SM1<br>serial RGB888 mode Output sequence at even lines of the panel (line 2, 4, 6, 8…)<br>00: R→G→B<br>01: B→R→G<br>10: G→B→R<br>11: R→G→B |
| 23:22 | R/W | 0 | YUV_SM<br>serial YUV mode Output sequence 2-pixel-pair of every scan line<br>00: YUYV<br>01: YVYU<br>10: UYVY<br>11: VYUY |
| 21:20 | R/W | 0 | YUV EAV/SAV F line delay<br>0:F toggle right after active video line<br>1:delay 2 line(CCIR NTSC)<br>2:delay 3 line(CCIR PAL)<br>3:reserved |
| 19 | R/W | 0 | CCIR_CSC_Dis<br>0: enable<br>1: disable<br>Only valid when HV mode is "1100",<br>select '0' TCON convert source from RGB to YUV |
| 18:0 | / | / | / |

### 6.3.5.15. TCON0_CPU_IF_REG

| Offset: 0x060 | | | Register Name: **TCON0_CPU_IF_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:28 | R/W | 0 | CPU_Mode<br>0000: 18bit/256K mode<br>0010: 16bit mode0<br>0100: 16bit mode1<br>0110: 16bit mode2<br>1000: 16bit mode3<br>1010: 9bit mode<br>1100: 8bit 256K mode<br>1110: 8bit 65K mode<br>xxx1: 24bit for DSI |
| 27 | / | / | / |
| 26 | R/W | 0 | DA<br>pin A1 value in 8080 mode auto/flash states |
| 25 | R/W | 0 | CA<br>pin A1 value in 8080 mode WR/RD execute |
| 24 | / | / | / |
| 23 | R | 0 | Wr_Flag<br>0:write operation is finishing<br>1:write operation is pending |
| 22 | R | 0 | Rd_Flag<br>0:read operation is finishing<br>1:read operation is pending |
| 21:18 | / | / | / |
| 17 | R/W | 0 | AUTO<br>auto Transfer Mode:<br>If it's 1, all the valid data during this frame are write to panel.<br>Note: This bit is sampled by Vsync |
| 16 | R/W | 0 | FLUSH<br>direct transfer mode:<br>If it's enabled, FIFO1 is regardless of the HV timing, pixels data keep being transferred unless the input FIFO was empty.<br>Data output rate control by DCLK. |
| 15:6 | / | / | / |
| 5:4 | R/W | 0 | Trigger_Sync_Mode<br>0: start frame flush immediately by bit1.<br>1: start frame flush sync to TE PIN. rising by bit1.<br>2. start frame flush sync to TE PIN. falling by bit1.<br>when set as 1 or 2, io0 is map as TE input. |
| 3 | R/W | 0 | Trigger_FIFO_Bist_En |

| | | | 0: disable |
|---|---|---|---|
| | | | 1: enable |
| | | | Entry addr is 0xFF8 |
| 2 | R/W | 0 | Trigger_FIFO_En |
| | | | 0:enable |
| | | | 1:disable |
| 1 | R/W | 0 | Trigger_Start |
| | | | write '1' to start a frame flush, write'0' has no effect. |
| | | | this flag indicated frame flush is running |
| | | | sofeware must make sure write '1' only when this flag is '0'. |
| 0 | R/W | 0 | Trigger_En |
| | | | 0: trigger mode disable |
| | | | 1: trigger mode enable |

### 6.3.5.16. TCON0_CPU_WR_REG

| Offset: 0x064 | | | Register Name: **TCON0_CPU_WR_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:24 | / | / | / |
| 23:0 | W | 0 | Data_Wr |
| | | | data write on 8080 bus, launch a write operation on 8080 bus |

### 6.3.5.17. TCON0_CPU_RD0_REG

| Offset: 0x068 | | | Register Name: **TCON0_CPU_RD0_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:24 | / | / | / |
| 23:0 | R | / | Data_Rd0 |
| | | | data read on 8080 bus, launch a new read operation on 8080 bus |

### 6.3.5.18. TCON0_CPU_RD1_REG

| Offset: 0x06C | | | Register Name: **TCON0_CPU_RD1_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:24 | / | / | / |
| 23:0 | R | / | Data_Rd1 |
| | | | data read on 8080 bus, without a new read operation on 8080 bus |

## 6.3.5.19. TCON0_LVDS_IF_REG

| Offset: 0x084 | | | Register Name: **TCON0_LVDS_IF_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0 | TCON0_LVDS_En<br>0: disable<br>1: enable |
| 30 | / | / | / |
| 29 | R/W | 0 | TCON0_LVDS_Even_Odd_Dir<br>0: normal<br>1: reverse |
| 28 | R/W | 0 | TCON0_LVDS_Dir<br>0: normal<br>1: reverse<br>NOTE: LVDS direction |
| 27 | R/W | 0 | TCON0_LVDS_Mode<br>0: NS mode<br>1: JEIDA mode |
| 26 | R/W | 0 | TCON0_LVDS_BitWidth<br>0: 24bit<br>1: 18bit |
| 25 | R/W | 0 | TCON0_LVDS_DeBug_En<br>0: disable<br>1: enable |
| 24 | R/W | 0 | TCON0_LVDS_DeBug_Mode<br>0: mode0    RANDOM DATA<br>1: mode1    output CLK period=7/2 LVDS CLK period |
| 23 | R/W | 0 | TCON0_LVDS_Correct_Mode<br>0: mode0<br>1: mode1 |
| 22:21 | / | / | / |
| 20 | R/W | 0 | TCON0_LVDS_Clk_Sel<br>0: MIPI PLL<br>1: TCON0 CLK |
| 19:5 | / | / | / |
| 4 | R/W | 0 | TCON0_LVDS_CLK_Polarity<br>0: reverse<br>1: normal |
| 3:0 | R/W | 0 | TCON0_LVDS_Data_Polarity<br>0: reverse<br>1: normal |

### 6.3.5.20. TCON0_IO_POL_REG

| Offset: 0x088 | | | Register Name: **TCON0_IO_POL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0 | IO_Output_Sel<br>0: normal output<br>1: register output<br>when set as '1', d[23:0], io0, io1,io3 sync to dclk |
| 30:28 | R/W | 0 | DCLK_Sel<br>000: used DCLK0(normal phase offset)<br>001: used DCLK1(1/3 phase offset)<br>010: used DCLK2(2/3 phase offset)<br>100: DCLK0/2    phase 0<br>101: DCLK0/2    phase 90<br>reserved |
| 27 | R/W | 0 | IO3_Inv<br>0: not invert<br>1: invert |
| 26 | R/W | 0 | IO2_ Inv<br>0: not invert<br>1: invert |
| 25 | R/W | 0 | IO1_Inv<br>0: not invert<br>1: invert |
| 24 | R/W | 0 | IO0_Inv<br>0: not invert<br>1: invert |
| 23:0 | R/W | 0 | Data_Inv<br>TCON0 output port D[23:0] polarity control, with independent bit control:<br>0s: normal polarity<br>1s: invert the specify output |

### 6.3.5.21. TCON0_IO_TRI_REG

| Offset: 0x088 | | | Register Name: **TCON0_IO_TRI_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:29 | / | / | / |
| 28 | / | / | RGB_Endian<br>0: normal<br>1: bits_invert |
| 27 | R/W | 1 | IO3_Output_Tri_En<br>1: disable<br>0: enable |

| 26 | R/W | 1 | IO2_Output_Tri_En |
| | | | 1: disable |
| | | | 0: enable |
| 25 | R/W | 1 | IO1_Output_Tri_En |
| | | | 1: disable |
| | | | 0: enable |
| 24 | R/W | 1 | IO0_Output_Tri_En |
| | | | 1: disable |
| | | | 0: enable |
| 23:0 | R/W | 0xFFFFFF | Data_Output_Tri_En |
| | | | TCON0 output port D[23:0] output enable, with independent bit control: |
| | | | 1s: disable |
| | | | 0s: enable |

### 6.3.5.22. TCON_DEBUG_REG

| Offset: 0x0FC | | | Register Name: **TCON_DEBUG_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | / | TCON0_FIFO_Under_Flow |
| 30 | / | / | / |
| 29 | R | / | TCON0_Field_Polarity |
| | | | 0: second field |
| | | | 1: first field |
| 28 | / | / | / |
| 27:16 | R | / | TCON0_Current_Line |
| 15:14 | / | / | / |
| 13 | R/W | 0 | ECC_FIFO_Bypass |
| | | | 0: used |
| | | | 1: bypass |
| 12 | / | / | / |
| 11:0 | / | / | / |

### 6.3.5.23. TCON_CEU_CTL_REG

| Offset: 0x100 | | | Register Name: **TCON_CEU_CTL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0 | CEU_en |
| | | | 0: bypass |
| | | | 1: enable |
| 30:0 | / | / | / |

### 6.3.5.24. TCON_CEU_COEF_MUL_REG

| Offset: 0x110+N*0x04 (N=0,1,2,4,5,6,8,9,10) | | | Register Name: **TCON_CEU_COEF_MUL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:13 | / | / | / |
| 12:0 | R/W | 0 | CEU_Coef_Mul_Value<br>signed 13bit value, range of (-16,16)<br>N=0: Rr<br>N=1: Rg<br>N=2: Rb<br>N=4: Gr<br>N=5: Gg<br>N=6: Gb<br>N=8: Br<br>N=9: Bg<br>N=10: Bb |

### 6.3.5.25. TCON_CEU_COEF_RANG_REG

| Offset: 0x140+N*0x04 (N=0,1,2) | | | Register Name: **TCON_CEU_COEF_RANG_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:24 | / | / | / |
| 23:16 | R/W | 0 | CEU_Coef _Range_Min<br>unsigned 8bit value, range of [0,255] |
| 15:8 | / | / | / |
| 7:0 | R/W | 0 | CEU Coef _Range_Max<br>unsigned 8bit value, range of [0,255] |

### 6.3.5.26. TCON0_CPU_TRI0_REG

| Offset: 0x160 | | | Register Name: **TCON0_CPU_TRI0_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:28 | / | / | / |
| 27:16 | R/W | 0 | Block_Space<br>should be set >20*pixel_cycle |
| 15:12 | / | / | / |
| 11:0 | R/W | 0 | Block_Size |

### 6.3.5.27. TCON0_CPU_TRI1_REG

| Offset: 0x164 | | | Register Name: **TCON0_CPU_TRI1_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:16 | R | 0 | Block_Current_Num |
| 15:0 | R/W | 0 | Block_Num |

### 6.3.5.28. TCON0_CPU_TRI2_REG

| Offset: 0x168 | | | Register Name: **TCON0_CPU_TRI2_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:16 | R/W | 0x20 | Start_Delay<br>Tdly = (Start_Delay +1) * be_clk*8 |
| 15 | R/W | 0 | Trans_Start_Mode<br>0: ecc_fifo+tri_fifo<br>1: tri_fifo |
| 14:13 | R/W | 0 | Sync_Mode<br>0x: auto<br>10: 0<br>11: 1 |
| 12:0 | R/W | 0 | Trans_Start_Set |

### 6.3.5.29. TCON0_CPU_TRI3_REG

| Offset: 0x16C | | | Register Name: **TCON0_CPU_TRI3_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:30 | / | / | / |
| 29:28 | R/W | 0 | |
| 27:24 | / | / | Tri_Int_Mode<br>00: disable<br>01: counter mode<br>10: te rising mode<br>11: te falling mode<br>when set as 01, Tri_Counter_Int occur in cycle of (Count_N+1)$\times$(Count_M+1)$\times$4 dclk.<br>when set as 10 or 11, io0 is map as TE input. |
| 23:8 | R/W | 0 | / |
| 7:0 | R/W | 0 | Counter_N |
| | | | Counter_M |

### 6.3.5.30. TCON0_CPU_TRI4_REG

| Offset: 0x170 | | | Register Name: **TCON0_CPU_TRI4_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:29 | / | / | |
| 28 | R/W | 0 | Plug_Mode_En<br>0: disable<br>1:enable |
| 27:25 | / | / | / |
| 24 | R/W | 0 | A1<br>Valid in first Block |
| 23:0 | R/W | 0 | D23-D0<br>Valid in first Block |

### 6.3.5.31. TCON0_CPU_TRI5_REG

| Offset: 0x174 | | | Register Name: **TCON0_CPU_TRI5_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:25 | / | / | / |
| 24 | R/W | 0 | A1<br>Valid in Block except first |
| 23:0 | R/W | 0 | D23-D0<br>Valid in Block except first |

### 6.3.5.32. TCON_CMAP_CTL_REG

| Offset: 0x180 | | | Register Name: **TCON_CMAP_CTL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0 | Color_Map_En<br>0: bypass<br>1: enable<br>This module only work when X is divided by 4 |
| 30:1 | / | / | / |
| 0 | R/W | 0 | Out_Format<br>0: 4 pixel output mode: Out0 -> Out1 -> Out2 -> Out3<br>1: 2 pixel output mode: Out0 -> Out1 |

### 6.3.5.33. TCON_CMAP_ODD0_REG

| Offset: 0x190 | Register Name: **TCON_CMAP_ODD0_REG** |
|---|---|

| Bit | R/W | Default/Hex | Description |
|---|---|---|---|
| 31:16 | R/W | 0 | Out_Odd1 |
| 15:0 | R/W | 0 | Out_Odd0<br>bit15-12: Reservd<br>bit11-08: Out_Odd0[23:16]<br>bit07-04: Out_Odd0[15:8]<br>bit03-00: Out_Odd0[7:0]<br>0x0: in_b0<br>0x1: in_g0<br>0x2: in_r0<br>0x3: reservd<br>0x4: in_b1<br>0x5: in_g1<br>0x6: in_r1<br>0x7: reservd<br>0x8: in_b2<br>0x9: in_g2<br>0xa: in_r2<br>0xb: reservd<br>0xc: in_b3<br>0xd: in_g3<br>0xe: in_r3<br>0xf: reservd |

## 6.3.5.34. TCON_CMAP_ODD1_REG

| Offset: 0x194 | | | Register Name: **TCON_CMAP_ODD1_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:16 | R/W | 0 | Out_Odd3 |
| 15:0 | R/W | 0 | Out_Odd2 |

## 6.3.5.35. TCON_CMAP_EVEN0_REG

| Offset: 0x198 | | | Register Name: **TCON_CMAP_EVEN0_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:16 | R/W | 0 | Out_Even1 |
| 15:0 | R/W | 0 | Out_Even0 |

## 6.3.5.36. TCON_CMAP_ EVEN1_REG

| Offset: 0x19C | Register Name: **TCON_CMAP_ EVEN1_REG** |
|---|---|

| Bit | R/W | Default/Hex | Description |
|---|---|---|---|
| 31:16 | R/W | 0 | Out_Even3 |
| 15:0 | R/W | 0 | Out_Even2 |

### 6.3.5.37. TCON_ SAFE_PERIOD_REG

| Offset: 0x1F0 | | | Register Name: **TCON_ SAFE_PERIOD_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:29 | / | / | / |
| 28:16 | R/W | 0 | Safe_Period_FIFO_Num |
| 15:2 | / | / | / |
| 1:0 | R/W | 0 | Safe_Period_Mode<br>0: unsafe<br>1: safe<br>2: safe at ecc_fifo_curr_num > safe_period_fifo_num<br>3: safe at 2 and safe at sync active |

### 6.3.5.38. TCON_ LVDS_ANA0_REG

| Offset: 0x220 | | | Register Name: **TCON_ LVDS_ANA0_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0 | lvds0_en_mb<br>enable the bias circuit of the LVDS_Ana module<br>0: Disable    1: Enable |
| 30:25 | / | / | / |
| 24 | R/W | 0 | lvds0_hpren_drvc<br>Enable clock channel drive<br>0: Disable    1: Enable |
| 23:20 | R/W | 0 | lvds0_hpren_drv<br>Enable data channel[3:0] drive<br>0: Disable    1: Enable |
| 19:17 | R/W | 0 | lvds0_reg_c<br>adjust current flowing through Rload of Rx to change the differential signals amplitude<br>0: 216mv    1: 252mv    2: 276mv    3: 312mv<br>4: 336mv    5: 372mv    6: 395mv    7: 432mv |
| 16 | R/W | 0 | lvds0_reg_denc<br>choose data output or PLL test clock output in LVDS_tx |
| 15:12 | R/W | 0 | lvds0_reg_den<br>choose data output or PLL test clock output in LVDS_tx |
| 11 | / | / | / |
| 10:8 | R/W | 0 | lvds0_reg_r |

| | | | adjust current flowing through Rload of Rx to change the common signals amplitude<br>0: 0.925v    1: 0.950v    2: 0.975v    3: 1.000v<br>4: 1.025v    5: 1.050v    6: 1.075v    7: 1.100v |
|---|---|---|---|
| 7:5 | / | / | / |
| 4 | R/W | 0 | lvds0_reg_plrc<br>Lvds clock channel direction<br>0: normal    1: reverse |
| 3:0 | R/W | 0 | lvds0_reg_plr<br>Lvds data channel [3:0] direction<br>0: normal    1: reverse |

## 6.3.6.  TCON1 Module Register List

| Module Name | Base Address |
|---|---|
| TCON1 | 0x01C0D000 |

| Register Name | Offset | Description |
|---|---|---|
| TCON_GCTL_REG | 0x000 | TCON global control register |
| TCON_GINT0_REG | 0x004 | TCON global interrupt register0 |
| TCON_GINT1_REG | 0x008 | TCON global interrupt register1 |
| TCON1_CTL_REG | 0x090 | TCON1 control register |
| TCON1_BASIC0_REG | 0x094 | TCON1 basic timing register0 |
| TCON1_BASIC1_REG | 0x098 | TCON1 basic timing register1 |
| TCON1_BASIC2_REG | 0x09C | TCON1 basic timing register2 |
| TCON1_BASIC3_REG | 0x0A0 | TCON1 basic timing register3 |
| TCON1_BASIC4_REG | 0x0A4 | TCON1 basic timing register4 |
| TCON1_BASIC5_REG | 0x0A8 | TCON1 basic timing register5 |
| TCON1_PS_SYNC_REG | 0x0B0 | TCON1 sync register |
| TCON1_IO_POL_REG | 0x0F0 | TCON1 IO polarity register |
| TCON1_IO_TRI_REG | 0x0F4 | TCON1 IO control register |
| TCON_ECC_FIFO_REG | 0x0F8 | TCON ECC FIFO register |
| TCON_DEBUG_REG | 0x0FC | TCON debug register |
| TCON_CEU_CTL_REG | 0x100 | TCON CEU control register |
| TCON_CEU_COEF_MUL_REG | 0x110+N*0x04 | TCON CEU coefficient register0<br><br>(N=0,1,2,4,5,6,8,9,10) |
| TCON_CEU_COEF_ADD_REG | 0x11C+N*0x10 | TCON CEU coefficient register1<br><br>(N=0,1,2) |

| TCON_CEU_COEF_RANG_REG | 0x140+N*0x04 | TCON CEU coefficient register2 (N=0,1,2) |
|---|---|---|
| TCON_SAFE_PERIOD_REG | 0x1F0 | TCON safe period register |
| TCON1_FILL_CTL_REG | 0x300 | TCON1 fill data control register |
| TCON1_FILL_BEGIN_REG | 0x304+N*0x0C | TCON1 fill data begin register (N=0,1,2) |
| TCON1_FILL_END_REG | 0x308+N*0x0C | TCON1 fill data end register (N=0,1,2) |
| TCON1_FILL_DATA0_REG | 0x30C+N*0x0C | TCON1 fill data value register (N=0,1,2) |
| TCON1_GAMMA_TABLE_REG | 0x400-0x7FF | |
| TCON_ECC_FIFO_BIST_REG | 0xFFC | |

## 6.3.7.  TCON1 Module Register Description

### 6.3.7.1.  TCON_GCTL_REG

| Offset: 0x0000 | | | Register Name: **TCON_GCTL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0 | TCON_En<br>0: disable<br>1: enable<br>When it's disabled, the module will be reset to idle state. |
| 30 | R/W | 0 | TCON_Gamma_En<br>0: disable<br>1: enable |
| 29:0 | / | / | / |

### 6.3.7.2.  TCON_GINT0_REG

| Offset: 0x0004 | | | Register Name: **TCON_GINT0_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | / | / | / |
| 30 | R/W | 0 | TCON1_Vb_Int_En<br>0: disable |

| | | | 1: enable |
|---|---|---|---|
| 29 | / | / | / |
| 28 | R/W | 0 | TCON1_Line_Int_En<br>0: disable<br>1: enable |
| 27:15 | / | / | / |
| 14 | R/W | 0 | TCON1_Vb_Int_Flag<br>Asserted during vertical no-display period every frame.<br>Write 0 to clear it. |
| 13 | / | / | / |
| 12 | R/W | 0 | TCON1_Line_Int_Flag<br>trigger when SY1 match the current TCON1 scan line<br>Write 0 to clear it. |
| 11:0 | / | 0 | / |

### 6.3.7.3. TCON_GINT1_REG

| Offset: 0x0008 | | | Register Name: **TCON_GINT1_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:12 | / | / | / |
| 11:0 | R/W | 0 | TCON1_Line_Int_Num<br>scan line for TCON1 line trigger(including inactive lines)<br>Setting it for the specified line for trigger 1.<br>Note: SY1 is writable only when LINE_TRG1 disable. |

### 6.3.7.4. TCON1_CTL_REG

| Offset: 0x0090 | | | Register Name: **TCON1_CTL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0 | TCON1_En<br>0: disable<br>1: enable |
| 30:9 | / | / | / |
| 8:4 | R/W | 0 | Start_Delay<br>This is for DE1 and DE2 |
| 3:2 | / | / | / |
| 1:0 | R/W | 0 | TCON1_Src_Sel<br>00: DE 0<br>01: reserved<br>1x: BLUE data(FIFO2 disable, RGB=0000FF) |

#### 6.3.7.5. TCON1_BASIC0_REG

| Offset: 0x0094 | | | Register Name: **TCON1_BASIC0_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:28 | / | / | / |
| 27:16 | R/W | 0 | TCON1_XI <br> source width is X+1 |
| 15:12 | / | / | / |
| 11:0 | R/W | 0 | TCON1_YI <br> source height is Y+1 |

#### 6.3.7.6. TCON1_BASIC1_REG

| Offset: 0x0098 | | | Register Name: **TCON1_BASIC1_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:28 | / | / | / |
| 27:16 | R/W | 0 | LS_XO <br> width is LS_XO+1 |
| 15:12 | / | / | / |
| 11:0 | R/W | 0 | LS_YO <br> width is LS_YO+1 <br> NOTE: this version LS_YO =    TCON1_YI |

#### 6.3.7.7. TCON1_BASIC2_REG

| Offset: 0x009C | | | Register Name: **TCON1_BASIC2_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:28 | / | / | / |
| 27:16 | R/W | 0 | TCON1_XO <br> width is TCON1_XO+1 |
| 15:12 | / | / | / |
| 11:0 | R/W | 0 | TCON1_YO <br> height is TCON1_YO+1 |

#### 6.3.7.8. TCON1_BASIC3_REG

| Offset: 0x00A0 | | | Register Name: **TCON1_BASIC3_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:29 | / | / | / |
| 28:16 | R/W | 0 | HT |

| | | | horizontal total time |
| --- | --- | --- | --- |
| | | | Thcycle = (HT+1) * Thdclk |
| 15:12 | / | / | / |
| 11:0 | R/W | 0 | HBP |
| | | | horizontal back porch |
| | | | Thbp = (HBP +1) * Thdclk |

### 6.3.7.9. TCON1_BASIC4_REG

| Offset: 0x00A4 | | | Register Name: **TCON1_BASIC4_REG** |
| --- | --- | --- | --- |
| Bit | R/W | Default/Hex | Description |
| 31:29 | / | / | / |
| 28:16 | R/W | 0 | VT |
| | | | horizontal total time (in HD line) |
| | | | Tvt = VT/2 * Th |
| 15:12 | / | / | / |
| 11:0 | R/W | 0 | VBP |
| | | | horizontal back porch (in HD line) |
| | | | Tvbp = (VBP +1) * Th |

### 6.3.7.10. TCON1_BASIC5_REG

| Offset: 0x00A8 | | | Register Name: **TCON1_BASIC5_REG** |
| --- | --- | --- | --- |
| Bit | R/W | Default/Hex | Description |
| 31:29 | / | / | / |
| 25:16 | R/W | 0 | HSPW |
| | | | horizontal Sync Pulse Width (in dclk) |
| | | | Thspw = (HSPW+1) * Tdclk |
| | | | Note: HT>　(HSPW+1) |
| 15:10 | / | / | / |
| 9:0 | R/W | 0 | VSPW |
| | | | vertical Sync Pulse Width (in lines) |
| | | | Tvspw = (VSPW+1) * Th |
| | | | Note: VT/2 >　(VSPW+1) |

### 6.3.7.11. TCON1_PS_SYNC_REG

| Offset: 0x00B0 | | | Register Name: **TCON1_PS_SYNC_REG** |
| --- | --- | --- | --- |
| Bit | R/W | Default/Hex | Description |
| 31:16 | R/W | 0 | SYNC_X |
| 15:0 | R/W | 0 | SYNC_Y |

### 6.3.7.12. TCON1_IO_POL_REG

| Offset: 0x00F0 | | | Register Name: **TCON1_IO_POL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:28 | / | / | / |
| 27 | R/W | 0 | IO3_Inv<br>0: not invert<br>1: invert |
| 26 | R/W | 0 | IO2 Inv<br>0: not invert<br>1: invert |
| 25 | R/W | 0 | IO1_Inv<br>0: not invert<br>1: invert |
| 24 | R/W | 0 | IO0_Inv<br>0: not invert<br>1: invert |
| 23:0 | R/W | 0 | Data_Inv<br>TCON1 output port D[23:0] polarity control, with independent bit control:<br>0s: normal polarity<br>1s: invert the specify output |

### 6.3.7.13. TCON1_IO_TRI_REG

| Offset: 0x00F4 | | | Register Name: **TCON1_IO_TRI_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:28 | / | / | / |
| 27 | R/W | 1 | IO3_Output_Tri_En<br>1: disable<br>0: enable |
| 26 | R/W | 1 | IO2_Output_Tri_En<br>1: disable<br>0: enable |
| 25 | R/W | 1 | IO1_Output_Tri_En<br>1: disable<br>0: enable |
| 24 | R/W | 1 | IO0_Output_Tri_En<br>1: disable<br>0: enable |
| 23:0 | R/W | 0xFFFFFF | Data_Output_Tri_En<br>TCON1 output port D[23:0] output enable, with independent bit control: |

| | | | 1s: disable |
| | | | 0s: enable |

### 6.3.7.14. TCON1_ECC_FIFO_REG

| Offset: 0x00F8 | | | Register Name: **TCON1_ECC_FIFO_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | / | ECC_FIFO_BIST_EN<br>0: disable<br>1: enable |
| 30 | R/W | / | ECC_FIFO_ERR_FLAG |
| 29:24 | / | / | / |
| 23:16 | R/W | / | ECC_FIFO_ERR_BITS |
| 15:9 | / | / | / |
| 8 | R/W | / | ECC_FIFO_BLANK_EN<br>0: disable ecc function in blanking<br>1: enable ecc function in blanking<br>ECC function is tent to trigged in blanking area at hv mode, set '0' when in hv mode |
| 7:0 | R/W | / | ECC_FIFO_SETTING<br>Note：bit3 0 enable，1 disable |

### 6.3.7.15. TCON_DEBUG_REG

| Offset: 0x00FC | | | Register Name: **TCON_DEBUG_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | / | / | / |
| 30 | R/W | / | TCON1_FIFO_Under_Flow |
| 29 | / | / | / |
| 28 | R | / | TCON1_Field_Polarity<br>0: second field<br>1: first field |
| 27:14 | / | / | / |
| 13 | R/W | 0 | ECC_FIFO_Bypass<br>0: used<br>1: bypass |
| 12 | / | / | / |
| 11:0 | R | / | TCON1_Current_Line |

### 6.3.7.16. TCON_CEU_CTL_REG

| Offset: 0x0100 | | | Register Name: **TCON1_CEU_CTL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0 | CEU_en<br>0: bypass<br>1: enable |
| 30:0 | / | / | / |

### 6.3.7.17. TCON_CEU_COEF_MUL_REG

| Offset: 0x0110+N*0x04<br>(N=0,1,2,4,5,6,8,9,10) | | | Register Name: **TCON_CEU_COEF_MUL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:13 | / | / | / |
| 12:0 | R/W | 0 | CEU_Coef_Mul_Value<br>signed 13bit value, range of (-16,16)<br>N=0: Rr<br>N=1: Rg<br>N=2: Rb<br>N=4: Gr<br>N=5: Gg<br>N=6: Gb<br>N=8: Br<br>N=9: Bg<br>N=10: Bb |

### 6.3.7.18. TCON_CEU_COEF_ADD_REG

| Offset: 0x011C+N*0x10<br>(N=0,1,2) | | | Register Name: **TCON_CEU_COEF_ADD_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:19 | / | / | / |
| 18:0 | R/W | 0 | CEU_Coef_Add_Value<br>signed 19bit value, range of (-16384, 16384)<br>N=0: Rc<br>N=1: Gc<br>N=2: Bc |

#### 6.3.7.19. TCON_CEU_COEF_RANG_REG

| Offset: 0x0140+N*0x4 (N=0,1,2) | | | Register Name: **TCON_CEU_COEF_RANG_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:24 | / | / | / |
| 23:16 | R/W | 0 | CEU_Coef _Range_Min<br>unsigned 8bit value, range of [0,255] |
| 15:8 | / | / | / |
| 7:0 | R/W | 0 | CEU Coef _Range_Max<br>unsigned 8bit value, range of [0,255] |

#### 6.3.7.20. TCON_SAFE_PERIOD_REG

| Offset: 0x01F0 | | | Register Name: **TCON_SAFE_PERIOD_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:29 | / | / | / |
| 28:16 | R/W | 0 | Safe_Period_FIFO_Num |
| 15:2 | / | / | / |
| 1:0 | R/W | 0 | Safe_Period_Mode<br>0: unsafe<br>1: safe<br>2: safe at ecc_fifo_curr_num > safe_period_fifo_num<br>3: safe at 2 and safe at sync active |

#### 6.3.7.21. TCON1_FILL_CTL_REG

| Offset: 0x0300 | | | Register Name: **TCON1_FILL_CTL_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0 | TCON1_Fill_En<br>0: bypass<br>1: enable |
| 30:0 | / | / | / |

#### 6.3.7.22. TCON1_FILL_BEGIN_REG

| Offset: 0x0304+N*0x0C(N=0,1,2) | | | Register Name: **TCON1_FILL_BEGIN_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:24 | / | / | / |
| 23:0 | R/W | 0 | Fill_Begin |

#### 6.3.7.23. TCON1_FILL_END_REG

| Offset: 0x0308+N*0x0C(N=0,1,2) | | | Register Name: **TCON1_FILL_END_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:24 | / | / | / |
| 23:0 | R/W | 0 | Fill_End |

#### 6.3.7.24. TCON1_FILL_DATA_REG

| Offset: 0x030C+N*0x0C(N=0,1,2) | | | Register Name: **TCON1_FILL_DATA_REG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:24 | / | / | / |
| 23:0 | R/W | 0 | Fill_Value |

# Chapter 7 Interfaces

This chapter describes the A83T interfaces, including:

- TWI
- SPI
- UART
- RSB
- CIR Receiver
- USB
- TDM
- EMAC

# 7.1. TWI

## 7.1.1. Overview

This TWI Controller is designed to be used as an interface between CPU host and the serial TWI bus. It can support all the standard TWI transfer, including Slave and Master. The communication to the TWI bus is carried out on a byte-wise basis using interrupt or polled handshaking. This TWI Controller can be operated in standard mode (100K bps) or fast-mode, supporting data rate up to 400K bps. Multiple Masters and 10-bit addressing Mode are supported for this specified application. General Call Addressing is also supported in Slave mode.

The TWI Controller includes the following features:

- Software-programmable for Slave or Master
- Support Repeated START signal
- Multi-master systems supported
- Allow 10-bit addressing with TWI bus
- Performs arbitration and clock synchronization
- Own address and General Call address detection
- Interrupt on address detection
- Support speeds up to 400Kbits/s ('fast mode')
- Allow operation from a wide range of input clock frequencies

## 7.1.2. Timing Description

Data transferred are always in a unit of 8-bit (byte), followed by an acknowledge bit. The number of bytes that can be transmitted per transfer is unrestricted. Data is transferred in serial with the MSB first. Between each byte of data transfer, a receiver device will hold the clock line SCL low to force the transmitter into a wait state while waiting the response from microprocessor.

Data transfer with acknowledge is obligatory. The clock line is driven by the master all the time, including the acknowledge-related clock cycle, except for the SCL holding between each bytes. After sending each byte, the transmitter releases the SDA line to allow the receiver to pull down the SDA line and send an acknowledge signal (or leave it high to send a "not acknowledge") to the transmitter.

When a slave receiver doesn't acknowledge the slave address (unable to receive because of no resource available), the data line must be left high by the slave so that the master can then generate a STOP condition to abort the transfer. Slave receiver can also indicate not to want to send more data during a transfer by leave the acknowledge signal high. And the master should generate the STOP condition to abort the transfer.

## 7.1.3.  TWI Controller Register List

| Module Name | Base Address |
|---|---|
| R_TWI | 0x01F02400 |
| TWI0 | 0x01C2AC00 |
| TWI1 | 0x01C2B000 |
| TWI2 | 0x01C2B400 |

| Register Name | Offset | Description |
|---|---|---|
| TWI_ADDR | 0x0000 | TWI Slave address |
| TWI_XADDR | 0x0004 | TWI Extended slave address |
| TWI_DATA | 0x0008 | TWI Data byte |
| TWI_CNTR | 0x000C | TWI Control register |
| TWI_STAT | 0x0010 | TWI Status register |
| TWI_CCR | 0x0014 | TWI Clock control register |
| TWI_SRST | 0x0018 | TWI Software reset |
| TWI_EFR | 0x001C | TWI Enhance Feature register |
| TWI_LCR | 0x0020 | TWI Line Control register |

## 7.1.4.  TWI Controller Register Description

### 7.1.4.1.  TWI Slave Address Register(Default Value: 0x0000_0000)

| Offset: 0x00 | | | Register Name: **TWI_ADDR** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:8 | / | / | / |
| 7:1 | R/W | 0 | SLA<br>Slave address<br>• 7-bit addressing<br>    SLA6, SLA5, SLA4, SLA3, SLA2, SLA1, SLA0<br>• 10-bit addressing<br>    1, 1, 1, 1, 0, SLAX[9:8] |
| 0 | R/W | 0 | GCE<br>General call address enable<br>0: Disable<br>1: Enable |

**Notes:**

For 7-bit addressing:

SLA6 – SLA0 is the 7-bit address of the TWI when in slave mode. When the TWI receives this address after a START condition, it will generate an interrupt and enter slave mode. (SLA6 corresponds to the first bit received from the TWI bus.) If **GCE** is set to '1', the TWI will also recognize the general call address (00h).

For 10-bit addressing:

When the address received starts with 11110b, the TWI recognizes this as the first part of a 10-bit address and if the next two bits match ADDR[2:1] (i.e. SLAX9 and SLAX8 of the device's extended address), it sends an ACK. (The device does not generate an interrupt at this point.) If the next byte of the address matches the **XADDR** register (SLAX7 – SLAX0), the TWI generates an interrupt and goes into slave mode.

#### 7.1.4.2.   TWI Extend Address Register(Default Value: 0x0000_0000)

| Offset: 0x04 | | | Register Name: **TWI_XADDR** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:8 | / | / | / |
| 7:0 | R/W | 0 | SLAX<br>Extend Slave Address<br>SLAX[7:0] |

#### 7.1.4.3.   TWI Data Register(Default Value: 0x0000_0000)

| Offset: 0x08 | | | Register Name: **TWI_DATA** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:8 | / | / | / |
| 7:0 | R/W | 0 | TWI_DATA<br>Data byte for transmitting or received |

#### 7.1.4.4.   TWI Control Register(Default Value: 0x0000_0000)

| Offset: 0x0C | | | Register Name: **TWI_ CNTR** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:8 | / | / | / |
| 7 | R/W | 0 | INT_EN<br>Interrupt Enable<br>1'b0: The interrupt line always low<br>1'b1: The interrupt line will go high when **INT_FLAG** is set. |
| 6 | R/W | 0 | BUS_EN |

| | | | TWI Bus Enable |
|---|---|---|---|
| | | | 1'b0: The TWI bus inputs ISDA/ISCL are ignored and the TWI Controller will not respond to any address on the bus |
| | | | 1'b1: The TWI will respond to calls to its slave address – and to the general call address if the **GCE** bit in the ADDR register is set. |
| | | | Notes: In master operation mode, this bit should be set to '1' |
| 5 | R/W | 0 | M_STA<br>Master Mode Start<br>When M_STA is set to '1', TWI Controller enters master mode and will transmit a START condition on the bus when the bus is free. If the M_STA bit is set to '1' when the TWI Controller is already in master mode and one or more bytes have been transmitted, then a repeated START condition will be sent. If the M_STA bit is set to '1' when the TWI is being accessed in slave mode, the TWI will complete the data transfer in slave mode then enter master mode when the bus has been released.<br>The M_STA bit is cleared automatically after a START condition has been sent: writing a '0' to this bit has no effect. |
| 4 | R/W | 0 | M_STP<br>Master Mode Stop<br>If M_STP is set to '1' in master mode, a STOP condition is transmitted on the TWI bus. If the M_STP bit is set to '1' in slave mode, the TWI will behave as if a STOP condition has been received, but no STOP condition will be transmitted on the TWI bus. If both **M_STA** and **M_STP** bits are set, the TWI will first transmit the STOP condition (if in master mode) then transmit the START condition.<br>The M_STP bit is cleared automatically: writing a '0' to this bit has no effect. |
| 3 | R/W | 0 | INT_FLAG<br>Interrupt Flag<br>INT_FLAG is automatically set to '1' when any of 28 (out of the possible 29) states is entered (see 'STAT Register' below). The only state that does not set INT_FLAG is state F8h. If the **INT_EN** bit is set, the interrupt line goes high when IFLG is set to '1'. If the TWI is operating in slave mode, data transfer is suspended when INT_FLAG is set and the low period of the TWI bus clock line (SCL) is stretched until '1' is written to INT_FLAG. The TWI clock line is then released and the interrupt line goes low. |
| 2 | R/W | 0 | A_ACK<br>Assert Acknowledge<br>When A_ACK is set to '1', an Acknowledge (low level on SDA) will be sent during the acknowledge clock pulse on the TWI bus if:<br>  1. Either the whole of a matching 7-bit slave address or the first or the second byte of a matching 10-bit slave address has been received.<br>2. The general call address has been received and the **GCE** bit in the ADDR register is set to '1'.<br>3. A data byte has been received in master or slave mode.<br>When A_ACK is '0', a Not Acknowledge (high level on SDA) will be sent when |

| | | | |
|---|---|---|---|
| | | | a data byte is received in master or slave mode.<br><br>If A_ACK is cleared to '0' in slave transmitter mode, the byte in the DATA register is assumed to be the 'last byte'. After this byte has been transmitted, the TWI will enter state C8h then return to the idle state (status code F8h) when INT_FLAG is cleared.<br><br>The TWI will not respond as a slave unless A_ACK is set. |
| 1:0 | R/W | 0 | / |

#### 7.1.4.5. TWI Status Register(Default Value: 0x0000_00F8)

| Offset: 0x10 | | | Register Name: **TWI_ STAT** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:8 | / | / | / |
| | | | STA<br>Status Information Byte<br>Code Status<br>0x00: Bus error<br>0x08: START condition transmitted<br>0x10: Repeated START condition transmitted<br>0x18: Address + Write bit transmitted, ACK received<br>0x20: Address + Write bit transmitted, ACK not received<br>0x28: Data byte transmitted in master mode, ACK received<br>0x30: Data byte transmitted in master mode, ACK not received<br>0x38: Arbitration lost in address or data byte<br>0x40: Address + Read bit transmitted, ACK received<br>0x48: Address + Read bit transmitted, ACK not received<br>0x50: Data byte received in master mode, ACK transmitted<br>0x58: Data byte received in master mode, not ACK transmitted<br>0x60: Slave address + Write bit received, ACK transmitted<br>0x68: Arbitration lost in address as master, slave address + Write bit received, ACK transmitted<br>0x70: General Call address received, ACK transmitted<br>0x78: Arbitration lost in address as master, General Call address received, ACK transmitted<br>0x80: Data byte received after slave address received, ACK transmitted<br>0x88: Data byte received after slave address received, not ACK transmitted<br>0x90: Data byte received after General Call received, ACK transmitted<br>0x98: Data byte received after General Call received, not ACK transmitted<br>0xA0: STOP or repeated START condition received in slave mode<br>0xA8: Slave address + Read bit received, ACK transmitted<br>0xB0: Arbitration lost in address as master, slave address + Read bit received, ACK transmitted |
| 7:0 | R | 0xF8 | 0xB8: Data byte transmitted in slave mode, ACK received<br>0xC0: Data byte transmitted in slave mode, ACK not received |

| | | | 0xC8: Last byte transmitted in slave mode, ACK received |
|---|---|---|---|
| | | | 0xD0: Second Address byte + Write bit transmitted, ACK received |
| | | | 0xD8: Second Address byte + Write bit transmitted, ACK not received |
| | | | 0xF8: No relevant status information, INT_FLAG=0 |
| | | | Others: Reserved |

### 7.1.4.6. TWI Clock Register(Default Value: 0x0000_0000)

| Offset: 0x14 | | | Register Name: **TWI_CCR** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:7 | / | / | / |
| 6:3 | R/W | 0 | CLK_M |
| 2:0 | R/W | 0 | CLK_N <br> The TWI bus is sampled by the TWI at the frequency defined by F0: <br> $Fsamp = F0 = Fin / 2^{CLK\_N}$ <br> The TWI OSCL output frequency, in master mode, is F1/10: <br> $F1 = F0 / (CLK\_M + 1)$ <br> $Foscl = F1 / 10 = Fin / (2^{CLK\_N} * (CLK\_M + 1)*10)$ <br> For Example： <br> Fin = 48Mhz (APB clock input) <br> For 400kHz full speed 2Wire, CLK_N = 2, CLK_M=2 <br> $F0 = 48M/2^2=12Mhz$, F1= F0/(10*(2+1)) = 0.4Mhz <br> For 100Khz standard speed 2Wire, CLK_N=2, CLK_M=11 <br> $F0=48M/2^2=12Mhz$, F1=F0/(10*(11+1)) = 0.1Mhz |

### 7.1.4.7. TWI Soft Reset Register(Default Value: 0x0000_0000)

| Offset: 0x18 | | | Register Name: **TWI_SRST** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:1 | / | / | / |
| 0 | R/W | 0 | SOFT_RST <br> Soft Reset <br> Write '1' to this bit to reset the TWI and clear to '0' when completing Soft Reset operation. |

### 7.1.4.8. TWI Enhance Feature Register(Default Value: 0x0000_0000)

| Offset: 0x1C | | | Register Name: **TWI_ EFR** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |

| 31:2 | / | / | / |
|------|-----|---|---|
| 0:1 | R/W | 0 | DBN<br>Data Byte number follow Read Command Control<br>0— No Data Byte to be written after read command<br>1— Only 1 byte data to be written after read command<br>2— 2 bytes data can be written after read command<br>3— 3 bytes data can be written after read command |

### 7.1.4.9. TWI Line Control Register(Default Value: 0x0000_003A)

| Offset: 0x20 | | | Register Name: **TWI_LCR** |
|------|-----|-------------|---|
| Bit | R/W | Default/Hex | Description |
| 31:6 | / | / | / |
| 5 | R | 1 | SCL_STATE<br>Current state of TWI_SCL<br>0 – low<br>1 - high |
| 4 | R | 1 | SDA_STATE<br>Current state of TWI_SDA<br>0 – low<br>1 - high |
| 3 | R/W | 1 | SCL_CTL<br>TWI_SCL line state control bit<br>When line control mode is enabled (bit[2] set), value of this bit decide the output level of TWI_SCL<br>0 – output low level<br>1 – output high level |
| 2 | R/W | 0 | SCL_CTL_EN<br>TWI_SCL line state control enable<br>When this bit is set, the state of TWI_SCL is control by the value of bit[3].<br>0-disable TWI_SCL line control mode<br>1-enable TWI_SCL line control mode |
| 1 | R/W | 1 | SDA_CTL<br>TWI_SDA line state control bit<br>When line control mode is enabled (bit[0] set), value of this bit decide the output level of TWI_SDA<br>0 – output low level<br>1 – output high level |
| 0 | R/W | 0 | SDA_CTL_EN<br>TWI_SDA line state control enable<br>When this bit is set, the state of TWI_SDA is control by the value of bit[1].<br>0-disable TWI_SDA line control mode<br>1-enable TWI_SDA line control mode |

**7.1.4.10.  TWI DVFS Register(Default Value: 0x0000_0000)**

| Offset: 0x24 | | | Register Name: **TWI_ DVFSCR** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:3 | / | / | / |
| 2 | R/W | 0 | MS_PRIORITY<br>CPU and DVFS BUSY set priority select<br>0: CPU has higher priority<br>1: DVFS has higher priority |
| 1 | R/W | 0 | CPU_BUSY_SET<br>CPU Busy set |
| 0 | R/W | 0 | DVFC_BUSY_SET<br>DVFS Busy set |

**Notes:**

This register is only implemented in TWI0.

## 7.1.5.  TWI Controller Special Requirement

### 7.1.5.1.  TWI Pin List

| Port Name | Width | Direction | Description |
|---|---|---|---|
| TWI_SCL | 1 | IN/OUT | TWI Clock line |
| TWI_SDA | 1 | IN/OUT | TWI Serial Data line |

### 7.1.5.2.  TWI Controller Operation

There are four operation modes on the TWI bus which dictates the communications method. They are Master Transmit, Master Receive, Slave Transmit and Slave Receive. In general, CPU host controls TWI by writing commands and data to its registers. The TWI interrupts the CPU host for the attention each time a byte transfer is done or a START/STOP conditions is detected. The CPU host can also poll the status register for current status if the interrupt mechanism is not disabled by the CPU host.

When the CPU host wants to start a bus transfer, it initiates a bus START to enter the master mode by setting IM_STA bit in the 2WIRE_CNTR register to high (before it must be low). The TWI will assert INT line and INT_FLAG to indicate a completion for the START condition and each consequent byte transfer. At each interrupt, the micro-processor needs to check the 2WIRE_STAT register for current status. A transfer has to be concluded with STOP condition by setting M_STP bit high.

In Slave Mode, the TWI also constantly samples the bus and look for its own slave address during addressing cycles. Once a match is found, it is addressed and interrupt the CPU host with the corresponding status. Upon request, the CPU host should read the status, read/write 2WIRE_DATA data register, and set the 2WIRE_CNTR control register. After each byte transfer, a slave device always halt the operation of remote master by holding the next low pulse on SCL line until the microprocessor responds to the status of previous byte transfer or START condition.

## 7.2. SPI

### 7.2.1. Overview

The SPI is the Serial Peripheral Interface which allows rapid data communication with fewer software interrupts. The SPI module contains one 64x8 receiver buffer (RXFIFO) and one64x8 transmit buffer (TXFIFO). It can work at two modes: Master mode and Slave mode.

   The SPI includes the following features:

- Full-duplex synchronous serial interface
- Master/Slave configurable
- Four chip selects to support multiple peripherals for SPI0,SPI1 has one chip select
- 8-bit wide by 64-entry FIFO for both transmit and receive date
- Polarity and phase of the Chip Select (SPI_SS) and SPI Clock (SPI_SCLK) are configurable

### 7.2.2. SPI Timing Diagram

The serial peripheral interface master uses the SPI_SCLK signal to transfer data in and out of the shift register. Data is clocked using any one of four programmable clock phase and polarity combinations.

During Phase 0, Polarity 0 and Phase 1, Polarity 1 operations, output data changes on the falling clock edge and input data is shifted in on the rising edge.

During Phase 1, Polarity 0 and Phase 0, Polarity 1 operations, output data changes on the rising edges of the clock and is shifted in on falling edges.

The POL defines the signal polarity when SPI_SCLK is in idle state. The SPI_SCLK is high level when POL is '1' and it is low level when POL is '0'. The PHA decides whether the leading edge of SPI_SCLK is used for setup or sample data. The leading edge is used for setup data when PHA is '1' and for sample data when PHA is '0'. The four kind of modes are listed below:

| SPI Mode | POL | PHA | Leading Edge | Trailing Edge |
|---|---|---|---|---|
| 0 | 0 | 0 | Rising, Sample | Falling, Setup |
| 1 | 0 | 1 | Rising, Setup | Falling, Sample |
| 2 | 1 | 0 | Falling, Sample | Rising, Setup |
| 3 | 1 | 1 | Failing, Setup | Rising, Sample |

SPI_SCLK (Mode 0)

SPI_SCLK (Mode 2)

SPI_MOSI

SPI_MISO

SPI_SS

Sample MOSI/ MISO pin
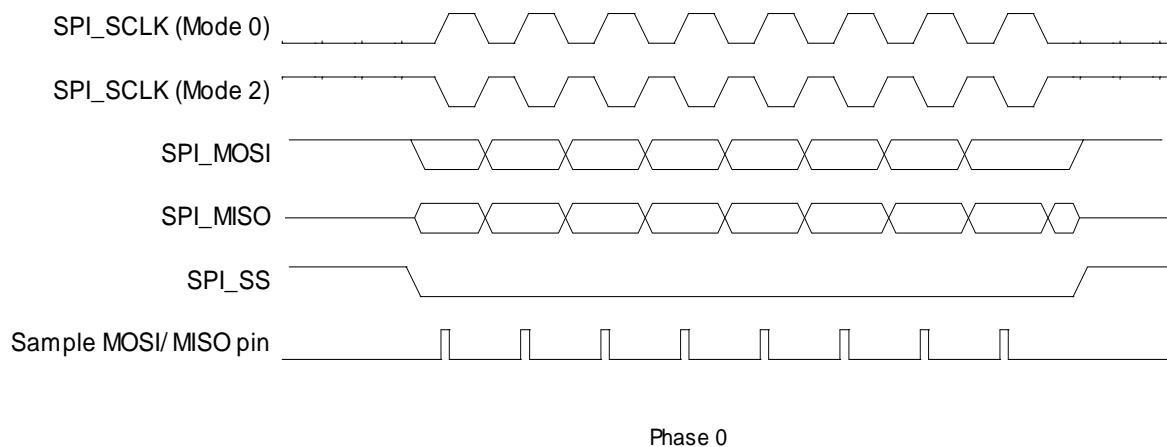
Phase 0

Figure 7-1. SPI Phase 0 Timing Diagram

SPI_SCLK (Mode 1)

SPI_SCLK (Mode 3)

SPI_MOSI

SPI_MISO

SPI_SS

Sample MOSI/ MISO pin
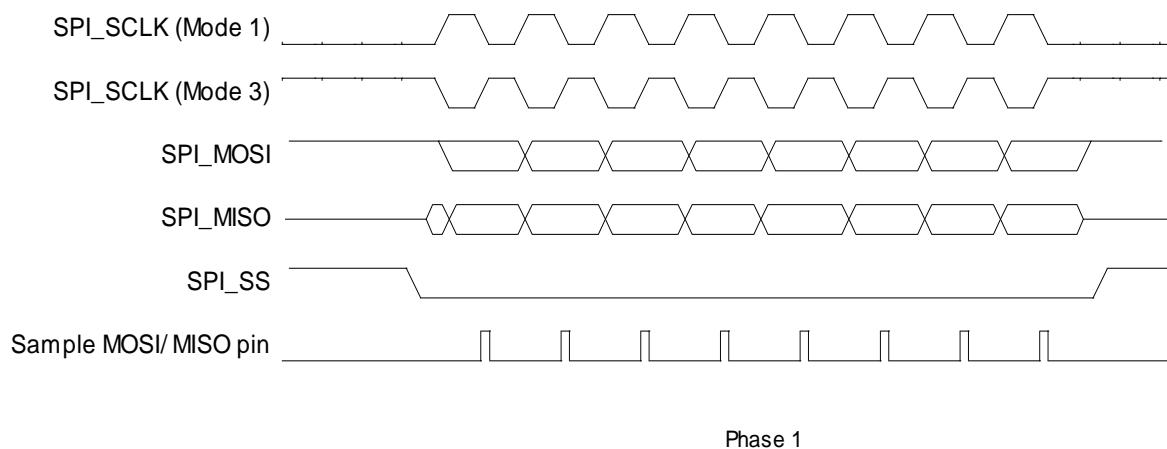
Phase 1

Figure 7-2. SPI Phase 1 Timing Diagram

## 7.2.3.   SPI Register List

| Module Name | Base Address |
|---|---|
| SPI0 | 0x01C68000 |
| SPI1 | 0x01C69000 |

| Register Name | Offset | Description |
|---|---|---|
| SPI_GCR | 0x04 | SPI Global Control Register |
| SPI_TCR | 0x08 | SPI Transfer Control register |
| / | 0x0c | reserved |
| SPI_IER | 0x10 | SPI Interrupt Control register |
| SPI_ISR | 0x14 | SPI Interrupt Status register |
| SPI_FCR | 0x18 | SPI FIFO Control register |
| SPI_FSR | 0x1C | SPI FIFO Status register |
| SPI_WCR | 0x20 | SPI Wait Clock Counter register |
| SPI_CCR | 0x24 | SPI Clock Rate Control register |
| / | 0x28 | reserved |
| / | 0x2c | reserved |
| SPI_MBC | 0x30 | SPI Burst Counter register |
| SPI_MTC | 0x34 | SPI Transmit Counter Register |
| SPI_BCC | 0x38 | SPI Burst Control register |
| SPI_TXD | 0x200 | SPI TX Data register |
| SPI_RXD | 0x300 | SPI RX Data register |

## 7.2.4.   SPI Register Description

### 7.2.4.1.   SPI Global Control Register(Default Value: 0x0000_0080)

| Offset: 0x04 | | | Register Name: **SPI_CTL** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0 | SRST<br>Soft reset<br>Write '1' to this bit will clear the SPI controller, and auto clear to '0' when reset operation completes<br>Write '0' has no effect. |
| 30:8 | / | / | / |
| 7 | R/W | 1 | TP_EN<br>Transmit Pause Enable |

| | | | In master mode, it is used to control transmit state machine to stop smart burst sending when RX FIFO is full.<br>1 – stop transmit data when RXFIFO full<br>0 – normal operation, ignore RXFIFO status<br>Note: Can't be written when XCH=1 |
|---|---|---|---|
| 6:2 | / | / | / |
| 1 | R/W | 0 | MODE<br>SPI Function Mode Select<br>0: Slave Mode<br>1: Master Mode<br>Note: Can't be written when XCH=1 |
| 0 | R/W | 0 | EN<br>SPI Module Enable Control<br>0: Disable<br>1: Enable |

### 7.2.4.2.  SPI Transfer Control Register(Default Value: 0x0000_0087)

| Offset: 0x08 | | | Register Name: **SPI_ INTCTL** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0x0 | XCH<br>Exchange Burst<br>In master mode it is used to start SPI burst<br>0: Idle<br>1: Initiates exchange.<br>Write "1" to this bit will start the SPI burst, and will auto clear after finishing the bursts transfer specified by BC. Write "1" to SRST will also clear this bit. Write '0' to this bit has no effect.<br>Note：Can't be written when XCH=1. |
| 30:14 | / | / | / |
| 13 | R/W | 0x0 | SDM<br>Master Sample Data Mode<br>0 - Delay Sample Mode<br>1 - Normal Sample Mode<br>In Normal Sample Mode, SPI master samples the data at the correct edge for each SPI mode;<br>In Delay Sample Mode, SPI master samples data at the edge that is half cycle delayed by the correct edge defined in respective SPI mode. |
| 12 | R/W | 0x0 | FBS<br>First Transmit Bit Select<br>0: MSB first<br>1: LSB first<br>Note：Can't be written when XCH=1. |
| 11 | R/W | 0x0 | SDC |

| | | | Master Sample Data Control |
|---|---|---|---|
| | | | Set this bit to '1' to make the internal read sample point with a delay of half cycle of SPI_CLK. It is used in high speed read operation to reduce the error caused by the time delay of SPI_CLK propagating between master and slave. |
| | | | 0 – normal operation, do not delay internal read sample point |
| | | | 1 – delay internal read sample point |
| | | | Note：Can't be written when XCH=1. |
| 10 | R/W | 0x0 | RPSM |
| | | | Rapids mode select |
| | | | Select Rapids mode for high speed write. |
| | | | 0: normal write mode |
| | | | 1: rapids write mode |
| | | | Note：Can't be written when XCH=1. |
| 9 | R/W | 0x0 | DDB |
| | | | Dummy Burst Type |
| | | | 0: The bit value of dummy SPI burst is zero |
| | | | 1: The bit value of dummy SPI burst is one |
| | | | Note：Can't be written when XCH=1. |
| 8 | R/W | 0x0 | DHB |
| | | | Discard Hash Burst |
| | | | In master mode it controls whether discarding unused SPI bursts |
| | | | 0: Receiving all SPI bursts in BC period |
| | | | 1: Discard unused SPI bursts, only fetching the SPI bursts during dummy burst period. The bursts number is specified by TC. |
| | | | Note：Can't be written when XCH=1. |
| 7 | R/W | 0x1 | SS_LEVEL |
| | | | When control SS signal manually (SPI_CTRL_REG.SS_CTRL==1), set this bit to '1' or '0' to control the level of SS signal. |
| | | | 0: set SS to low |
| | | | 1: set SS to high |
| | | | Note：Can't be written when XCH=1. |
| 6 | R/W | 0x0 | SS_OWNER |
| | | | SS Output Owner Select |
| | | | Usually, controller sends SS signal automatically with data together. When this bit is set to 1, software must manually write SPI_CTL_REG.SS_LEVEL to 1 or 0 to control the level of SS signal. |
| | | | 0: SPI controller |
| | | | 1: Software |
| | | | Note：Can't be written when XCH=1. |
| 5:4 | R/W | 0x0 | SS_SEL |
| | | | SPI Chip Select |
| | | | Select one of four external SPI Master/Slave Devices |
| | | | 00: SPI_SS0 will be asserted |
| | | | 01: SPI_SS1 will be asserted |
| | | | 10: SPI_SS2 will be asserted |

| | | | |
|---|---|---|---|
| | | | 11: SPI_SS3 will be asserted<br>Note：Can't be written when XCH=1. |
| 3 | R/W | 0x0 | SSCTL<br>In master mode, this bit selects the output wave form for the SPI_SSx signal.<br>Only valid when SS_OWNER = 0.<br>0: SPI_SSx remains asserted between SPI bursts<br>1: Negate SPI_SSx between SPI bursts<br>Note：Can't be written when XCH=1. |
| 2 | R/W | 0x1 | SPOL<br>SPI Chip Select Signal Polarity Control<br>0: Active high polarity (0 = Idle)<br>1: Active low polarity (1 = Idle)<br>Note：Can't be written when XCH=1. |
| 1 | R/W | 0x1 | CPOL<br>SPI Clock Polarity Control<br>0: Active high polarity (0 = Idle)<br>1: Active low polarity (1 = Idle)<br>Note：Can't be written when XCH=1. |
| 0 | R/W | 0x1 | CPHA<br>SPI Clock/Data Phase Control<br>0: Phase 0 (Leading edge for sample data)<br>1: Phase 1 (Leading edge for setup data)<br>Note：Can't be written when XCH=1. |

### 7.2.4.3.    SPI Interrupt Control Register(Default Value: 0x0000_0000)

| Offset: 0x10 | | | Register Name: **SPI_ IER** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:14 | R | 0x0 | Reserved. |
| 13 | R/W | 0x0 | SS_INT_EN<br>SSI Interrupt Enable<br>Chip Select Signal (SSx) from valid state to invalid state<br>0: Disable<br>1: Enable |
| 12 | R/W | 0x0 | TC_INT_EN<br>Transfer Completed Interrupt Enable<br>0: Disable<br>1: Enable |
| 11 | R/W | 0x0 | TF_UDR_INT_EN<br>TXFIFO under run Interrupt Enable<br>0: Disable<br>1: Enable |
| 10 | R/W | 0x0 | TF_OVF_INT_EN<br>TX FIFO Overflow Interrupt Enable |

| | | | 0: Disable |
|---|---|---|---|
| | | | 1: Enable |
| 9 | R/W | 0x0 | RF_UDR_INT_EN |
| | | | RXFIFO under run Interrupt Enable |
| | | | 0: Disable |
| | | | 1: Enable |
| 8 | R/W | 0x0 | RF_OVF_INT_EN |
| | | | RX FIFO Overflow Interrupt Enable |
| | | | 0: Disable |
| | | | 1: Enable |
| 7 | R | 0x0 | Reserved. |
| 6 | R/W | 0x0 | TF_FUL_INT_EN |
| | | | TX FIFO Full Interrupt Enable |
| | | | 0: Disable |
| | | | 1: Enable |
| 5 | R/W | 0x0 | TX_EMP_INT_EN |
| | | | TX FIFO Empty Interrupt Enable |
| | | | 0: Disable |
| | | | 1: Enable |
| 4 | R/W | 0x0 | TX_ERQ_INT_EN |
| | | | TX FIFO Empty Request Interrupt Enable |
| | | | 0: Disable |
| | | | 1: Enable |
| 3 | R | 0x0 | Reserved |
| 2 | R/W | 0x0 | RF_FUL_INT_EN |
| | | | RX FIFO Full Interrupt Enable |
| | | | 0: Disable |
| | | | 1: Enable |
| 1 | R/W | 0x0 | RX_EMP_INT_EN |
| | | | RX FIFO Empty Interrupt Enable |
| | | | 0: Disable |
| | | | 1: Enable |
| 0 | R/W | 0x0 | RF_RDY_INT_EN |
| | | | RX FIFO Ready Request Interrupt Enable |
| | | | 0: Disable |
| | | | 1: Enable |

### 7.2.4.4. SPI Interrupt Status Register(Default Value: 0x0000_0022)

| Offset: 0x14 | | | Register Name: **SPI_ INT_STA** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:14 | / | 0x0 | / |
| 13 | R/W | 0 | SSI |
| | | | SS Invalid Interrupt |

| | | | When SSI is 1, it indicates that SS has changed from valid state to invalid state. Writing 1 to this bit clears it. |
|---|---|---|---|
| 12 | R/W | 0 | TC<br>Transfer Completed<br>In master mode, it indicates that all bursts specified by BC has been exchanged. In other condition, When set, this bit indicates that all the data in TXFIFO has been loaded in the Shift register, and the Shift register has shifted out all the bits. Writing 1 to this bit clears it.<br>0: Busy<br>1: Transfer Completed |
| 11 | R/W | 0 | TF_UDF<br>TXFIFO Underrun<br>This bit is set when if the TXFIFO is underrun. Writing 1 to this bit clears it.<br>0: TXFIFO is not underrun<br>1: TXFIFO is underrun |
| 10 | R/W | 0 | TF_OVF<br>TXFIFO Overflow<br>This bit is set when if the TXFIFO is overflow. Writing 1 to this bit clears it.<br>0: TXFIFO is not overflow<br>1: TXFIFO is overflowed |
| 9 | R/W | 0 | RX_UDF<br>RXFIFO Underrun<br>When set, this bit indicates that RXFIFO has underrun. Writing 1 to this bit clears it. |
| 8 | R/W | 0 | RX_OVF<br>RXFIFO Overflow<br>When set, this bit indicates that RXFIFO has overflowed. Writing 1 to this bit clears it.<br>0: RXFIFO is available.<br>1: RXFIFO has overflowed. |
| 7 | / | / | / |
| 6 | R/W | 0 | TX_FULL<br>TXFIFO Full<br>This bit is set when if the TXFIFO is full . Writing 1 to this bit clears it.<br>0: TXFIFO is not Full<br>1: TXFIFO is Full |
| 5 | R/W | 1 | TX_EMP<br>TXFIFO Empty<br>This bit is set if the TXFIFO is empty. Writing 1 to this bit clears it.<br>0: TXFIFO contains one or more words.<br>1: TXFIFO is empty |
| 4 | R/W | 0 | TX_READY<br>TXFIFO Ready<br>0: TX_WL > TX_TRIG_LEVEL<br>1: TX_WL <= TX_TRIG_LEVEL |

| | | | This bit is set any time if TX_WL <= TX_TRIG_LEVEL. Writing "1" to this bit clears it. Where TX_WL is the water level of RXFIFO |
|---|---|---|---|
| 3 | / | / | reserved |
| 2 | R/W | 0 | RX_FULL<br>RXFIFO Full<br>This bit is set when the RXFIFO is full . Writing 1 to this bit clears it.<br>0: Not Full<br>1: Full |
| 1 | R/W | 1 | RX_EMP<br>RXFIFO Empty<br>This bit is set when the RXFIFO is empty . Writing 1 to this bit clears it.<br>0: Not empty<br>1: empty |
| 0 | R/W | 0 | RX_RDY<br>RXFIFO Ready<br>0: RX_WL < RX_TRIG_LEVEL<br>1: RX_WL >= RX_TRIG_LEVEL<br>This bit is set any time if RX_WL >= RX_TRIG_LEVEL. Writing "1" to this bit clears it. Where RX_WL is the water level of RXFIFO. |

### 7.2.4.5. SPI FIFO Control Register(Default Value: 0x0040_0001)

| Offset: 0x18 | | | Register Name: **SPI_ FCR** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0 | TX_FIFO_RST<br>TX FIFO Reset<br>Write '1' to this bit will reset the control portion of the TX FIFO and auto clear to '0' when completing reset operation, write to '0' has no effect. |
| 30 | R/W | 0 | TF_TEST_ENB<br>TX Test Mode Enable<br>0: disable<br>1: enable<br>Note: In normal mode, TX FIFO can only be read by SPI controller, write '1' to this bit will switch TX FIFO read and write function to AHB bus. This bit is used to test the TX FIFO, don't set in normal operation and don't set RF_TEST and TF_TEST at the same time. |
| 29:26 | / | / | / |
| 25 | / | / | / |
| 24 | R/W | 0x0 | TF_ DRQ_EN<br>TX FIFO DMA Request Enable<br>0: Disable<br>1: Enable |
| 23:16 | R/W | 0x40 | TX_TRIG_LEVEL |

| | | | TX FIFO Empty Request Trigger Level |
|---|---|---|---|
| 15 | R/W | 0x0 | RF_RST<br>RXFIFO Reset<br>Write '1' to this bit will reset the control portion of the receiver FIFO, and auto clear to '0' when completing reset operation, write '0' to this bit has no effect. |
| 14 | R/W | 0x0 | RF_TEST<br>RX Test Mode Enable<br>0: Disable<br>1: Enable<br>Note: In normal mode, RX FIFO can only be written by SPI controller, write '1' to this bit will switch RX FIFO read and write function to AHB bus. This bit is used to test the RX FIFO, don't set in normal operation and don't set RF_TEST and TF_TEST at the same time. |
| 13:10 | R | 0x0 | Reserved |
| 9 | R/W | 0x0 | RX_DMA_MODE<br>SPI RX DMA Mode Control<br>0: Normal DMA mode<br>1: Dedicate DMA mode |
| 8 | R/W | 0x0 | RF_ DRQ_EN<br>RX FIFO DMA Request Enable<br>0: Disable<br>1: Enable |
| 7:0 | R/W | 0x1 | RX_TRIG_LEVEL<br>RX FIFO Ready Request Trigger Level |

### 7.2.4.6.  SPI FIFO Status Register(Default Value: 0x0000_0000)

| Offset: 0x1C | | | Register Name: **SPI_ FSR** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R | 0x0 | TB_WR<br>TX FIFO Write Buffer Write Enable |
| 30:28 | R | 0x0 | TB_CNT<br>TX FIFO Write Buffer Counter<br>These bits indicate the number of words in TX FIFO Write Buffer |
| 27:24 | R | 0x0 | Reserved |
| 23:16 | R | 0x0 | TF_CNT<br>TX FIFO Counter<br>These bits indicate the number of words in TX FIFO<br>0: 0 byte in TX FIFO<br>1: 1 byte in TX FIFO<br>…<br>64: 64 bytes in TX FIFO |
| 15 | R | 0x0 | RB_WR |

| | | | RX FIFO Read Buffer Write Enable |
|---|---|---|---|
| 14:12 | R | 0x0 | RB_CNT<br>RX FIFO Read Buffer Counter<br>These bits indicate the number of words in RX FIFO Read Buffer |
| 11:8 | R | 0x0 | Reserved |
| 7:0 | R | 0x0 | RF_CNT<br>RX FIFO Counter<br>These bits indicate the number of words in RX FIFO<br>0: 0 byte in RX FIFO<br>1: 1 byte in RX FIFO<br>…<br>64:64 bytes in RX FIFO |

### 7.2.4.7.  SPI Wait Clock Register(Default Value: 0x0000_0000)

| Offset: 0x20 | | | Register Name: **SPI_ WAIT** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:20 | / | / | / |
| 19:16 | R/W | 0x0 | SWC<br>Dual mode direction switch wait clock counter (for master mode only).<br>0: No wait states inserted<br>n: n SPI_SCLK wait states inserted<br>Note: These bits control the number of wait states to be inserted before start dual data transfer in dual SPI mode. The SPI module counts SPI_SCLK by SWC for delaying next word data transfer.<br>Note：Can't be written when XCH=1. |
| 15:0 | R/W | 0 | WCC<br>Wait Clock Counter (In Master mode)<br>These bits control the number of wait states to be inserted in data transfers. The SPI module counts SPI_SCLK by WCC for delaying next word data transfer.<br>0: No wait states inserted<br>N: N SPI_SCLK wait states inserted |

### 7.2.4.8.  SPI Clock Control Register(Default Value: 0x0000_0002)

| Offset: 0x24 | | | Register Name: **SPI_ CCTL** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:13 | / | / | / |
| 12 | R/W | 0 | DRS<br>Divide Rate Select (Master Mode Only)<br>0: Select Clock Divide Rate 1 |

| | | | 1: Select Clock Divide Rate 2 |
|---|---|---|---|
| 11:8 | R/W | 0 | CDR1<br>Clock Divide Rate 1 (Master Mode Only)<br>The SPI_SCLK is determined according to the following equation: SPI_CLK = Source_CLK / 2^n. |
| 7:0 | R/W | 0x2 | CDR2<br>Clock Divide Rate 2 (Master Mode Only)<br>The SPI_SCLK is determined according to the following equation: SPI_CLK = Source_CLK / (2*(n + 1)). |

### 7.2.4.9.  SPI Master Burst Counter Register(Default Value: 0x0000_0000)

| Offset: 0x30 | | | Register Name: **SPI_BC** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:24 | / | / | / |
| 23:0 | R/W | 0 | MBC<br>Master Burst Counter<br>In master mode, this field specifies the total burst number.<br>0: 0 burst<br>1: 1 burst<br>…<br>N: N bursts |

### 7.2.4.10.  SPI Master Transmit Counter Register(Default Value: 0x0000_0000)

| Offset: 0x34 | | | Register Name: **SPI_TC** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:24 | / | / | / |
| 23:0 | R/W | 0 | MWTC<br>Master Write Transmit Counter<br>In master mode, this field specifies the burst number that should be sent to TXFIFO before automatically sending dummy burst. For saving bus bandwidth, the dummy burst (all zero bits or all one bits) is sent by SPI Controller automatically.<br>0: 0 burst<br>1: 1 burst<br>…<br>N: N bursts |

### 7.2.4.11. SPI Master Burst Control Counter Register(Default Value: 0x0000_0000)

| Offset: 0x38 | | | Register Name: **SPI_BCC** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:29 | R | 0x0 | Reserved |
| 28 | R/W | 0x0 | DRM<br>Master Dual Mode RX Enable<br>0: RX use single-bit mode<br>1: RX use dual mode<br>Note：Can't be written when XCH=1. |
| 27:24 | R/W | 0x0 | DBC<br>Master Dummy Burst Counter<br>In master mode, this field specifies the burst number that should be sent before receive in dual SPI mode. The data is don't care by the device.<br>0: 0 burst<br>1: 1 burst<br>…<br>N: N bursts<br>Note：Can't be written when XCH=1. |
| 23:0 | R/W | 0x0 | STC<br>Master Single Mode Transmit Counter<br>In master mode, this field specifies the burst number that should be sent in single mode before automatically sending dummy burst. This is the first transmit counter in all bursts.<br>0: 0 burst<br>1: 1 burst<br>…<br>N: N bursts<br>Note：Can't be written when XCH=1. |

### 7.2.4.12. SPI TX Data Register(Default Value: 0x0000_0000)

| Offset: 0x200 | | | Register Name: **SPI_ TXD** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | W/R | 0x0 | TDATA<br>Transmit Data<br>This register can be accessed in byte, half-word or word unit by AHB. In byte accessing method, if there are rooms in RXFIFO, one burst data is written to RXFIFO and the depth is increased by 1. In half-word accessing method, two SPI burst data are written and the TXFIFO depth is increase by 2. In word accessing method, four SPI burst data are written and the TXFIFO depth is increased by 4.<br>Note: This address is writing-only if TF_TEST is '0', and if TF_TEST is set to '1', this address is readable and writable to test the TX FIFO through the AHB |

| | | | bus. |
|---|---|---|---|

### 7.2.4.13. SPI RX Data Register(Default Value: 0x0000_0000)

| Offset: 0x300 | | | Register Name: **SPI_ RXD** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R | 0 | RDATA<br>Receive Data<br>This register can be accessed in byte, half-word or word unit by AHB. In byte accessing method, if there are data in RXFIFO, the top word is returned and the RXFIFO depth is decreased by 1. In half-word accessing method, two SPI bursts are returned and the RXFIFO depth is decrease by 2. In word accessing method, the four SPI bursts are returned and the RXFIFO depth is decreased by 4.<br>Note: This address is read-only if RF_TEST is '0', and if RF_TEST is set to '1', this address is readable and writable to test the RX FIFO through the AHB bus. |

## 7.2.5.   SPI Special Requirement

### 7.2.5.1.   SPI Pin List

The direction of SPI pin is different in two work modes: Master Mode and Slave Mode.

| Port Name | Width | Direction(M) | Direction(S) | Description |
|---|---|---|---|---|
| SPI_CLK | 1 | OUT | IN | SPI Clock |
| SPI_MOSI | 1 | OUT | IN | SPI Master Output Slave Input Data Signal |
| SPI_MISO | 1 | IN | OUT | SPI Master Input Slave Output Data Signal |
| SPI_CS | 1 | OUT | IN | SPI Chip Select Signal |

**7.2.5.2.  SPI Module Clock Source and Frequency**

The SPI module uses two clock source: AHB_CLK and SPI_CLK. The SPI_SCLK can in the range from 3Khz to 100 MHZ and AHB_CLK >= 2xSPI_SCLK.

| Clock Name | Description | Requirement |
|---|---|---|
| AHB_CLK | AHB bus clock, as the clock source of SPI module | AHB_CLK >= 2xSPI_SCLK |
| SPI_CLK | SPI serial input clock | |

## 7.3. UART

### 7.3.1. Overview

The UART is used for serial communication with a peripheral, modem (data carrier equipment, DCE) or data set. Data is written from a master (CPU) over the APB bus to the UART and it is converted to serial form and transmitted to the destination device. Serial data is also received by the UART and stored for the master (CPU) to read back.

The UART contains registers to control the character length, baud rate, parity generation/checking, and interrupt generation. Although there is only one interrupt output signal from the UART, there are several prioritized interrupt types that can be responsible for its assertion. Each of the interrupt types can be separately enabled/disabled with the control registers.

The UART has 16450 and 16550 modes of operation, which are compatible with a range of standard software drivers. In 16550 mode, transmit and receive operations are both buffered by FIFOs. In 16450 mode, these FIFOs are disabled.

The UART supports data lengths from five to eight bits, an optional parity bit and 1, 1 ½ or 2 stop bits, and is fully programmable by an AMBA APB CPU interface. A 16-bit programmable baud rate generator and an 8-bit scratch register are included, together with separate transmit and receive FIFOs. Eight modem control lines and a diagnostic loop-back mode are provided.

Interrupts can be generated for a range of TX Buffer/FIFO, RX Buffer/FIFO, Modem Status and Line Status conditions.

The UART includes the following features:

- Compatible with industry-standard 16550 UARTs
- 64-Bytes Transmit and receive data FIFOs
- DMA controller interface
- Software/ Hardware Flow Control
- Programmable Transmit Holding Register Empty interrupt
- Interrupt support for FIFOs, Status Change
- Support IrDa 1.0 SIR

## 7.3.2. UART Timing Diagram



Figure 7-3. UART Serial Data Format

## 7.3.3. UART Controller Register List

There are 6 UART controllers.All UART controllers can be configured as Serial IrDA.

| Module Name | Base Address |
|---|---|
| UART0 | 0x01C28000 |
| UART1 | 0x01C28400 |
| UART2 | 0x01C28800 |
| UART3 | 0x01C28C00 |
| UART4 | 0x01C29000 |
| R-UART | 0x01F02800 |

| Register Name | Offset | Description |
|---|---|---|
| UART_RBR | 0x00 | UART Receive Buffer Register |
| UART_THR | 0x00 | UART Transmit Holding Register |
| UART_DLL | 0x00 | UART Divisor Latch Low Register |
| UART_DLH | 0x04 | UART Divisor Latch High Register |
| UART_IER | 0x04 | UART Interrupt Enable Register |
| UART_IIR | 0x08 | UART Interrupt Identity Register |
| UART_FCR | 0x08 | UART FIFO Control Register |
| UART_LCR | 0x0C | UART Line Control Register |
| UART_MCR | 0x10 | UART Modem Control Register |
| UART_LSR | 0x14 | UART Line Status Register |
| UART_MSR | 0x18 | UART Modem Status Register |
| UART_SCH | 0x1C | UART Scratch Register |
| UART_USR | 0x7C | UART Status Register |
| UART_TFL | 0x80 | UART Transmit FIFO Level |
| UART_RFL | 0x84 | UART_RFL |
| UART_HALT | 0xA4 | UART Halt TX Register |

## 7.3.4. UART Register Description

### 7.3.4.1. UART Receiver Buffer Register(Default Value: 0x0000_0000)

| Offset: 0x0000 | | | Register Name: **UART_RBR** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:8 | / | / | / |
| 7:0 | R | 0 | RBR<br>Receiver Buffer Register<br>Data byte received on the serial input port . The data in this register is valid only if the Data Ready (DR) bit in the Line Status Register (LCR) is set.<br>If in FIFO mode and FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO is preserved, but any incoming data are lost and an overrun error occurs. |

### 7.3.4.2. UART Transmit Holding Register(Default Value: 0x0000_0000)

| Offset: 0x0000 | | | Register Name: **UART_ THR** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:8 | / | / | / |
| 7:0 | W | 0 | THR<br>Transmit Holding Register<br>Data to be transmitted on the serial output port . Data should only be written to the THR when the THR Empty (THRE) bit (LSR[5]) is set.<br>If in FIFO mode and FIFOs are enabled (FCR[0] = 1) and THRE is set, 16 number of characters of data may be written to the THR before the FIFO is full. Any attempt to write data when the FIFO is full results in the write data being lost. |

### 7.3.4.3. UART Divisor Latch Low Register(Default Value: 0x0000_0000)

| Offset: 0x0000 | | | Register Name: **UART_ DLL** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:8 | / | / | / |
| 7:0 | R/W | 0 | DLL<br>Divisor Latch Low<br>Lower 8 bits of a 16-bit, read/write, Divisor Latch register that contains the |

baud rate divisor for the UART. This register may only be accessed when the DLAB bit (LCR[7]) is set and the UART is not busy (USR[0] is zero).

The output baud rate is equal to the serial clock (sclk) frequency divided by sixteen times the value of the baud rate divisor, as follows: baud rate = (serial clock freq) / (16 * divisor).

Note that with the Divisor Latch Registers (DLL and DLH) set to zero, the baud clock is disabled and no serial communications occur. Also, once the DLL is set, at least 8 clock cycles of the slowest UART clock should be allowed to pass before transmitting or receiving data.

### 7.3.4.4. UART Divisor Latch High Register(Default Value: 0x0000_0000)

| Offset: 0x0004 | | | Register Name: **UART_ DLH** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:8 | / | / | / |
| 7:0 | R/W | 0 | DLH<br>Divisor Latch High<br>Upper 8 bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. This register may only be accessed when the DLAB bit (LCR[7]) is set and the UART is not busy (USR[0] is zero).<br>The output baud rate is equal to the serial clock (sclk) frequency divided by sixteen times the value of the baud rate divisor, as follows: baud rate = (serial clock freq) / (16 * divisor).<br>Note that with the Divisor Latch Registers (DLL and DLH) set to zero, the baud clock is disabled and no serial communications occur. Also, once the DLH is set, at least 8 clock cycles of the slowest UART clock should be allowed to pass before transmitting or receiving data. |

### 7.3.4.5. UART Interrupt Enable Register(Default Value: 0x0000_0000)

| Offset: 0x0004 | | | Register Name: **UART_ IER** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:8 | / | / | / |
| 7 | R/W | | PTIME<br>Programmable THRE Interrupt Mode Enable<br>This is used to enable/disable the generation of THRE Interrupt.<br>0: Disable<br>1: Enable |
| 6:4 | / | / | / |
| 3 | R/W | 0 | EDSSI<br>Enable Modem Status Interrupt<br>This is used to enable/disable the generation of Modem Status Interrupt. |

| | | | This is the fourth highest priority interrupt.<br>0: Disable<br>1: Enable |
|---|---|---|---|
| 2 | R/W | 0 | ELSI<br>Enable Receiver Line Status Interrupt<br>This is used to enable/disable the generation of Receiver Line Status Interrupt. This is the highest priority interrupt.<br>0: Disable<br>1: Enable |
| 1 | R/W | 0 | ETBEI<br>Enable Transmit Holding Register Empty Interrupt<br>This is used to enable/disable the generation of Transmitter Holding Register Empty Interrupt. This is the third highest priority interrupt.<br>0: Disable<br>1: Enable |
| 0 | R/W | 0 | ERBFI<br>Enable Received Data Available Interrupt<br>This is used to enable/disable the generation of Received Data Available Interrupt and the Character Timeout Interrupt (if in FIFO mode and FIFOs enabled). These are the second highest priority interrupts.<br>0: Disable<br>1: Enable |

### 7.3.4.6. UART Interrupt Identity Register(Default Value: 0x0000_0000)

| Offset: 0x0008 | | | Register Name: **UART_ IIR** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:8 | / | / | / |
| 7:6 | R | 0 | FEFLAG<br>FIFOs Enable Flag<br>This is used to indicate whether the FIFOs are enabled or disabled.<br>00: Disable<br>11: Enable |
| 5:4 | / | / | / |
| 3:0 | R | 0x1 | IID<br>Interrupt ID<br>This indicates the highest priority pending interrupt which can be one of the following types:<br>0000: modem status<br>0001: no interrupt pending<br>0010: THR empty<br>0100: received data available<br>0110: receiver line status |

| | | | 0111: busy detect |
|---|---|---|---|
| | | | 1100: character timeout |
| | | | Bit 3 indicates an interrupt can only occur when the FIFOs are enabled and used to distinguish a Character Timeout condition interrupt. |

| Interrupt ID | Priority Level | Interrupt Type | Interrupt Source | Interrupt Reset |
|---|---|---|---|---|
| 0001 | - | None | None | - |
| 0110 | Highest | Receiver line status | Overrun/parity/ framing errors or break interrupt | Reading the line status register |
| 0100 | Second | Received data available | Receiver data available (non-FIFO mode or FIFOs disabled) or RCVR FIFO trigger level reached (FIFO mode and FIFOs enabled) | Reading the receiver buffer register (non-FIFO mode or FIFOs disabled) or the FIFO drops below the trigger level (FIFO mode and FIFOs enabled) |
| 1100 | Second | Character timeout indication | No characters in or out of the RCVR FIFO during the last 4 character times and there is at least 1character in it during This time | Reading the receiver buffer register |
| 0010 | Third | Transmit holding register empty | Transmitter holding register empty (Program THRE Mode disabled) or XMIT FIFO at or below threshold (Program THRE Mode enabled) | Reading the IIR register (if source of interrupt); or, writing into THR (FIFOs or THRE Mode not selected or disabled) or XMIT FIFO above threshold (FIFOs and THRE Mode selected and enabled). |
| 0000 | Fourth | Modem status | Clear to send or data set ready or ring indicator or data carrier detect. Note that if auto flow control mode is enabled, a change in CTS (that is, DCTS set) does not cause an interrupt. | Reading the Modem status Register |
| 0111 | Fifth | Busy detect indication | UART_16550_COMPATIBLE = NO and master has tried to write to the Line Control Register while the UART is busy (USR[0] is set to one). | Reading the UART status register |

### 7.3.4.7. UART FIFO Control Register(Default Value: 0x0000_0000)

| Offset: 0x0008 | | | Register Name: **UART_ FCR** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:8 | / | / | / |
| 7:6 | W | 0 | RT |

| Bit | R/W | Default/Hex | Description |
|---|---|---|---|
| | | | RCVR Trigger |
| | | | This is used to select the trigger level in the receiver FIFO at which the Received Data Available Interrupt is generated. In auto flow control mode it is used to determine when the rts_n signal is de-asserted. It also determines when the dma_rx_req_n signal is asserted in certain modes of operation. |
| | | | 00: 1 character in the FIFO |
| | | | 01: FIFO ¼ full |
| | | | 10: FIFO ½ full |
| | | | 11: FIFO-2 less than full |
| | | | TFT |
| | | | TX Empty Trigger |
| | | | Writes have no effect when THRE_MODE_USER = Disabled. This is used to select the empty threshold level at which the THRE Interrupts are generated when the mode is active. It also determines when the dma_tx_req_n signal is asserted when in certain modes of operation. |
| | | | 00: FIFO empty |
| | | | 01: 2 characters in the FIFO |
| | | | 10: FIFO ¼ full |
| 5:4 | W | 0 | 11: FIFO ½ full |
| | | | DMAM |
| | | | DMA Mode |
| | | | 0: Mode 0 |
| 3 | W | 0 | 1: Mode 1 |
| | | | XFIFOR |
| | | | XMIT FIFO Reset |
| | | | This resets the control portion of the transmit FIFO and treats the FIFO as empty. This also de-asserts the DMA TX request. |
| 2 | W | 0 | It is 'self-clearing'. It is not necessary to clear this bit. |
| | | | RFIFOR |
| | | | RCVR FIFO Reset |
| | | | This resets the control portion of the receive FIFO and treats the FIFO as empty. This also de-asserts the DMA RX request. |
| 1 | W | 0 | It is 'self-clearing'. It is not necessary to clear this bit. |
| | | | FIFOE |
| | | | Enable FIFOs |
| | | | This enables/disables the transmit (XMIT) and receive (RCVR) FIFOs. Whenever the value of this bit is changed both the XMIT and RCVR |
| 0 | W | 0 | controller portion of FIFOs is reset. |

### 7.3.4.8.  UART Line Control Register(Default Value: 0x0000_0000)

| Offset: 0x000C | | | Register Name: **UART_ LCR** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:8 | / | / | / |

| | | | |
|---|---|---|---|
| 7 | R/W | 0 | DLAB<br>Divisor Latch Access Bit<br>It is writeable only when UART is not busy (USR[0] is zero) and always readable. This bit is used to enable reading and writing of the Divisor Latch register (DLL and DLH) to set the baud rate of the UART. This bit must be cleared after initial baud rate setup in order to access other registers.<br>0: Select RX Buffer Register (RBR) / TX Holding Register(THR) and Interrupt Enable Register (IER)<br>1: Select Divisor Latch LS Register (DLL) and Divisor Latch MS Register (DLM) |
| 6 | R/W | 0 | BC<br>Break Control Bit<br>This is used to cause a break condition to be transmitted to the receiving device. If set to one the serial output is forced to the spacing (logic 0) state. When not in Loopback Mode, as determined by MCR[4], the sout line is forced low until the Break bit is cleared. If SIR_MODE = Enabled and active (MCR[6] set to one) the sir_out_n line is continuously pulsed. When in Loopback Mode, the break condition is internally looped back to the receiver and the sir_out_n line is forced low. |
| 5:4 | R/W | 0 | EPS<br>Even Parity Select<br>It is writeable only when UART is not busy (USR[0] is zero) and always writable readable. This is used to select between even and odd parity, when parity is enabled (PEN set to one). Setting the LCR[5] is to reverse the LCR[4].<br>00: Odd Parity<br>01: Even Parity<br>1X: Reverse LCR[4] |
| 3 | R/W | 0 | PEN<br>Parity Enable<br>It is writeable only when UART is not busy (USR[0] is zero) and always readable. This bit is used to enable and disable parity generation and detection in transmitted and received serial character respectively.<br>0: parity disabled<br>1: parity enabled |
| 2 | R/W | 0 | STOP<br>Number of stop bits<br>It is writeable only when UART is not busy (USR[0] is zero) and always readable. This is used to select the number of stop bits per character that the peripheral transmits and receives. If set to zero, one stop bit is transmitted in the serial data. If set to one and the data bits are set to 5 (LCR[1:0] set to zero) one and a half stop bits is transmitted. Otherwise, two stop bits are transmitted. Note that regardless of the number of stop bits selected, the receiver checks only the first stop bit.<br>0: 1 stop bit<br>1: 1.5 stop bits when DLS (LCR[1:0]) is zero, else 2 stop bit |
| 1:0 | R/W | 0 | DLS |

| | | | Data Length Select |
|---|---|---|---|
| | | | It is writeable only when UART is not busy (USR[0] is zero) and always readable. This is used to select the number of data bits per character that the peripheral transmits and receives. The number of bit that may be selected areas follows: |
| | | | 00: 5 bits |
| | | | 01: 6 bits |
| | | | 10: 7 bits |
| | | | 11: 8 bits |

### 7.3.4.9.  UART Modem Control Register(Default Value: 0x0000_0000)

| Offset: 0x0010 | | | Register Name: **UART_MCR** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:6 | / | / | / |
| 5 | R/W | 0 | AFCE<br>Auto Flow Control Enable<br>When FIFOs are enabled and the Auto Flow Control Enable (AFCE) bit is set, Auto Flow Control features are enabled.<br>0: Auto Flow Control Mode disabled<br>1: Auto Flow Control Mode enabled |
| 4 | R/W | 0 | LOOP<br>Loop Back Mode<br>0: Normal Mode<br>1: Loop Back Mode<br>This is used to put the UART into a diagnostic mode for test purposes. If operating in UART mode (SIR_MODE != Enabled or not active, MCR[6] set to zero), data on the sout line is held high, while serial data output is looped back to the sin line, internally. In this mode all the interrupts are fully functional. Also, in loopback mode, the modem control inputs (dsr_n, cts_n, ri_n, dcd_n) are disconnected and the modem control outputs (dtr_n, rts_n, out1_n, out2_n) are looped back to the inputs, internally. If operating in infrared mode (SIR_MODE == Enabled AND active, MCR[6] set to one), data on the sir_out_n line is held low, while serial data output is inverted and looped back to the sir_in line. |
| 3:2 | / | / | / |
| 1 | R/W | 0 | RTS<br>Request to Send<br>This is used to directly control the Request to Send (rts_n) output. The Request To Send (rts_n) output is used to inform the modem or data set that the UART is ready to exchange data. When Auto RTS Flow Control is not enabled (MCR[5] set to zero), the rts_n signal is set low by programming MCR[1] (RTS) to a high.In Auto Flow Control, AFCE_MODE == Enabled and active (MCR[5] set to one) and FIFOs enable (FCR[0] set to one), the rts_n |

| | | | output is controlled in the same way, but is also gated with the receiver FIFO threshold trigger (rts_n is inactive high when above the threshold). The rts_n signal is de-asserted when MCR[1] is set low. <br> 0: rts_n de-asserted (logic 1) <br> 1: rts_n asserted (logic 0) <br> Note that in Loopback mode (MCR[4] set to one), the rts_n output is held inactive high while the value of this location is internally looped back to an input. |
|---|---|---|---|
| 0 | R/W | 0 | DTR <br> Data Terminal Ready <br> This is used to directly control the Data Terminal Ready (dtr_n) output. The value written to this location is inverted and driven out on dtr_n. <br> 0：dtr_n de-asserted (logic 1) <br> 1：dtr_n asserted (logic 0) <br> The Data Terminal Ready output is used to inform the modem or data set that the UART is ready to establish communications. <br> Note that in Loopback mode (MCR[4] set to one), the dtr_n output is held inactive high while the value of this location is internally looped back to an input. |

### 7.3.4.10. UART Line Status Register(Default Value: 0x0000_0060)

| Offset: 0x0014 | | | Register Name: **UART_LSR** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:8 | / | / | / |
| 7 | R | 0 | FIFOERR <br> RX Data Error in FIFO <br> When FIFOs are disabled, this bit is always 0. When FIFOs are enabled, this bit is set to 1 when there is at least one PE, FE, or BI in the RX FIFO. It is cleared by a read from the LSR register provided there are no subsequent errors in the FIFO. |
| 6 | R | 1 | TEMT <br> Transmitter Empty <br> If the FIFOs are disabled, this bit is set to "1" whenever the TX Holding Register and the TX Shift Register are empty. If the FIFOs are enabled, this bit is set whenever the TX FIFO and the TX Shift Register are empty. In both cases, this bit is cleared when a byte is written to the TX data channel. |
| 5 | R | 1 | THRE <br> TX Holding Register Empty <br> If the FIFOs are disabled, this bit is set to "1" whenever the TX Holding Register is empty and ready to accept new data and it is cleared when the CPU writes to the TX Holding Register. <br> If the FIFOs are enabled, this bit is set to "1" whenever the TX FIFO is empty and it is cleared when at least one byte is written |

| | | | |
|---|---|---|---|
| | | | to the TX FIFO. |
| 4 | R | 0 | BI<br>Break Interrupt<br>This is used to indicate the detection of a break sequence on the serial input data.<br>It is set whenever the serial input, sin, is held in a logic '0' state for longer than the sum of start time + data bits + parity + stop bits.<br>In the FIFO mode, the character associated with the break condition is carried through the FIFO and is revealed when the character is at the top of the FIFO. Reading the LSR clears the BI bit. In the non-FIFO mode, the BI indication occurs immediately and persists until the LSR is read. |
| 3 | R | 0 | FE<br>Framing Error<br>This is used to indicate the occurrence of a framing error in the receiver. A framing error occurs when the receiver does not detect a valid STOP bit in the received data.<br>In the FIFO mode, since the framing error is associated with a character received, it is revealed when the character with the framing error is at the top of the FIFO. When a framing error occurs, the UART tries to resynchronize. It does this by assuming that the error was due to the start bit of the next character and then continues receiving the other bit i.e. data, and/or parity and stop. It should be noted that the Framing Error (FE) bit (LSR[3]) is set if a break interrupt has<br>occurred, as indicated by Break Interrupt (BI) bit (LSR[4]).<br>0: no framing error<br>1:framing error<br>Reading the LSR clears the FE bit. |
| 2 | R | 0 | PE<br>Parity Error<br>This is used to indicate the occurrence of a parity error in the receiver if the Parity Enable (PEN) bit (LCR[3]) is set. In the FIFO mode, since the parity error is associated with a character received, it is revealed when the character with the parity error arrives at the top of the FIFO. It should be noted that the Parity Error (PE) bit (LSR[2]) is set if a break interrupt has occurred, as indicated by Break Interrupt (BI) bit (LSR[4]).<br>0: no parity error<br>1: parity error<br>Reading the LSR clears the PE bit. |
| 1 | R | 0 | OE<br>Overrun Error<br>This occurs if a new data character was received before the previous data was read. In the non-FIFO mode, the OE bit is set when a new character arrives in the receiver before the previous character was read from the RBR. When this happens, the data in the RBR is overwritten. In the FIFO mode, an overrun error occurs when the FIFO is full and a new character arrives at the |

| | | | receiver. The data in the FIFO is retained and the data in the receive shift register is lost. |
|---|---|---|---|
| | | | 0: no overrun error |
| | | | 1: overrun error |
| | | | Reading the LSR clears the OE bit. |
| 0 | R | 0 | DR<br>Data Ready<br>This is used to indicate that the receiver contains at least one character in the RBR or the receiver FIFO.<br>0: no data ready<br>1: data ready<br>This bit is cleared when the RBR is read in non-FIFO mode, or when the receiver FIFO is empty, in FIFO mode. |

### 7.3.4.11. UART Modem Status Register(Default Value: 0x0000_0000)

| Offset: 0x0018 | | | Register Name: **UART_MSR** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:8 | / | / | / |
| 7 | R | 0 | DCD<br>Line State of Data Carrier Detect<br>This is used to indicate the current state of the modem control line dcd_n. This bit is the complement of dcd_n. When the Data Carrier Detect input (dcd_n) is asserted it is an indication that the carrier has been detected by the modem or data set.<br>0: dcd_n input is de-asserted (logic 1)<br>1: dcd_n input is asserted (logic 0) |
| 6 | R | 0 | RI<br>Line State of Ring Indicator<br>This is used to indicate the current state of the modem control line ri_n. This bit is the complement of ri_n. When the Ring Indicator input (ri_n) is asserted it is an indication that a telephone ringing signal has been received by the modem or data set.<br>0: ri_n input is de-asserted (logic 1)<br>1: ri_n input is asserted (logic 0) |
| 5 | R | 0 | DSR<br>Line State of Data Set Ready<br>This is used to indicate the current state of the modem control line dsr_n. This bit is the complement of dsr_n. When the Data Set Ready input (dsr_n) is asserted it is an indication that the modem or data set is ready to establish communications with UART.<br>0: dsr_n input is de-asserted (logic 1)<br>1: dsr_n input is asserted (logic 0)<br>In Loopback Mode (MCR[4] set to one), DSR is the same as MCR[0] (DTR). |

| | | | |
|---|---|---|---|
| 4 | R | 0 | CTS<br>Line State of Clear To Send<br>This is used to indicate the current state of the modem control line cts_n.<br>This bit is the complement of cts_n. When the Clear to Send input (cts_n) is asserted it is an indication that the modem or data set is ready to exchange data with UART.<br>0: cts_n input is de-asserted (logic 1)<br>1: cts_n input is asserted (logic 0)<br>In Loopback Mode (MCR[4] = 1), CTS is the same as MCR[1] (RTS). |
| 3 | R | 0 | DDCD<br>Delta Data Carrier Detect<br>This is used to indicate that the modem control line dcd_n has changed since the last time the MSR was read.<br>0: no change on dcd_n since last read of MSR<br>1: change on dcd_n since last read of MSR<br>Reading the MSR clears the DDCD bit.<br>Note: Ff the DDCD bit is not set and the dcd_n signal is asserted (low) and a reset occurs (software or otherwise), then the DDCD bit is set when the reset is removed if the dcd_n signal remains asserted. |
| 2 | R | 0 | TERI<br>Trailing Edge Ring Indicator<br>This is used to indicate that a change on the input ri_n (from an active-low to an inactive-high state) has occurred since the last time the MSR was read.<br>0: no change on ri_n since last read of MSR<br>1: change on ri_n since last read of MSR<br>Reading the MSR clears the TERI bit. |
| 1 | R | 0 | DDSR<br>Delta Data Set Ready<br>This is used to indicate that the modem control line dsr_n has changed since the last time the MSR was read.<br>0: no change on dsr_n since last read of MSR<br>1: change on dsr_n since last read of MSR<br>Reading the MSR clears the DDSR bit. In Loopback Mode (MCR[4] = 1), DDSR reflects changes on MCR[0] (DTR).<br>Note: If the DDSR bit is not set and the dsr_n signal is asserted (low) and a reset occurs (software or otherwise), then the DDSR bit is set when the reset is removed if the dsr_n signal remains asserted. |
| 0 | R | 0 | DCTS<br>Delta Clear to Send<br>This is used to indicate that the modem control line cts_n has changed since the last time the MSR was read.<br>0: no change on ctsdsr_n since last read of MSR<br>1: change on ctsdsr_n since last read of MSR<br>Reading the MSR clears the DCTS bit. In Loopback Mode (MCR[4] = 1), DCTS reflects changes on MCR[1] (RTS). |

| | | | Note: If the DCTS bit is not set and the cts_n signal is asserted (low) and a reset occurs (software or otherwise), then the DCTS bit is set when the reset isremoved if the cts_n signal remains asserted. |

### 7.3.4.12. UART Scratch Register(Default Value: 0x0000_0000)

| Offset: 0x001C | | | Register Name: **UART_ SCH** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:8 | / | / | / |
| 7:0 | R/W | 0 | SCRATCH_REG<br>Scratch Register<br>This register is for programmers to use as a temporary storage space. It has no defined purpose in the UART. |

### 7.3.4.13. UART Status Register(Default Value: 0x0000_0006)

| Offset: 0x007C | | | Register Name: **UART_ USR** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:5 | / | / | / |
| 4 | R | 0 | RFF<br>Receive FIFO Full<br>This is used to indicate that the receive FIFO is completely full.<br>0: Receive FIFO not full<br>1: Receive FIFO Full<br>This bit is cleared when the RX FIFO is no longer full. |
| 3 | R | 0 | RFNE<br>Receive FIFO Not Empty<br>This is used to indicate that the receive FIFO contains one or more entries.<br>0: Receive FIFO is empty<br>1: Receive FIFO is not empty<br>This bit is cleared when the RX FIFO is empty. |
| 2 | R | 1 | TFE<br>Transmit FIFO Empty<br>This is used to indicate that the transmit FIFO is completely empty.<br>0: Transmit FIFO is not empty<br>1: Transmit FIFO is empty<br>This bit is cleared when the TX FIFO is no longer empty. |
| 1 | R | 1 | TFNF<br>Transmit FIFO Not Full<br>This is used to indicate that the transmit FIFO in not full.<br>0: Transmit FIFO is full<br>1: Transmit FIFO is not full |

| | | | This bit is cleared when the TX FIFO is full. |
|---|---|---|---|
| 0 | R | 0 | BUSY<br>UART Busy Bit<br>0: Idle or inactive<br>1: Busy |

### 7.3.4.14. UART Transmit FIFO Level Register(Default Value: 0x0000_0000)

| Offset: 0x0080 | | | Register Name: **UART_ TFL** |
|---|---|---|---|
| Bit | R/W | Defalut/Hex | Description |
| 31:7 | / | / | / |
| 6:0 | R | 0 | TFL<br>Transmit FIFO Level<br>This is indicates the number of data entries in the transmit FIFO. |

### 7.3.4.15. UART Receive FIFO Level Register(Default Value: 0x0000_0000)

| Offset: 0x0084 | | | Register Name: **UART_ RFL** |
|---|---|---|---|
| Bit | R/W | Defalut/Hex | Description |
| 31:7 | / | / | / |
| 6:0 | R | 0 | RFL<br>Receive FIFO Level<br>This is indicates the number of data entries in the receive FIFO. |

### 7.3.4.16. UART Halt TX Register(Default Value: 0x0000_0000)

| Offset: 0x00A4 | | | Register Name: **UART_ HALT** |
|---|---|---|---|
| Bit | R/W | Defalut/Hex | Description |
| 31:6 | / | / | / |
| 5 | R/W | 0 | SIR_RX_INVERT<br>SIR Receiver Pulse Polarity Invert<br>0: Not invert receiver signal<br>1: Invert receiver signal |
| 4 | R/W | 0 | SIR_TX_INVERT<br>SIR Transmit Pulse Polarity Invert<br>0: Not invert transmit pulse<br>1: Invert transmit pulse |
| 3 | / | / | / |
| 2 | R/W | 0 | CHANGE_UPDATE<br>After the user using HALT[1] to change the baudrate or LCR configuration,<br>write 1 to update the configuration and waiting this bit self clear to 0 to |

| | | | finish update process. Write 0 to this bit has no effect.<br>1: Update trigger, Self clear to 0 when finish update. |
|---|---|---|---|
| 1 | R/W | 0 | CHCFG_AT_BUSY<br>This is an enable bit for the user to change LCR register configuration (except for the DLAB bit) and baudrate register (DLH and DLL) when the UART is busy (USB[0] is 1).<br>1: Enable change when busy |
| 0 | R/W | 0 | HALT_TX<br>Halt TX<br>This register is use to halt transmissions for testing, so that the transmit FIFO can be filled by the master when FIFOs are implemented and enabled.<br>0 : Halt TX disabled<br>1 : Halt TX enabled<br>Note: If FIFOs are not enabled, the setting of the halt TX register has no effect on operation. |

## 7.3.5. UART Pin List

| Port Name | Width | Direction | Description |
|---|---|---|---|
| UART0_TX | 1 | OUT | UART Serial Bit output |
| UART0_RX | 1 | IN | UART Serial Bit input |
| UART1_TX | 1 | OUT | UART Serial Bit output |
| UART1_RX | 1 | IN | UART Serial Bit input |
| UART1_RTS | 1 | OUT | UART Request To Send<br>This active low output signal informs Modem that the UART is ready to send data |
| UART1_CTS | 1 | IN | UART Clear To End<br>This active low signal is an input showing when Modem is ready to accept data |
| UART2_TX | 1 | OUT | UART Serial Bit output |
| UART2_RX | 1 | IN | UART Serial Bit input |
| UART2_RTS | 1 | OUT | UART Request To Send<br>This active low output signal informs Modem that the UART is ready to send data |
| UART2_CTS | 1 | IN | UART Clear To End<br>This active low signal is an input showing when Modem is ready to accept data |
| UART3_TX | 1 | OUT | UART Serial Bit output |
| UART3_RX | 1 | IN | UART Serial Bit input |
| UART3_RTS | 1 | OUT | UART Request To Send |

| | | | This active low output signal informs Modem that the UART is ready to send data |
|---|---|---|---|
| UART3_CTS | 1 | IN | UART Clear To End<br>This active low signal is an input showing when Modem is ready to accept data |
| UART4_TX | 1 | OUT | UART Serial Bit output |
| UART4_RX | 1 | IN | UART Serial Bit input |
| UART4_RTS | 1 | OUT | UART Request To Send<br>This active low output signal informs Modem that the UART is ready to send data |
| UART4_CTS | 1 | IN | UART Clear To End<br>This active low signal is an input showing when Modem is ready to accept data |
| R_UART_TX | 1 | OUT | UART Serial Bit output |
| R_UART_RX | 1 | IN | UART Serial Bit input |

## 7.4. RSB

### 7.4.1. Overview

The RSB (reduced serial bus) Host Controller is designed to communicate with RSB Device using two push-pull wires. It supports a simplified two wire protocol (RSB) on a push-pull bus. The transfer speed can be up to 20MHz and the performance will be improved much.

The RSB includes the following features:

- Supports industry-standard AMBA Peripheral Bus (APB) and it is fully compliant with the AMBA Specification, Revision 2.0.
- Supports speed up to 20MHz with ultra low power
- Supports Push-Pull bus
- Supports Master mode
- Supports programmable output delay of CD signal
- Supports parity check for address and data transmission
- Supports multi-slaves

### 7.4.2. Terminology Definition

| TERM | Description |
|------|-------------|
| CK | A line that is used to transmit clock from Host to Device |
| CD | A line that is used to transmit Command and Data between Host and Device |
| DA | Device Address is a 16bits address that is the ID of each type device. |
| RTA | Run-Time Address is an 8bits address that is used to address device during Read or Write transmission. The valid RTA is 0x17 0x2D 0x3A 0x4E 0x59 0x63 0x74 0x8B 0x9C 0xA6 0xB1 0xC5 0xD2 0xE8 and 0xFF. |
| HD | Host to Device Handshake is used to change the ownership of CD from Host to Device. |
| DH | Device to Host Handshake is used to change the ownership of CD from Device to Host. |
| SB | Start Bit: a HIGH to LOW transition on the CD while CK is high. |

### 7.4.3. RSB Command Set

| Command | Value | Description |
|---------|-------|-------------|
| SRTA | 0xE8 | Set Run-Time-Address |
| RD8 | 0x8B | Read one byte from Device |
| RD16 | 0x9C | Read two bytes from Device |
| RD32 | 0xA6 | Read four bytes from Device |
| WR8 | 0x4E | Write one byte to Device |
| WR16 | 0x59 | Write two bytes to Device |
| WR32 | 0x63 | Write four bytes to Device |

### 7.4.4. Software Operation Flow
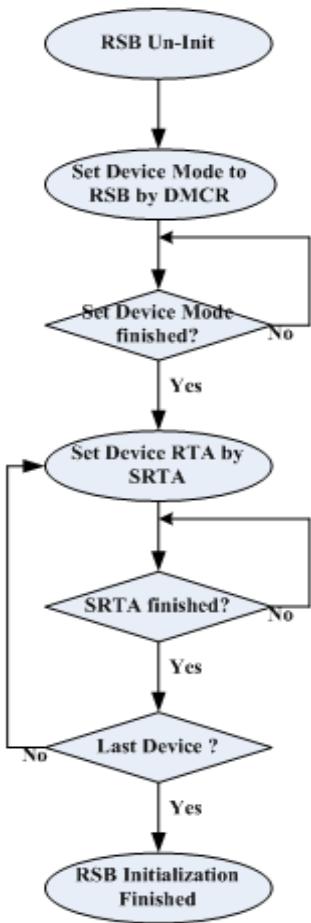


Figure 7-4. RSB System Initialization
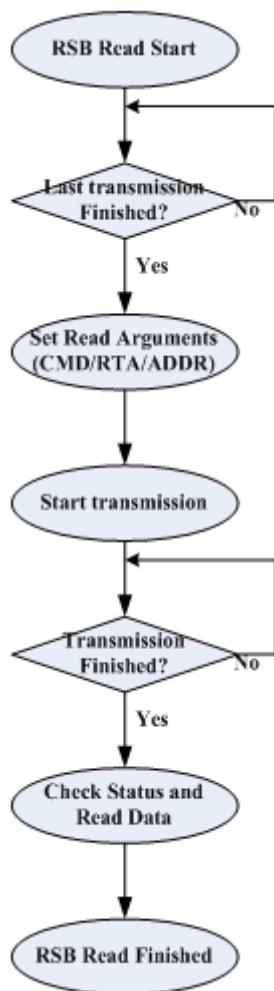
Figure 7-5. RSB Read from Device
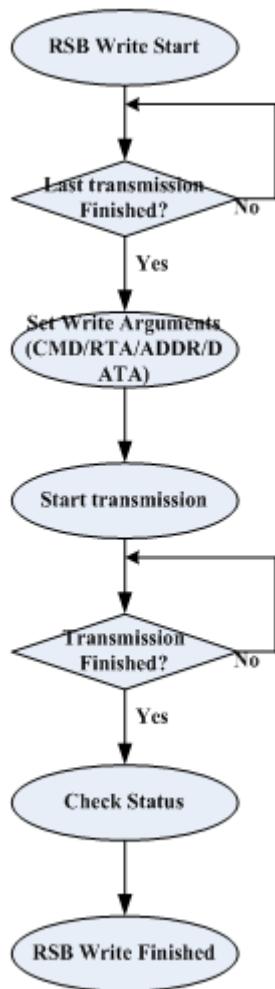
Figure 7-6. RSB Write to Device

## 7.4.5.  RSB Controller Register List

| Module Name | Base Address |
|---|---|
| RSB | 0x01F03400 |

| Register Name | Offset | Description |
|---|---|---|
| RSB_CTRL | 0x0000 | RSB Control Register |
| RSB_CCR | 0x0004 | RSB Clock Control Register |
| RSB_INTE | 0x0008 | RSB Interrupt Enable Register |
| RSB_STAT | 0x000c | RSB Status Register |
| RSB_DADDR0 | 0x0010 | RSB Data Acess Address Register0 |
| RSB_DLEN | 0x0018 | RSB Data Length Register |

| RSB_DATA0 | 0x001c | RSB Data Buffer0 Register |
|-----------|--------|---------------------------|
| RSB_LCR | 0x0024 | RSB Line Control register |
| RSB_PMCR | 0x0028 | RSB PMU Mode Control register |
| RSB_CMD | 0x002C | RSB Command Register |
| RSB_SADDR | 0x0030 | RSB Slave address Register |

## 7.4.6.    RSB Register Description

### 7.4.6.1.   RSB Control Register (Default Value: 0x0000_0000)

| Offset: 0x0000 | | | Register Name: **RSB_CTRL** |
|------|-----|-------------|-------------|
| Bit | R/W | Default/Hex | Description |
| 31:8 | / | / | / |
| 7 | R/W | 0 | START_TRANS<br>Write '1' to this bit will start a new transmission with the configuration of other registers. It is cleared to '0' automatically when the transaction completes or an error happens in the transmission. |
| 6 | R/W | 0 | ABORT_TRANS<br>Write '1' to this bit will abort the current transmission. It is cleared to '0' automatically when the transmission has been aborted. |
| 5:2 | / | / | / |
| 1 | R/W | 0 | GLOBAL_INT_ENB<br>Global interrupt enable bit<br>1 – enable interrupt<br>0 – disable interrupt |
| 0 | R/W | 0 | Soft Reset<br>Write '1' to this bit will reset the controller into default state. All of the status of controller will be cleared. And this bit will be cleared to '0' automatically when reset operation completes. |

### 7.4.6.2.   RSB Clock Control Register (Default Value: 0x0000_0000)

| Offset: 0x0004 | | | Register Name: **RSB_ CCR** |
|------|-----|-------------|-------------|
| Bit | R/W | Default/Hex | Description |
| 31:11 | / | / | / |
| 10:8 | R/W | 0 | CD_ODLY<br>CD output delay Delay time of n source clock cycles before output CD signal. |
| 7:0 | R/W | 0 | CK_DIV |

| | | | Fck = Fsource / 2*(divider+1) |
|---|---|---|---|

### 7.4.6.3.  RSB Interrupt Enable Register (Default Value: 0x0000_0000)

| Offset: 0x0008 | | | Register Name: **RSB_ INTE** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:3 | / | / | / |
| 2 | R/W | 0 | LOAD_BSY_ENB<br>Loading Busy Interrupt Enable<br>1 – enable<br>0 – disable |
| 1 | R/W | 0 | TRANS_ERR_ENB<br>Transfer Error Interrupt Enable<br>1 – enable<br>0 – disable |
| 0 | R/W | 0 | TRANS_OVER_ENB<br>Transfer complete Interrupt Enable<br>1 – enable<br>0 – disable |

### 7.4.6.4.  RSB Status Register (Default Value: 0x0000_0000)

| Offset: 0x000C | | | Register Name: **RSB_ INTS** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:17 | / | / | / |
| 16 | R | 0 | TRANS_ERR_ACK.<br>If a negative ACK is received from Device, then this bit is set to '1' by hardware.<br>This bit is cleared when a new transmission is started. |
| 15:12 | / | / | / |
| 11:8 | R | 0 | TRANS_ERR_DATA<br>If the parity check of 1st byte is negative, then bit8 is set to '1' by hardware.<br>If the parity check of 2nd byte is negative, then bit9 is set to '1' by hardware; and so on.<br>These bits are cleared when a new transmission is started. |
| 7:3 | / | / | / |
| 2 | R/W | 0 | LOAD_BSY<br>Loading Busy Flag<br>If software writes any control registers during transmission, this bit will be set to '1'.<br>If LOAD_BSY_ENB=1, an interrupt will be generated.<br>Software can clear this flag by writing '1' to this bit. |

| | | | TRANS_ERR |
|---|---|---|---|
| | | | Transfer Error Flag |
| | | | If an error happened during transmission, This bit will be set to '1'. |
| | | | If TRANS_ERR_ENB=1, an interrupt will be generated. |
| 1 | R/W | 0 | Software can clear this flag by writing '1' to this bit. |
| | | | TRANS_OVER |
| | | | Transfer Over Flag |
| | | | If the transmission has transfer over, this bit is set to '1'. |
| | | | If TRANS_OVER_ENB=1, an interrupt will be generated. |
| 0 | R/W | 0 | Software can clear this flag by writing '1' to this bit. |

### 7.4.6.5.  RSB Address Register (Default Value: 0x0000_0000)

| Offset: 0x0010 | | | Register Name: **RSB_ AR** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:8 | / | / | / |
| 7:0 | R/W | 0 | ADDR<br>The ADDR is send to device during Read and Write command. |

### 7.4.6.6.  RSB Data Length Register (Default Value: 0x0000_0000)

| Offset: 0x0018 | | | Register Name: **RSB_ DLEN** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:5 | / | / | / |
| 4 | R/W | 0 | READ_WRITE_FLAG<br>Read/Write flag<br>1:read<br>0:write |
| 3 | / | / | / |
| 2:0 | R/W | 0 | Data Acess Length<br>Only use 0,1,3 now- Package length is n+1 bytes |

### 7.4.6.7.  RSB Data Buffer0 Register (Default Value: 0x0000_0000)

| Offset: 0x001C | | | Register Name: **RSB_ DATA0** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:24 | R/W | 0 | Data Byte 4 |
| 23:16 | R/W | 0 | Data Byte 3 |
| 15:8 | R/W | 0 | Data Byte 2 |
| 7:0 | R/W | 0 | Data Byte 1 |

#### 7.4.6.8. RSB Line Control Register (Default Value: 0x0000_003A)

| Offset: 0x0024 | | | Register Name: **RSB_ LCR** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:6 | / | / | / |
| 5 | R | 1 | CK_STATE<br>Current state of CK pin<br>0 – low<br>1 - high |
| 4 | R | 1 | CD_STATE<br>Current state of CD pin<br>0 – low<br>1 – high |
| 3 | R/W | 1 | CK_CTL<br>CK line state control bit<br>When line control mode is enabled (bit[2] set), value of this bit decide the output level of CK<br>0 – output low level<br>1 – output high level |
| 2 | R/W | 0 | CK_CTL_EN<br>CK line state control enable<br>When this bit is set, the state of CK is control by the value of bit[3].<br>0-disable CK line control mode<br>1-enable CK line control mode |
| 1 | R/W | 1 | CD_CTL<br>CD line state control bit<br>When line control mode is enabled (bit[0] set), value of this bit decide the output level of CD<br>0 – output low level<br>1 – output high level |
| 0 | R/W | 0 | CD_CTL_EN<br>CD line state control enable<br>When this bit is set, the state of CD is control by the value of bit [1].<br>0-disable CD line control mode<br>1-enable CD line control mode |

#### 7.4.6.9. RSB PMU Mode Control Register (Default Value: 0x003E_3E00)

| Offset: 0x0028 | | | Register Name: **RSB_PMCR** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0 | PMU_INIT_SEND<br>1:Send initial sequence to PMU to switch PMU's bus mode from NTWI to RSB. |

| | | | 0:write ignore |
|---|---|---|---|
| | | | This bit will be self-cleared when initial sequence is sent onto the RSB bus. |
| 30:24 | / | / | / |
| 23:16 | R/W | 0x3E | PMU_INIT_DATA<br>Value of PMU's initial data |
| 15:8 | R/W | 0x3E | PMU MODE Control Register Address |
| 7:0 | R/W | 0 | PMU Device Address |

### 7.4.6.10. RSB Command Register (Default Value: 0x0000_0000)

| Offset: 0x002C | | | Register Name: **RSB_ CMD** |
|---|---|---|---|
| Bit | R/W | Defaul/Hex | Description |
| 31:8 | / | / | / |
| 7:0 | R/W | 0 | CMD_IDX<br>command index |
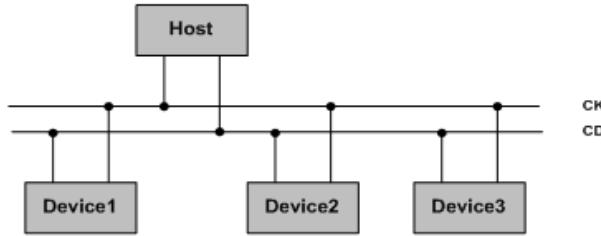
### 7.4.6.11. RSB Slave Address Register (Default Value: 0x0000_0000)

| Offset: 0x0030 | | | Register Name: **RSB_ SADDR** |
|---|---|---|---|
| Bit | R/W | Defaul/Hex | Description |
| 31:24 | / | / | / |
| 23:16 | R/W | 0 | RTSADDR<br>Run-Time Slave Address |
| 15:0 | R/W | 0 | SADDR<br>Slave Address |

## 7.4.7.   RSB General Specification
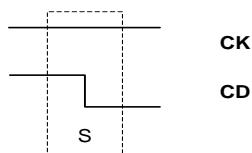
RSB uses push-pull bus，and supports multi-devices. It uses CK as clock and uses CD to transmit command and data. The Bus Topology is showed below:

**RSB Bus Topology**
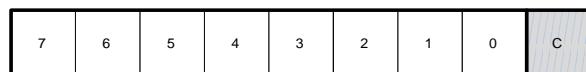
The start bit marks the beginning of a transaction. The Start bit is defined as a HIGH to LOW transition on the CD while CK is high.

**Start signal**



RSB protocol uses parity bit to check the correction of address and data.
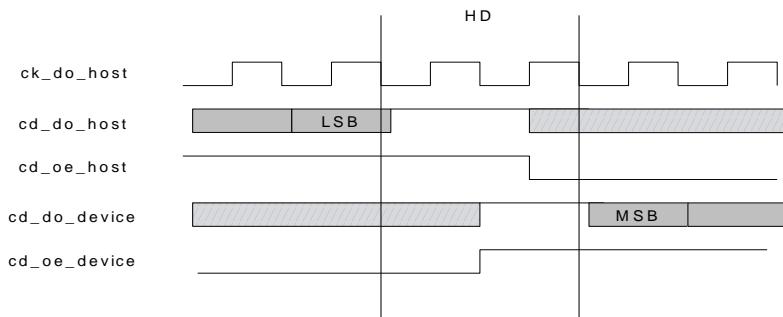
Parity bit
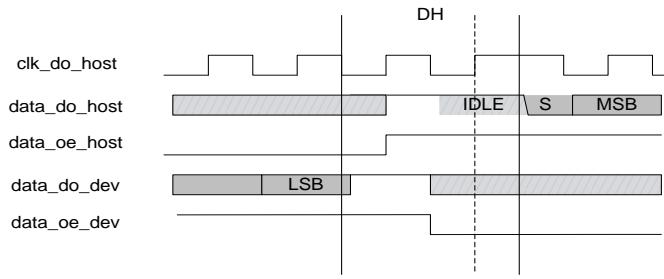


ACK bit is the acknowledgement from device to host, The ACK is low active. When device finds the parity bit is error, it will not send ACK to host, so host can know that an error happens in the transaction.

Both Host and Device can drive the CD, so there are two handshakes, HD (host to device) and DH (device to host), for Host and device to convert the direction of data transmission.

**HD Handshake**



**DH Handshake**

To improve transaction efficiency and to be flexible in device address assignment, RSB use Device Address (DA) and Run-Time Address (RTA). RTA is assigned dynamically by host. Host software shall ensure that different device has different RTA in the same system. Device's default RTA is 0 and 0 is the reserved address. If RTA is 0 when setting RTA, the setting is invalid.

There are three command types in RSB:

1)  Set run-time address (RTA): It is used to set run time address (RTA) for different devices in the same system. There are 15 devices in a system at most. The RTA can be selected from the RTA code set and a device's RTA can be modified many times by using set run-time address command.

**SRTA Timing**

| 1 | 8 | 16 | 8 | 2 | 1 | 2 |
|---|---|----|---|---|---|---|
| SB | 0xE8 | DA | RTA | HD | A | DH |

2)  Read command: It is used to read data from device. It has byte, half word and word operation. When the device receives the command, they shall check if the command's RTA matches their own RTA.

**Read Timing**

| 1 | 8 | 8 | 9 | 2 | 9/18/36 | 2 |
|---|---|---|---|---|---------|---|
| SB | RD | RTA | ADDR (C) | HD | DATA (C) | DH |

Write command: It is used to write data to the devices. It has byte, half word and word operation. When the device receives the command, they shall check if the command's RTA matches their own RTA.

**Write Timing**

| 1 | 8 | 8 | 9 | 9/18/36 | 2 | 1 | 2 |
|---|---|---|---|---------|---|---|---|
| SB | WR | RTA | ADDR (C) | DATA (C) | HD | A | DH |

## 7.5. CIR Receiver

### 7.5.1. Overview

For saving CPU resource, CIR receiver is implemented in hardware. The CIR receiver samples the input signal on the programmable frequency and records these samples into RX FIFO when one CIR signal is found on the air. The CIR receiver uses Run-Length Code (RLC) to encode pulse width. The encoded data is buffered in a 64 levels and 8-bit width RX FIFO; the MSB bit is used to record the polarity of the receiving CIR signal. The high level is represented as '1' and the low level is represented as '0'. The rest 7 bits are used for the length of RLC. The maximum length is 128. If the duration of one level (high or low level) is more than 128, another byte is used.

In the air, there is always some noise. One threshold can be set to filter the noise to reduce system loading and improve the system stability.

The CIR includes the following features:

- Full physical layer implementation
- Support CIR for remote control
- 64x8 bits FIFO for data buffer
- Programmable FIFO thresholds

### 7.5.2. CIR Receiver Register List

| Module Name | Base Address |
|---|---|
| CIR | 0x01F02000 |

| Register Name | Offset | Description |
|---|---|---|
| CIR_CTL | 0x00 | CIR Control Register |
| CIR_RXCTL | 0x10 | CIR Receiver Configure Register |
| CIR_RXFIFO | 0x20 | CIR Receiver FIFO Register |
| CIR_RXINT | 0x2C | CIR Receiver Interrupt Control Register |
| CIR_RXSTA | 0x30 | CIR Receiver Status Register |
| CIR_CONFIG | 0x34 | CIR Configure Register |

## 7.5.3. CIR Receiver Register Description

### 7.5.3.1. CIR Receiver Control Register(Default Value: 0x0000_0000)

| Offset: 0x0000 | | | Register Name: **CIR_CTL** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:9 | / | / | / |
| 8 | R/W | 0 | CGPO<br>General Program Output (GPO) Control in CIR mode for TX Pin<br>0: Low level<br>1: High level |
| 7:6 | / | / | / |
| 5:4 | R/W | 0 | CIR ENABLE<br>00~10: Reserved<br>11: CIR mode enable |
| 3:2 | / | / | /. |
| 1 | R/W | 0 | RXEN<br>Receiver Block Enable<br>0: Disable<br>1: Enable |
| 0 | R/W | 0 | GEN<br>Global Enable<br>A disable on this bit overrides any other block or channel enables and flushes all FIFOs.<br>0: Disable<br>1: Enable |

### 7.5.3.2. CIR Receiver Configure Register(Default Value: 0x0000_0004)

| Offset: 0x0010 | | | Register Name: **CIR_RXCTL** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:3 | / | / | / |
| 2 | R/W | 1 | RPPI<br>Receiver Pulse Polarity Invert<br>0: Not invert receiver signal<br>1: Invert receiver signal |
| 1:0 | / | / | / |

### 7.5.3.3. CIR Receiver FIFO Register(Default Value: 0x0000_0000)

| Offset: 0x0020 | | | Register Name: **CIR_RXFIFO** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:8 | / | / | / |
| 7:0 | R | 0 | Receiver Byte FIFO |

### 7.5.3.4. CIR Receiver Interrupt Control Register(Default Value: 0x0000_0000)

| Offset: 0x002C | | | Register Name: **CIR_RXINT** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:14 | / | / | / |
| 13:8 | R/W | 0 | RAL<br>RX FIFO Available Received Byte Level for interrupt and DMA request<br>TRIGGER_LEVEL = RAL + 1 |
| 5 | R/W | 0 | DRQ_EN<br>RX FIFO DMA Enable<br>0: Disable<br>1: Enable<br>When set to '1', the Receiver FIFO DRQ is asserted if reaching RAL. The DRQ is de-asserted when condition fails. |
| 4 | R/W | 0 | RAI_EN<br>RX FIFO Available Interrupt Enable<br>0: Disable<br>1: Enable<br>When set to '1', the Receiver FIFO IRQ is asserted if reaching RAL. The IRQ is de-asserted when condition fails. |
| 3:2 | / | / | / |
| 1 | R/W | 0 | RPEI_EN<br>Receiver Packet End Interrupt Enable<br>0: Disable<br>1: Enable |
| 0 | R/W | 0 | ROI_EN<br>Receiver FIFO Overrun Interrupt Enable<br>0: Disable<br>1: Enable |

### 7.5.3.5. CIR Receiver Status Register(Default Value: 0x0000_0000)

| Offset: 0x0030 | | | Register Name: **CIR_RXSTA** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:15 | / | / | / |

| | | | RAC |
|---|---|---|---|
| | | | RX FIFO Available Counter |
| | | | 0: No available data in RX FIFO |
| | | | 1: 1 byte available data in RX FIFO |
| | | | 2: 2 byte available data in RX FIFO |
| | | | … |
| 14:8 | R | 0 | 64: 64 byte available data in RX FIFO |
| | | | STAT |
| | | | Status of CIR |
| | | | 0x0 – Idle |
| 7 | R | 0x0 | 0x1 – busy |
| 6:5 | / | / | / |
| | | | RA |
| | | | RX FIFO Available |
| | | | 0: RX FIFO not available according its level |
| | | | 1: RX FIFO available according its level |
| 4 | R/W | 0 | This bit is cleared by writing a '1'. |
| 3:2 | / | / | / |
| | | | RPE |
| | | | Receiver Packet End Flag |
| | | | 0: STO was not detected. In CIR mode, one CIR symbol is receiving or not detected. |
| | | | 1: STO field or packet abort symbol (7'b0000,000 and 8'b0000,0000 for MIR and FIR) is detected. In CIR mode, one CIR symbol is received. |
| 1 | R/W | 0 | This bit is cleared by writing a '1'. |
| | | | ROI |
| | | | Receiver FIFO Overrun |
| | | | 0: Receiver FIFO not overrun |
| | | | 1: Receiver FIFO overrun |
| 0 | R/W | 0 | This bit is cleared by writing a '1'. |

### 7.5.3.6.   CIR Receiver Configure Register(Default Value: 0x0000_0000)

| Offset: 0x0034 | | | Register Name: **CIR_RCR** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:25 | / | / | / |
| | | | SCS2 |
| | | | Bit2 of Sample Clock Select for CIR |
| 24 | R/W | 0x0 | This bit is defined by SCS bits below. |
| | | | ATHC |
| | | | Active Threshold Control for CIR |
| | | | 0x0 –ATHR in Unit of (Sample Clock) |
| 23 | R/W | 0x0 | 0x1 –ATHR in Unit of (128*Sample Clocks) |
| 22:16 | R/W | 0x0 | ATHR |

| | | | |
|---|---|---|---|
| | | | Active Threshold for CIR<br><br>These bits control the duration of CIR from Idle to Active State. The duration can be calculated by ((ATHR + 1)*(ATHC? Sample Clock: 128*Sample Clock)). |
| 15:8 | R/W | 0x18 | ITHR<br><br>Idle Threshold for CIR<br><br>The Receiver uses it to decide whether the CIR command has been received. If there is no CIR signal on the air, the receiver is staying in IDLE status. One active pulse will bring the receiver from IDLE status to Receiving status. After the CIR is end, the inputting signal will keep the specified level (high or low level) for a long time. The receiver can use this idle signal duration to decide that it has received the CIR command. The corresponding flag is asserted. If the corresponding interrupt is enable, the interrupt line is asserted to CPU. When the duration of signal keeps one status (high or low level) for the specified duration ( (ITHR + 1)*128 sample_clk ), this means that the previous CIR command has been finished. |
| 7:2 | R/W | 0xa | NTHR<br><br>Noise Threshold for CIR<br><br>When the duration of signal pulse (high or low level) is less than NTHR, the pulse is taken as noise and should be discarded by hardware.<br><br>0: all samples are recorded into RX FIFO<br>1: If the signal is only one sample duration, it is taken as noise and discarded.<br>2: If the signal is less than (<=) two sample duration, it is taken as noise and discarded.<br>…<br>61: if the signal is less than (<=) sixty-one sample duration, it is taken as noise and discarded. |
| 1:0 | R/W | 0 | SCS<br><br>Sample Clock Select for CIR<br><br>(see table below) |

| SCS2 | SCS[1] | SCS[0] | Sample Clock |
|---|---|---|---|
| 0 | 0 | 0 | ir_clk/64 |
| 0 | 0 | 1 | ir_clk/128 |
| 0 | 1 | 0 | ir_clk/256 |
| 0 | 1 | 1 | ir_clk/512 |
| 1 | 0 | 0 | ir_clk |
| 1 | 0 | 1 | Reserved |
| 1 | 1 | 0 | Reserved |
| 1 | 1 | 1 | Reserved |

# 7.6.  USB

## 7.6.1.  USB Block Diagram
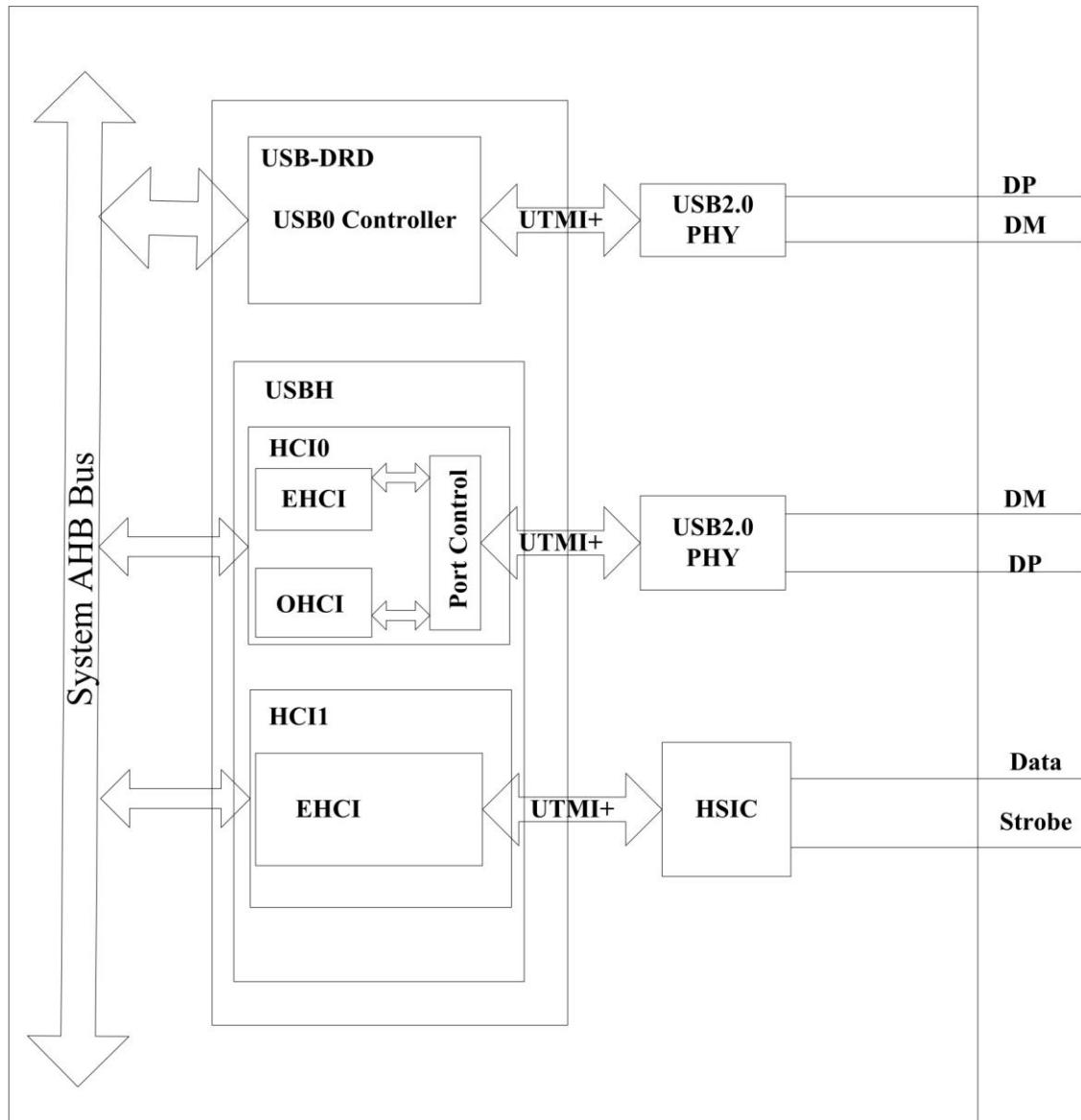


Figure 7-7. USB Block Diagram

## 7.6.2.  USB DRD Controller

The USB2.0 controller has following features:

- Support Device or Host operation at a time
- Supports High-Speed (HS, 480-Mbps), Full-Speed (FS, 12-Mbps), and Low-Speed (LS, 1.5-Mbps) in Host mode and support High-Speed (HS, 480-Mbps), Full-Speed (FS, 12-Mbps) in Device mode
- Supports the UTMI+ Level 3 interface . The 8-bit bidirectional data buses are used
- Supports bi-directional endpoint0 for Control transfer
- Supports up to 10 User-Configurable Endpoints for Bulk , Isochronousl and Interrupt bi-directional transfers (Endpoint1, Endpoint2, Endpoint3, Endpoint4, Endpoint5)
- Supports up to 8128Bytes (8KB-64B) FIFO for EPs (Excluding EP0)
- Supports High-Bandwidth Isochronous & Interrupt transfers
- Automated splitting/combining of packets for Bulk transfers
- Supports point-to-point and point-to-multipoint transfer in both Host and Peripheral mode
- Includes automatic ping capabilities
- Soft connect/disconnect function
- Performs all transaction scheduling in hardware
- Power Optimization and Power Management capabilities
- Includes interface to an external Normal DMA controller for every EPs

## 7.6.3.  USB Host Controller

### 7.6.3.1.  Overview

USB Host Controller is fully compliant with the USB 2.0 specification, Enhanced Host Controller Interface (EHCI) Specification, Revision 1.0, and the Open Host Controller Interface (OHCI) Specification Release 1.0a. The controller supports high-speed, 480-Mbps transfers (40 times faster than USB 1.1 full-speed mode)   using an EHCI Host Controller, as well as full and low speeds through one or more integrated OHCI Host   Controllers.

The USB host controller includes the following features:

- Supports industry-standard AMBA High-Performance Bus (AHB) and it is fully compliant with the AMBA Specification, Revision 2.0. Supports bus.
- Supports 32-bit Little Endian AMBA AHB Slave Bus for Register Access.
- Supports 32-bit Little Endian AMBA AHB Master Bus for Memory Access.
- Including an internal DMA Controller for data transfer with memory.
- Complies with Enhanced Host Controller Interface (EHCI) Specification, Version 1.0, and the Open Host Controller Interface (OHCI) Specification, Version 1.0a.
- Supports High-Speed (HS, 480-Mbps), Full-Speed (FS, 12-Mbps), and Low-Speed (LS, 1.5-Mbps) Device.
- Supports the UTMI+ Level 3 interface . The 8-bit bidirectional data buses are used.
- Supports only 1 USB Root Port shared between EHCI and OHCI.

**7.6.3.2. Block Diagram**
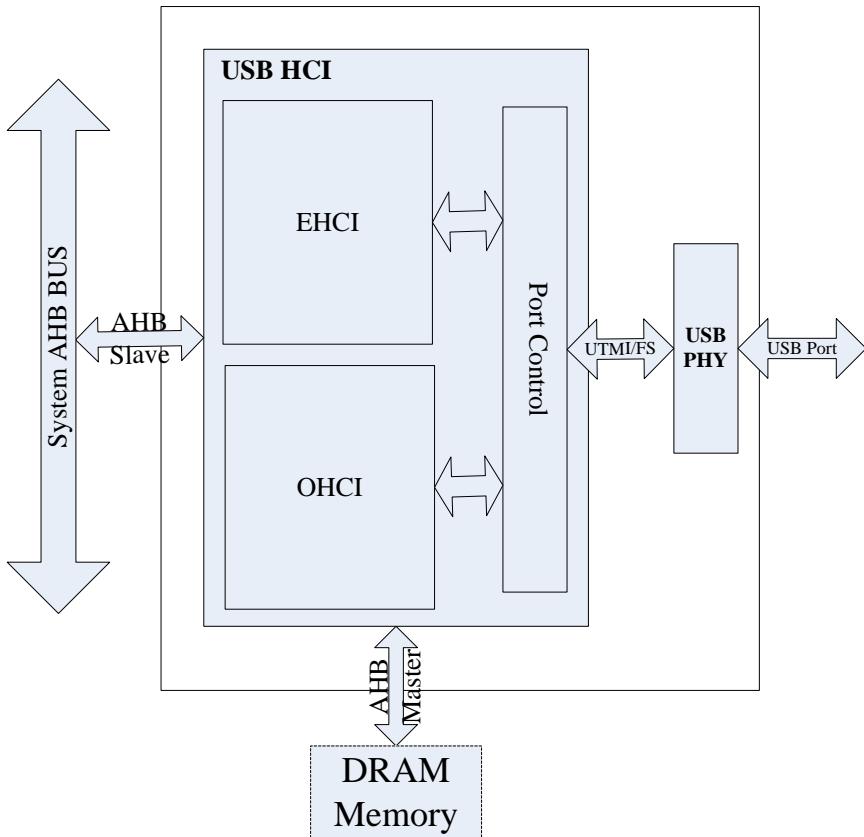


Figure 7-8. USB host controller System-Level block diagram

**7.6.3.3. USB Host Timing Diagram**

Please refer USB2.0 Specification, Enhanced Host Controller Interface (EHCI) Specification, Version 1.0, and the Open Host Controller Interface (OHCI) Specification, Version 1.0a.

**7.6.3.4. USB Host Register List**

| Module Name | Base Address |
|---|---|
| USB_HCI0 | 0x01C1A000 |
| USB_HCI1 | 0x01C1B000 |

| Register Name | Offset | Description |
|---|---|---|
| **EHCI Capability Register** | | |
| E_CAPLENGTH | 0x000 | EHCI Capability register Length Register |
| E_HCIVERSION | 0x002 | EHCI Host Interface Version Number Register |
| E_HCSPARAMS | 0x004 | EHCI Host Control Structural Parameter Register |
| E_HCCPARAMS | 0x008 | EHCI Host Control Capability Parameter Register |
| E_HCSPPORTROUTE | 0x00c | EHCI Companion Port Route Description |
| **EHCI Operational Register** | | |
| E_USBCMD | 0x010 | EHCI USB Command Register |
| E_USBSTS | 0x014 | EHCI USB Status Register |
| E_USBINTR | 0x018 | EHCI USB Interrupt Enable Register |
| E_FRINDEX | 0x01c | EHCI USB Frame Index Register |
| E_CTRLDSSEGMENT | 0x020 | EHCI 4G Segment Selector Register |
| E_PERIODICLISTBASE | 0x024 | EHCI Frame List Base Address Register |
| E_ASYNCLISTADDR | 0x028 | EHCI Next Asynchronous List Address Register |
| E_CONFIGFLAG | 0x050 | EHCI Configured Flag Register |
| E_PORTSC | 0x054 | EHCI Port Status/Control Register |
| **OHCI Control and Status Partition Register** | | |
| O_HcRevision | 0x400 | OHCI Revision Register |
| O_HcControl | 0x404 | OHCI Control Register |
| O_HcCommandStatus | 0x408 | OHCI Command Status Register |
| O_HcInterruptStatus | 0x40c | OHCI Interrupt Status Register |
| O_HcInterruptEnable | 0x410 | OHCI Interrupt Enable Register |
| O_HcInterruptDisable | 0x414 | OHCI Interrupt Disable Register |
| **OHCI Memory Pointer Partition Register** | | |
| O_HcHCCA | 0x418 | OHCI HCCA Base |
| O_HcPeriodCurrentED | 0x41c | OHCI Period Current ED Base |
| O_HcControlHeadED | 0x420 | OHCI Control Head ED Base |
| O_HcControlCurrentED | 0x424 | OHCI Control Current ED Base |
| O_HcBulkHeadED | 0x428 | OHCI Bulk Head ED Base |
| O_HcBulkCurrentED | 0x42c | OHCI Bulk Current ED Base |
| O_HcDoneHead | 0x430 | OHCI Done Head Base |
| **OHCI Frame Counter Partition Register** | | |
| O_HcFmInterval | 0x434 | OHCI Frame Interval Register |
| O_HcFmRemaining | 0x438 | OHCI Frame Remaining Register |
| O_HcFmNumber | 0x43c | OHCI Frame Number Register |
| O_HcPerioddicStart | 0x440 | OHCI Periodic Start Register |
| O_HcLSThreshold | 0x444 | OHCI LS Threshold Register |
| **OHCI Root Hub Partition Register** | | |
| O_HcRhDescriptorA | 0x448 | OHCI Root Hub Descriptor Register A |
| O_HcRhDesriptorB | 0x44c | OHCI Root Hub Descriptor Register B |
| O_HcRhStatus | 0x450 | OHCI Root Hub Status Register |
| O_HcRhPortStatus | 0x454 | OHCI Root Hub Port Status Register |

### 7.6.3.5. EHCI Register Description

#### 7.6.3.5.1. EHCI Identification Register(Default Value: Implementation Dependent)

| Offset: 0x0000 | | | Register Name: **CAPLENGTH** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 7:0 | R | 0x10 | CAPLENGTH<br>The value in these bits indicates an offset to add to register base to find the beginning of the Operational Register Space. |

#### 7.6.3.5.2. EHCI Host Interface Version Number Register(Default Value: 0x0100)

| Offset: 0x0002 | | | Register Name: **HCIVERSION** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 15:0 | R | 0x0100 | HCIVERSION<br>This is a 16-bits register containing a BCD encoding of the EHCI revision number supported by this host controller. The most significant byte of this register represents a major revision and the least significant byte is the minor revision. |

#### 7.6.3.5.3. EHCI Host Control Structural Parameter Register(Default Value: Implementation Dependent)

| Offset: 0x0004 | | | Register Name: **HCSPARAMS** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:24 | / | 0 | Reserved.<br>These bits are reserved and should be set to zero. |
| 23:20 | R | 0 | Debug Port Number<br>This register identifies which of the host controller ports is the debug port. The value is the port number (one based) of the debug port.<br>This field will always be '0'. |
| 19:16 | / | 0 | Reserved.<br>These bits are reserved and should be set to zero. |
| 15:12 | R | 0 | Number of Companion Controller (N_CC)<br>This field indicates the number of companion controllers associated with this USB2.0 host controller. A zero in this field indicates there are no companion host controllers. And a value larger than zero in this field indicates there are companion USB1.1 host controller(s).<br>This field will always be '0'. |

| | | | |
|---|---|---|---|
| 11:8 | R | 0 | Number of    Port per Companion Controller(N_PCC)<br>This field indicates the number of ports supported per companion host controller host controller. It is used to indicate the port routing configuration to system software.<br>This field will always fix with '0'. |
| 7 | R | 0 | Port Routing Rules<br>This field indicates the method used by this implementation for how all ports are mapped to companion controllers. The value of this field has the following interpretation:<br><br>| Value | Meaning |<br>|---|---|<br>| 0 | The first N_PCC ports are routed to the lowest numbered function companion host controller, the next N_PCC port are routed to the next lowest function companion controller, and so on. |<br>| 1 | The port routing is explicitly enumerated by the first N_PORTS elements of the HCSP-PORTTOUTE array. |<br><br>This field will always be '0'. |
| 6:4 | / | 0 | Reserved.<br>These bits are reserved and should be set to zero. |
| 3:0 | R | 1 | N_PORTS<br>This field specifies the number of physical downstream ports implemented on this host controller. The value of this field determines how many port registers are addressable in the Operational Register Space. Valid values are in the range of 0x1 to 0x0f.<br>This field is always 1. |

### 7.6.3.5.4. EHCI Host Control Capability Parameter Register(Default Value: Implementation Dependent)

| Offset: 0x0008 | | | Register Name: **HCCPARAMS** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:16 | / | 0 | Reserved<br>These bits are reserved and should be set to zero. |
| 15:18 | R | 0 | EHCI Extended Capabilities Pointer (EECP)<br>This optional field indicates the existence of a capabilities list. A value of 00b indicates no extended capabilities are implemented. A non-zero value in this register indicates the offset in PCI configuration space of the first EHCI extended capabiliby. The pointer value must be 40h or greater if implemented to maintain to consistency of the PCI header defined for this calss of device.<br>The value of this field is always '00b'. |
| 7:4 | R | | Isochronous Scheduling Threshold<br>This field indicates, relative to the current position of the executing host controller, where software can reliably update the isochronous schedule. When bit[7] is zero, the value of the least significant 3 bits indicates the |

| | | | number of micro-frames a host controller can hold a set of isochronous data structures(one or more) before flushing the state. When bit[7] is a one, then host software assumes the host controller may cache an isochronous data structure for an entire frame. |
|---|---|---|---|
| 3 | R | 0 | Reserved<br>These bits are reserved and should be set to zero. |
| 2 | R | | Asynchronous Schedule Park Capability<br>If this bit is set to a one, then the host controller supports the park feature for high-speed queue heads in the Asynchronous Schedule. The 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 | R | | Programmable Frame List Flag<br>If this bit is set to a zero, then system software must use a frame list length of 1024 elements with this host controller.The USBCMD register<br>Frame List Size field is a read-only register and should be set to zero.<br>If set to 1,then system software can specify and use the frame list in the USBCMD register Frame List Size field to cofigure the host controller.<br>The frame list must always aligned on a 4K page boundary.This requirement ensures that the frame list is always physically contiguous. |
| 0 | R | 0 | Reserved<br>These bits are reserved for future use and should return a value of zero when read. |

### 7.6.3.5.5. EHCI Companion Port Route Description (Default Value: UNDEFINED)

| Offset: 0x000C | | | Register Name: **HCSP-PORTROUTE** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R | | HCSP-PORTROUTE<br>This optional field is valid only if Port Routing Rules field in HCSPARAMS register is set to a one.<br>This field is used to allow a host controller implementation to explicitly describe to which companion host controller each implemented port is mapped. This field is a 15-element nibble array (each 4 bit is one array element). Each array location corresponds one-to-one with a physical port provided by the host controller (e.g. PORTROUTE [0] corresponds to the first PORTSC port, PORTROUTE [1] to the second PORTSC port, etc.). The value of each element indicates to which of the companion host controllers this port is routed. Only the first N_PORTS elements have valid information. A value of zero indicates that the port is routed to the lowest numbered function companion host controller. A value of one indicates that the port is routed to the next lowest numbered function companion host controller, and so on. |

### 7.6.3.5.6. EHCI USB Command Register (Default Value: 0x00080000,0x00080B00 if Asynchronous Schedule Park Capability is a one)

| Offset: 0x0010 | | | Register Name: **USBCMD** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:24 | / | 0 | Reserved<br>These bits are reserved and should be set to zero. |
| 23:16 | R/W | 0x08 | Interrupt Threshold Control<br>The value in this field is used by system software to select the maximum rate at which the host controller will issue interrupts. The only valid values are defined below:<br><br>Value / Minimum Interrupt Interval<br>0x00 Reserved<br>0x01 1 micro-frame<br>0x02 2 micro-frame<br>0x04 4 micro-frame<br>0x08 8 micro-frame(default, equates to 1 ms)<br>0x10 16 micro-frame(2ms)<br>0x20 32 micro-frame(4ms)<br>0x40 64 micro-frame(8ms)<br><br>Any other value in this register yields undefined results.<br>The default value in this field is 0x08 .<br>Software modifications to this bit while HC Halted bit is equal to zero results in undefined behavior. |
| 15:12 | / | 0 | Reserved<br>These bits are reserved and should be set to zero. |
| 11 | R/W or R | 0 | Asynchronous Schedule Park Mode Enable(OPTIONAL)<br>If the Asynchronous Park Capability bit in the HCCPARAMS register is a one, then this bit defaults to a 1 and is R/W. Otherwise the bit must be a zero and is Read Only. Software uses this bit to enable or disable Park mode. When this bit is one, Park mode is enabled. When this bit is zero, Park mode is disabled. |
| 10 | / | 0 | Reserved<br>These bits are reserved and should be set to zero. |
| 9:8 | R/W or R | 0 | Asynchronous Schedule Park Mode Count(OPTIONAL)<br>Asynchronous Park Capability bit in the HCCPARAMS register is a one,<br>Then this field defaults to 0x3 and is W/R. Otherwise it defaults to zero and is R. It contains a count of the number of successive transactions the host controller is allowed to execute from a high-speed queue head on the Asynchronous schedule before continuing traversal of the Asynchronous schedule.<br>Valid value are 0x1 to 0x3.Software must not write a zero to this bit when Park Mode Enable is a one as it will result in undefined behavior. |

| | | | |
|---|---|---|---|
| 7 | R/W | 0 | Light Host Controller Reset(OPTIONAL)<br>This control bit is not required.<br>If implemented, it allows the driver to reset the EHCI controller without affecting the state of the ports or relationship to the companion host controllers. For example, the PORSTC registers should not be reset to their default values and the CF bit setting should not go to zero (retaining port ownership relationships).<br>A host software read of this bit as zero indicates the Light Host Controller Reset has completed and it si safe for software to re-initialize the host controller. A host software read of this bit as a one indicates the Light Host |
| 6 | R/W | 0 | Interrupt on Async Advance Doorbell<br>This bit is used as a doorbell by software to tell the host controller to issue an interrupt the next time it advances asynchronous schedule. Soft-Ware must write a 1 to this bit to ring the doorbell.<br>When the host controller has evicted all appropriate cached schedule state, it sets the Interrupt on Async Advance status bit in the USBSTS. if the Interrupt on Async Advance Enable bit in the USBINTR register is a one then the host controller will assert an interrupt at the next interrupt threshold. The host controller sets this bit to a zero after it has set the Interrupt on Async Advance status bit in the USBSTS register to a one.<br>Software should not write a one to this bit when the asynchronous schedule is disabled. Doing so will yield undefined results. |

| | | | |
|---|---|---|---|
| 5 | R/W | 0 | Asynchronous Schedule Enable<br>This bit controls whether the host controller skips processing the Asynchronous Schedule. Values mean:<br><br>| Bit Value | Meaning |<br>|---|---|<br>| 0 | Do not process the Asynchronous Schedule. |<br>| 1 | Use the ASYNLISTADDR register to access the Asynchronous Schedule. |<br><br>The default value of this field is '0b'. |
| 4 | R/W | 0 | Periodic Schedule Enable<br>This bit controls whether the host controller skips processing the Periodic Schedule. Values mean:<br><br>| Bit Value | Meaning |<br>|---|---|<br>| 0 | Do not process the Periodic Schedule. |<br>| 1 | Use the PERIODICLISTBASE register to access the Periodic Schedule. |<br><br>The default value of this field is '0b'. |
| 3:2 | R/W  or R | 0 | Frame List Size<br>This field is R/W only if Programmable Frame List Flag in the HCCPARAMS registers is set to a one. This field specifies the size of the Frame list. The size the frame list controls which bits in the Frame Index Register should be used for the Frame List Current index. Values mean:<br><br>| Bits | Meaning |<br>|---|---|<br>| 00b | 1024 elements(4096bytes)Default value | |

| | | | |
|---|---|---|---|
| | | | 01b | 512 elements(2048byts) |
| | | | 10b | 256 elements(1024bytes)For resource-constrained condition |
| | | | 11b | reserved |
| | | | The default value is '00b'. |
| 1 | R/W | 0 | Host Controller Reset<br>This control bit is used by software to reset the host controller. The effects of this on Root Hub registers are similar to a Chip Hardware Reset.<br>When software writes a one to this bit, the Host Controller resets its internal pipelines, timers, counters, state machines, etc. to their initial value. Any transaction currently in progress on USB is immediately terminated. A USB reset is not driven on downstream ports.<br>All operational registers, including port registers and port state machines are set to their initial values. Port ownership reverts to the companion host controller(s). Software must reinitialize the host controller as described in Section 4.1 of the CHEI Specification in order to return the host controller to an operational state.<br>This bit is set to zero by the Host Controller when the reset process is complete. Software cannot terminate the reset process early by writing a zero to this register.<br>Software should not set this bit to a one when the HC Halted bit in the USBSTS register is a zero. Attempting to reset an actively running host controller will result in undefined behavior. |
| 0 | R/W | 0 | Run/Stop<br>When set to a 1, the Host Controller proceeds with execution of the schedule. When set to 0, the Host Controller completes the current and any actively pipelined transactions on the USB and then halts. The Host Controller must halt within 16 micro-frames after software clears this bit. The HC Halted bit indicates when the Host Controller has finished its pending pipelined transactions and has entered the stopped state.<br>Software must not write a one to this field unless the Host Controller is in the Halt State.<br>The default value is 0x0. |

### 7.6.3.5.7. EHCI USB Status Register (Default Value: 0x00001000)

| Offset: 0x0014 | | | Register Name: **USBSTS** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:16 | / | 0 | Reserved<br>These bits are reserved and should be set to zero. |
| 15 | R | 0 | Asynchronous Schedule Status<br>The bit reports the current real status of Asynchronous Schedule. If this bit is a zero then the status of the Asynchronous Schedule is disabled. If this bit is a one then the status of the Asynchronous Schedule is enabled. The Host |

| | | | |
|---|---|---|---|
| | | | Controller is not required to immediately disable or enable the Asynchronous Schedule when software transitions the Asynchronous Schedule Enable bit in the USBCMD register. When this bit and the Asynchronous Schedule Enable bit are the same value, the Asynchronous Schedule is either enabled (1) or disabled (0). |
| 14 | R | 0 | Periodic Schedule Status<br>The bit reports the current real status of the Periodic Schedule. If this bit is a zero then the status of the Periodic Schedule is disabled. If this bit is a one then the status of the Periodic Schedule is enabled. The Host Controller is not required to immediately disable or enable the Periodic Schedule when software transitions the Periodic Schedule Enable bit in the USBCMD register. When this bit and the Periodic Schedule Enable bit are the same value, the Periodic Schedule is either enabled (1) or disabled (0). |
| 13 | R | 0 | Reclamation<br>This is a read-only status bit, which is used to detect an empty asynchronous schedule. |
| 12 | R | 1 | HC Halted<br>This bit is a zero whenever the Run/Stop bit is a one. The Host Controller Sets this bit to one after it has stopped executing as a result of the Run/Stop bit being set to 0, either by software or by the Host Controller Hardware (e.g. internal error).<br>The default value is '1'. |
| 11:6 | / | 0 | Reserved<br>These bits are reserved and should be set to zero. |
| 5 | R/WC | 0 | Interrupt on Async Advance<br>System software can force the host controller to issue an interrupt the next time the host controller advances the asynchronous schedule by writing a one to the Interrupt on Async Advance Doorbell bit in the USBCMD register. This status bit indicates the assertion of that interrupt source. |
| 4 | R/WC | 0 | Host System Error<br>The Host Controller set this bit to 1 when a serious error occurs during a host system access involving the Host Controller module. When this error occurs, the Host Controller clears the Run/Stop bit in the Command register to prevent further execution of the scheduled TDs. |
| 3 | R/WC | 0 | Frame List Rollover<br>The Host Controller sets this bit to a one when the Frame List Index rolls over from its maximum value to zero. The exact value at which the rollover occurs depends on the frame list size. For example, if the frame list size is 1024, the Frame Index Register rolls over every time FRINDEX [13] toggles. Similarly, if the size is 512, the Host Controller sets this bit to a one every time FRINDEX [12] toggles. |
| 2 | R/WC | 0 | Port Change Detect<br>The Host Controller sets this bit to a one when any port for which the Port Owner bit is set to zero has a change bit transition from a zero to a one or a Force Port Resume bit transition from a zero to a one as a result of a J-K |

| | | | transition detected on a suspended port. This bit will also be set as a result of the Connect Status Chang being set to a one after system software has relinquished ownership of a connected port by writing a one to a port's Port Owner bit. |
|---|---|---|---|
| 1 | R/WC | 0 | USB Error Interrupt(USBERRINT)<br>The Host Controller sets this bit to 1 when completion of USB transaction results in an error condition(e.g. error counter underflow).If the TD on which the error interrupt occurred also had its IOC bit set, both.<br>This bit and USBINT bit are set. |
| 0 | R/WC | 0 | USB Interrupt(USBINT)<br>The Host Controller sets this bit to a one on the completion of a USB transaction, which results in the retirement of a Transfer Descriptor that had its IOC bit set.<br>The Host Controller also sets this bit to 1 when a short packet is detected (actual number of bytes received was less than the expected number of bytes) |

**7.6.3.5.8. EHCI USB Interrupt Enable Register (Default Value: 0x00000000)**

| Offset: 0x0018 | | | Register Name: **USBINTR** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:6 | / | 0 | Reserved<br>These bits are reserved and should be zero. |
| 5 | R/W | 0 | Interrupt on Async Advance Enable<br>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. The interrupt is acknowledged by software clearing the Interrupt on Async Advance bit. |
| 4 | R/W | 0 | Host System Error Enable<br>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. The interrupt is acknowledged by software clearing the Host System Error bit. |
| 3 | R/W | 0 | Frame List Rollover Enable<br>When this bit is 1, and the Frame List Rollover bit in the USBSTS register is 1, the host controller will issue an interrupt. The interrupt is acknowledged by software clearing the Frame List Rollover bit. |
| 2 | R/W | 0 | Port Change Interrupt Enable<br>When this bit is 1, and the Port Chang Detect bit in the USBSTS register is 1, the host controller will issue an interrupt. The interrupt is acknowledged by software clearing the Port Chang Detect bit. |
| 1 | R/W | 0 | USB Error Interrupt Enable<br>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.<br>The interrupt is acknowledged by software clearing the USBERRINT bit. |

| | | | 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. |
| 0 | R/W | 0 | The interrupt is acknowledged by software clearing the USBINT bit |

### 7.6.3.5.9. EHCI Frame Index Register (Default Value: 0x00000000)

| Offset: 0x001C | | | Register Name: **FRINDEX** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:14 | / | 0 | Reserved These bits are reserved and should be zero. |
| 13:0 | R/W | 0 | Frame Index The value in this register increment at the end of each time frame (e.g. micro-frame).Bits[N:3] are used for the Frame List current index. It Means that each location of the frame list is accessed 8 times(frames or Micro-frames) before moving to the next index. The following illustrates Values of N based on the value of the Frame List Size field in the USBCMD register. <br><br> |
| USBCMD[Frame List Size] | Number Elements | N |
|---|---|---|
| 00b | 1024 | 12 |
| 01b | 512 | 11 |
| 10b | 256 | 10 |
| 11b | Reserved | |

Note: This register must be written as a DWord. Byte writes produce undefined results.

### 7.6.3.5.10.     EHCI Periodic Frame List Base Address Register (Default Value: Undefined)

| Offset: 0x0024 | | | Register Name: **PERIODICLISTBASE** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:12 | R/W | | Base Address These bits correspond to memory address signals [31:12], respectively. This register contains the beginning address of the Periodic Frame List in the system memory. System software loads this register prior to starting the schedule execution by the Host Controller. The memory structure referenced by this physical memory pointer is assumed to be 4-K byte aligned. The contents of this register are combined with the Frame Index Register (FRINDEX) to enable the Host Controller to step through the Periodic Frame List in sequence. |
| 11:0 | / | | Reserved Must be written as 0x0 during runtime, the values of these bits are undefined. |

Note: Writes must be Dword Writes.

#### 7.6.3.5.11.    EHCI Current Asynchronous List Address Register (Default Value: Undefined)

| Offset: 0x0028 | | | Register Name: **ASYNCLISTADDR** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:5 | R/W | | Link Pointer (LP)<br>This field contains the address of the next asynchronous queue head to be executed.<br>These bits correspond to memory address signals [31:5], respectively. |
| 4:0 | / | / | Reserved<br>These bits are reserved and their value has no effect on operation.<br>Bits in this field cannot be modified by system software and will always return a zero when read. |

Note: Write must be DWord Writes.

#### 7.6.3.5.12.    EHCI Configure Flag Register (Default Value: 0x00000000)

| Offset: 0x0050 | | | Register Name: **CONFIGFLAG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:1 | / | 0 | Reserved<br>These bits are reserved and should be set to zero. |
| 0 | R/W | 0 | Configure Flag(CF)<br>Host software sets this bit as the last action in its process of configuring the Host Controller. This bit controls the default port-routing control logic as follow:<br><table><tr><td>Value</td><td>Meaning</td></tr><tr><td>0</td><td>Port routing control logic default-routs each port to an implementation dependent classic host controller.</td></tr><tr><td>1</td><td>Port routing control logic default-routs all ports to this host controller.</td></tr></table>The default value of this field is '0'. |

Note: This register is not use in the normal implementation.

#### 7.6.3.5.13.    EHCI Port Status and Control Register (Default Value: 0x00002000(w/PPC set to one);0x00003000

#### (w/PPC set to a zero))

| Offset: 0x0054 | | | Register Name: **PORTSC** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |

| 31:22 | / | 0 | Reserved<br>These bits are reserved for future use and should return a value of zero when read. |
|---|---|---|---|
| 21 | R/W | 0 | Wake on Disconnect Enable(WKDSCNNT_E)<br>Writing this bit to a one enables the port to be sensitive to device disconnects as wake-up events.<br>This field is zero if Port Power is zero.<br>The default value in this field is '0'. |
| 20 | R/W | 0 | Wake on Connect Enable(WKCNNT_E)<br>Writing this bit to a one enable the port to be sensitive to device connects as wake-up events.<br>This field is zero if Port Power is zero.<br>The default value in this field is '0'. |
| 19:16 | R/W | 0 | Port Test Control<br>The value in this field specifies the test mode of the port. The encoding of the test mode bits are as follow:<br><br>Bits / Test Mode table:<br>0000b — The port is NOT operating in a test mode.<br>0001b — Test J_STATE<br>0010b — Test K_STATE<br>0011b — Test SE0_NAK<br>0100b — Test Packet<br>0101b — Test FORCE_ENABLE<br>0110b - 1111b — Reserved<br><br>The default value in this field is '0000b'. |
| 15:14 | R/W | 0 | Reserved<br>These bits are reserved for future use and should return a value of zero when read. |
| 13 | R/W | 1 | Port Owner<br>This bit unconditionally goes to a 0b when the Configured bit in the CONFIGFLAG register makes a 0b to 1b transition. This bit unconditionally goes to 1b whenever the Configured bit is zero.<br>System software uses this field to release ownership of the port to selected host controller (in the event that the attached device is not a high-speed device).Software writes a one to this bit when the attached device is not a high-speed device. A one in this bit means that a companion host controller owns and controls the port.<br>Default Value = 1b. |
| 12 | / | 0 | Reserved<br>These bits are reserved for future use and should return a value of zero when read. |
| 11:10 | R | 0 | Line Status<br>These bits reflect the current logical levels of the D+ (bit11) and D-(bit10) |

| | | | signal lines. These bits are used for detection of low-speed USB devices prior to port reset and enable sequence. This read only field is valid only when the port enable bit is zero and the current connect status bit is set to a one. The encoding of the bits are: |
|---|---|---|---|
| | | | <table><tr><td>Bit[11:10]</td><td>USB State</td><td>Interpretation</td></tr><tr><td>00b</td><td>SE0</td><td>Not Low-speed device, perform EHCI reset.</td></tr><tr><td>10b</td><td>J-state</td><td>Not Low-speed device, perform EHCI reset.</td></tr><tr><td>01b</td><td>K-state</td><td>Low-speed device, release ownership of port.</td></tr><tr><td>11b</td><td>Undefined</td><td>Not Low-speed device, perform EHCI reset.</td></tr></table> This value of this field is undefined if Port Power is zero. |
| 9 | / | 0 | Reserved<br>This bit is reserved for future use, and should return a value of zero when read. |
| 8 | R/W | 0 | Port Reset<br>1=Port is in Reset. 0=Port is not in Reset. Default value = 0.<br>When software writes a one to this bit (from a zero), the bus reset sequence as defined in the USB Specification Revision 2.0 is started. Software writes a zero to this bit to terminate the bus reset sequence. Software must keep this bit at a one long enough to ensure the reset sequence, as specified in the USB Specification Revision 2.0, completes. Notes: when software writes this bit to a one , it must also write a zero to the Port Enable bit.<br>Note that when software writes a zero to this bit there may be a delay before the bit status changes to a zero. The bit status will not read as a zero until after the reset has completed. If the port is in high-speed mode after reset is complete, the host controller will automatically enable this port (e.g. set the Port Enable bit to a one). A host controller must terminate the reset and stabilize the state of the port within 2 milliseconds of software transitioning this bit from a one to a zero. For example: if the port detects that the attached device is high-speed during reset, then the host controller must have the port in the enabled state with 2ms of software writing this bit to a zero.<br>The HC Halted bit in the USBSTS register should be a zero before software attempts to use this bit. The host controller may hold Port Reset asserted to a one when the HC Halted bit is a one.<br>This field is zero if Port Power is zero. |
| 7 | R/W | 0 | Suspend<br>Port Enabled Bit and Suspend bit of this register define the port states as follows:<br><table><tr><td>Bits[Port Enables, Suspend]</td><td>Port State</td></tr><tr><td>0x</td><td>Disable</td></tr><tr><td>10</td><td>Enable</td></tr></table> |

| | | | | |
|---|---|---|---|---|
| | | | 11 | Suspend |

When in suspend state, downstream propagation of data is blocked on this port, except for port reset. The blocking occurs at the end of the current transaction, if a transaction was in progress when this bit was written to 1. In the suspend state, the port is sensitive to resume detection. Not that the bit status does not change until the port is suspend and that there may be a delay in suspending a port if there is a transaction currently in progress on the USB.

A write of zero to this bit is ignored by the host controller. The host controller will unconditionally set this bit to a zero when:

① Software sets the Force Port Resume bit to a zero(from a one).

② Software sets the Port Reset bit to a one(from a zero).

If host software sets this bit to a one when the port is not enabled(i.e. Port enabled bit is a zero), the results are undefined.

This field is zero if Port Power is zero.

The default value in this field is '0'.

| | | | |
|---|---|---|---|
| | | | Force Port Resume<br><br>1 = Resume detected/driven on port. 0 = No resume (K-state) detected/ driven on port. Default value = 0.<br><br>This functionality defined for manipulating this bit depends on the value of the Suspend bit. For example, if the port is not suspend and software transitions this bit to a one, then the effects on the bus are undefined. Software sets this bit to a 1 drive resume signaling. The Host Controller sets this bit to a 1 if a J-to-K transition is detected while the port is in the Suspend state. When this bit transitions to a one because a J-to-K transition is detected, the Port Change Detect bit in the USBSTS register is also set to a one. If software sets this bit to a one, the host controller must not set the Port Change Detect bit.<br><br>Note that when the EHCI controller owns the port, the resume sequence follows the defined sequence documented in the USB Specification Revision 2.0. The resume signaling (Full-speed 'K') is driven on the port as long as this remains a one. Software must appropriately time the Resume and set this bit to a zero when the appropriate amount of time has elapsed. Writing a zero (from one) causes the port to return high-speed mode (forcing the bus below the port into a high-speed idle). This bit will remain a one until the port has switched to high-speed idle. The host controller must complete this transition within 2 milliseconds of software setting this bit to a zero. |
| 6 | R/W | 0 | This field is zero if Port Power is zero. |
| 5 | R/WC | 0 | Over-current Change<br><br>Default = 0. This bit gets set to a one when there is a change to Over-current Active. Software clears this bit by writing a one to this bit position. |
| 4 | R | 0 | Over-current Active<br><br>0 = This port does not have an over-current condition. 1 = This port currently has an over-current condition. This bit will automatically transition from a one to a zero when the over current condition is removed. |

| | | | The default value of this bit is '0'. |
|---|---|---|---|
| 3 | R/WC | 0 | Port Enable/Disable Change<br>Default = 0. 1 = Port enabled/disabled status has changed. 0 = No change.<br>For the root hub, this bit gets set to a one only when a port is disabled due to the appropriate conditions existing at the EOF2 point (See Chapter 11 of the USB Specification for the definition of a Port Error). Software clears this bit by writing a 1 to it.<br>This field is zero if Port Power is zero. |
| 2 | R/W | 0 | Port Enabled/Disabled<br>1=Enable, 0=Disable. Ports can only be enabled by the host controller as a part of the reset and enable. Software cannot enable a port by writing a one to this field. The host controller will only set this bit to a one when the reset sequence determines that the attached device is a high-speed device.<br>Ports can be disabled by either a fault condition(disconnect event or other fault condition) or by host software. Note that the bit status does not change until the port state actually changes. There may be a delay in disabling or enabling a port due to other host controller and bus events. When the port is disabled, downstream propagation of data is blocked on this port except for reset.<br>The default value of this field is '0'.<br>This field is zero if Port Power is zero. |
| 1 | R/WC | 0 | Connect Status Change<br>1=Change in Current Connect Status, 0=No change, Default=0.<br>Indicates a change has occurred in the port's Current Connect Status. The host controller sets this bit for all changes to the port device connect status, even if system software has not cleared an existing connect status change. For example, the insertion status changes twice before system software has cleared the changed condition, hub hardware will be "setting" an already-set bit. Software sets this bit to 0 by writing a 1 to it.<br>This field is zero if Port Power is zero. |
| 0 | R | 0 | Current Connect Status<br>Device is present on port when the value of this field is a one, and no device is present on port when the value of this field is a zero. This value reflects the current state of the port, and may not correspond directly to the event that caused the Connect Status Change(Bit 1) to be set.<br>This field is zero if Port Power zero. |

Note: This register is only reset by hardware or in response to a host controller reset.

### 7.6.3.6. OHCI Register Description

### 7.6.3.6.1. HcRevision Register(Default Value: 0x10)

| Offset: 0x400 | | | | Register Name: HcRevision |
|---|---|---|---|---|
| | Read/Write | | | |
| Bit | HCD | HC | Default/HEX | Description |
| 31:8 | / | / | 0x00 | Reserved |
| 7:0 | R | R | 0x10 | Revision<br>This read-only field contains the BCD representation of the version of the HCI specification that is implemented by this HC. For example, a value of 0x11 corresponds to version 1.1. All of the HC implementations that are compliant with this specification will have a value of 0x10. |

### 7.6.3.6.2. HcControl Register(Default Value: 0x0)

| Offset: 0x404 | | | | Register Name: HcRevision |
|---|---|---|---|---|
| | Read/Write | | | |
| Bit | HCD | HC | Default/HEX | Description |
| 31:11 | / | / | 0x00 | Reserved |
| 10 | R/W | R | 0x0 | RemoteWakeupEnable<br>This bit is used by HCD to enable or disable the remote wakeup feature upon the detection of upstream resume signaling. When this bit is set and the ResumeDetected bit in HcInterruptStatus is set, a remote wakeup is signaled to the host system. Setting this bit has no impact on the generation of hardware interrupt. |
| 9 | R/W | R/W | 0x0 | RemoteWakeupConnected<br>This bit indicates whether HC supports remote wakeup signaling. If remote wakeup is supported and used by the system, it is the responsibility of system firmware to set this bit during POST. HC clear the bit upon a hardware reset but does not alter it upon a software reset. Remote wakeup signaling of the host system is host-bus-specific and is not described in this specification. |
| 8 | R/W | R | 0x0 | InterruptRouting<br>This bit determines the routing of interrupts generated by events registered in HcInterruptStatus. If clear, all interrupt are routed to the normal host bus interrupt mechanism. If set interrupts are routed to the System Management Interrupt. HCD clears this bit upon a hardware reset, but it does not alter this bit upon a software reset. HCD uses this bit as a tag to indicate the ownership of HC. |
| 7:6 | R/W | R/W | 0x0 | HostControllerFunctionalState for USB<br><table><tr><td>00b</td><td>USBReset</td></tr></table> |

| | | | | | |
|---|---|---|---|---|---|
| | | | | 01b | USBResume |
| | | | | 10b | USBOperational |
| | | | | 11b | USBSuspend |
| | | | | A transition to USBOperational from another state causes SOF generation to begin 1 ms later. HCD may determine whether HC has begun sending SOFs by reading the StartoFrame field of HcInterruptStatus. This field may be changed by HC only when in the USBSUSPEND state. HC may move from the USBSUSPEND state to the USBRESUME state after detecting the resume signaling from a downstream port. HC enters USBSUSPEND after a software reset, whereas it enters USBRESET after a hardware reset. The latter also resets the Root Hub and asserts subsequent reset signaling to downstream ports. | |
| 5 | R/W | R | 0x0 | BulkListEnable This bit is set to enable the processing of the Bulk list in the next Frame. If cleared by HCD, processing of the Bulk list does not occur after the next SOF. HC checks this bit whenever it determines to process the list. When disabled, HCD may modify the list. If HcBulkCurrentED is pointing to an ED to be removed, HCD must advance the pointer by updating HcBulkCurrentED before re-enabling processing of the list. | |
| 4 | R/W | R | 0x0 | ControlListEnable This bit is set to enable the processing of the Control list in the next Frame. If cleared by HCD, processing of the Control list does not occur after the next SOF. HC must check this bit whenever it determines to process the list. When disabled, HCD may modify the list. If HcControlCurrentED is pointing to an ED to be removed, HCD must advance the pointer by updating HcControlCurrentED before re-enabling processing of the list. | |
| 3 | R/W | R | 0x0 | IsochronousEnable This bit is used by HCD to enable/disable processing of isochronous EDs. While processing the periodic list in a Frame, HC checks the status of this bit when it finds an Isochronous ED (F=1). If set (enabled), HC continues processing the EDs. If cleared (disabled), HC halts processing of the periodic list (which now contains only isochronous EDs) and begins processing the Bulk/Control lists. Setting this bit is guaranteed to take effect in the next Frame (not the current Frame). | |
| 2 | R/W | R | 0x0 | PeriodicListEnable This bit is set to enable the processing of periodic list in the next Frame. If cleared by HCD, processing of the periodic list does not occur after the next SOF. HC must check this bit before it starts processing the list. | |
| 1:0 | R/W | R | 0x0 | ControlBulkServiceRatio This specifies the service ratio between Control and Bulk EDs. Before processing any of the nonperiodic lists, HC must compare the ratio specified with its internal count on how many nonempty Control EDs have been processed, in determining whether to continue serving another | |

| 1 | | | 0x0 | Control ED or switching to Bulk EDs. The internal count will be retained when crossing the frame boundary. In case of reset, HCD is responsible for restoring this value. |

| CBSR | No. of Control EDs Over Bulk EDs Served |
|---|---|
| 0 | 1:1 |
| 1 | 2:1 |
| 2 | 3:1 |
| 3 | 4:1 |

The default value is 0x0.

### 7.6.3.6.3. HcCommandStatus Register(Default Value: 0x0)

| Offset: 0x408 | | | | Register Name: HcCommandStatus |
|---|---|---|---|---|
| | Read/Write | | | |
| Bit | HCD | HC | Default/HEX | Description |
| 31:18 | / | / | 0x0 | Reserved |
| 17:16 | R | R/W | 0x0 | SchedulingOverrunCount<br>These bits are incremented on each scheduling overrun error. It is initialized to 00b and wraps around at 11b. This will be incremented when a scheduling overrun is detected even if SchedulingOverrun in HcInterruptStatus has already been set. This is used by HCD to monitor any persistent scheduling problem. |
| 15:4 | / | / | 0x0 | Reserved |
| 3 | R/W | R/W | 0x0 | OwershipChangeRequest<br>This bit is set by an OS HCD to request a change of control of the HC. When set HC will set the OwnershipChange field in HcInterruptStatus. After the changeover, this bit is cleared and remains so until the next request from OS HCD. |
| 2 | R/W | R/W | 0x0 | BulklListFilled<br>This bit is used to indicate whether there are any TDs on the Bulk list. It is set by HCD whenever it adds a TD to an ED in the Bulk list.<br>When HC begins to process the head of the Bulk list, it checks BLF. As long as BulkListFilled is 0, HC will not start processing the Bulk list. If BulkListFilled is 1, HC will start processing the Bulk list and will set BF to 0. If HC finds a TD on the list, then HC will set BulkListFilled to 1 causing the Bulk list processing to continue. If no TD is found on the Bulk list, and if HCD does not set BulkListFilled, then BulkListFilled will still be 0 when HC completes processing the Bulk list and Bulk list processing will stop. |
| 1 | R/W | R/W | 0x0 | ControlListFilled<br>This bit is used to indicate whether there are any TDs on the Control list. It is set by HCD whenever it adds a TD to an ED in the Control list.<br>When HC begins to process the head of the Control list, it checks CLF. As long as ControlListFilled is 0, HC will not start processing the Control list. If |

| | | | | CF is 1, HC will start processing the Control list and will set ControlListFilled to 0. If HC finds a TD on the list, then HC will set ControlListFilled to 1 causing the Control list processing to continue. If no TD is found on the Control list, and if the HCD does not set ControlListFilled, then ControlListFilled will still be 0 when HC completes processing the Control list and Control list processing will stop. |
| | | | | HostControllerReset |
| | | | | This bit is by HCD to initiate a software reset of HC. Regardless of the functional state of HC, it moves to the USBSuspend state in which most of the operational registers are reset except those stated otherwise; e.g, the InteruptRouting field of HcControl, and no Host bus accesses are allowed. This bit is cleared by HC upon the completion of the reset operation. The reset operation must be completed within 10 ms. This bit, when set, should not cause a reset to the Root Hub and no subsequent |
| 0 | R/W | R/E | 0x0 | reset signaling should be asserted to its downstream ports. |

### 7.6.3.6.4. HcInterruptStatus Register(Default Value: 0x0)

| Offset: 0x40c | | | | Register Name: HcInterruptStatus |
|---|---|---|---|---|
| | Read/Write | | | |
| Bit | HCD | HC | Default/HEX | Description |
| 31:7 | / | / | 0x0 | Reserved |
| 6 | R/W | R/W | 0x0 0x1？ | RootHubStatusChange<br>This bit is set when the content of HcRhStatus or the content of any of HcRhPortStatus[NumberofDownstreamPort] has changed. |
| 5 | R/W | R/W | 0x0 | FrameNumberOverflow<br>This bit is set when the MSb of HcFmNumber (bit 15) changes value, from 0 to 1 or from 1 to 0, and after HccaFrameNumber has been updated. |
| 4 | R/W | R/W | 0x0 | UnrecoverableError<br>This bit is set when HC detects a system error not related to USB. HC should not proceed with any processing nor signaling before the system error has been corrected. HCD clears this bit after HC has been reset. |
| 3 | R/W | R/W | 0x0 | ResumeDetected<br>This bit is set when HC detects that a device on the USB is asserting resume signaling. It is the transition from no resume signaling to resume signaling causing this bit to be set. This bit is not set when HCD sets the USBRseume state. |
| 2 | R/W | R/W | 0x0 | StartofFrame<br>This bit is set by HC at each start of frame and after the update of HccaFrameNumber. HC also generates a SOF token at the same time. |
| 1 | R/W | R/W | 0x0 | WritebackDoneHead<br>This bit is set immediately after HC has written HcDoneHead to HccaDoneHead. Further updates of the HccaDoneHead will not occur until this bit has been cleared. HCD should only clear this bit after it has saved |

| | | | | the content of HccaDoneHead. |
|---|---|---|---|---|
| 0 | R/W | R/W | 0x0 | SchedulingOverrun<br>This bit is set when the USB schedule for the current Frame overruns and after the update of HccaFrameNumber. A scheduling overrun will also cause the SchedulingOverrunCount of HcCommandStatus to be Incremented. |

### 7.6.3.6.5. HcInterruptEnable Register(Default Value: 0x0)

| Offset: 0x410 | | | | Register Name: **HcInterruptEnable Register** |
|---|---|---|---|---|
| Bit | Read/Write | | Default | Description |
| | HCD | HC | | |
| 31 | R/W | R | 0x0 | MasterInterruptEnable<br>A '0' writtern to this field is ignored by HC. A '1' written to this field enables interrupt generation due to events specified in the other bits of this register. This is used by HCD as Master Interrupt Enable. |
| 30:7 | / | / | 0x0 | Reserved |
| 6 | R/W | R | 0x0 | RootHubStatusChange Interrupt Enable<br><table><tr><td>0</td><td>Ignore;</td></tr><tr><td>1</td><td>Enable interrupt generation due to Root Hub Status Change;</td></tr></table> |
| 5 | R/W | R | 0x0 | FrameNumberOverflow Interrupt Enable<br><table><tr><td>0</td><td>Ignore;</td></tr><tr><td>1</td><td>Enable interrupt generation due to Frame Number Over Flow;</td></tr></table> |
| 4 | R/W | R | 0x0 | UnrecoverableError Interrupt Enable<br><table><tr><td>0</td><td>Ignore;</td></tr><tr><td>1</td><td>Enable interrupt generation due to Unrecoverable Error;</td></tr></table> |
| 3 | R/W | R | 0x0 | ResumeDetected Interrupt Enable<br><table><tr><td>0</td><td>Ignore;</td></tr><tr><td>1</td><td>Enable interrupt generation due to Resume Detected;</td></tr></table> |
| 2 | R/W | R | 0x0 | StartofFrame Interrupt Enable<br><table><tr><td>0</td><td>Ignore;</td></tr><tr><td>1</td><td>Enable interrupt generation due to Start of Flame;</td></tr></table> |
| 1 | R/W | R | 0x0 | WritebackDoneHead Interrupt Enable<br><table><tr><td>0</td><td>Ignore;</td></tr><tr><td>1</td><td>Enable interrupt generation due to Write back Done Head;</td></tr></table> |
| 0 | R/W | R | 0x0 | SchedulingOverrun Interrupt Enable<br><table><tr><td>0</td><td>Ignore;</td></tr><tr><td>1</td><td>Enable interrupt generation due to Scheduling Overrun;</td></tr></table> |

### 7.6.3.6.6. HcInterruptDisable Register(Default Value: 0x0)

| Offset: 0x414 | | | | Register Name: HcInterruptDisable Register |
|---|---|---|---|---|
| | Read/Write | | | |
| Bit | HCD | HC | Default/HEX | Description |
| 31 | R/W | R | 0x0 | MasterInterruptEnable<br>A written '0' to this field is ignored by HC. A '1' written to this field disables interrupt generation due events specified in the other bits of this register. This field is set after a hardware or software reset. |
| 30:7 | / | / | 0x00 | Reserved |
| 6 | R/W | R | 0x0 | RootHubStatusChange Interrupt Disable<br>0 Ignore;<br>1 Disable interrupt generation due to Root Hub Status Change; |
| 5 | R/W | R | 0x0 | FrameNumberOverflow Interrupt Disable<br>0 Ignore;<br>1 Disable interrupt generation due to Frame Number Over Flow; |
| 4 | R/W | R | 0x0 | UnrecoverableError Interrupt Disable<br>0 Ignore;<br>1 Disable interrupt generation due to Unrecoverable Error; |
| 3 | R/W | R | 0x0 | ResumeDetected Interrupt Disable<br>0 Ignore;<br>1 Disable interrupt generation due to Resume Detected; |
| 2 | R/W | R | 0x0 | StartofFrame Interrupt Disable<br>0 Ignore;<br>1 Disable interrupt generation due to Start of Flame; |
| 1 | R/W | R | 0x0 | WritebackDoneHead Interrupt Disable<br>0 Ignore;<br>1 Disable interrupt generation due to Write back Done Head; |
| 0 | R/w | R | 0x0 | SchedulingOverrun Interrupt Disable<br>0 Ignore;<br>1 Disable interrupt generation due to Scheduling Overrun; |

### 7.6.3.6.7. HcHCCA Register(Default Value: 0x0)

| Offset: 0x418 | | | | Register Name: HcHCCA |
|---|---|---|---|---|
| | Read/Write | | | |
| Bit | HCD | HC | Default/HEX | Description |
| 31:8 | R/W | R | 0x0 | HCCA[31:8]<br>This is the base address of the Host Controller Communication Area. This area is used to hold the control structures and the Interrupt table that are accessed by both the Host Controller and the Host Controller Driver. |
| 7:0 | R | R | 0x0 | HCCA[7:0] |

The alignment restriction in HcHCCA register is evaluated by examining the number of zeros in the lower order bits. The minimum alignment is 256 bytes, therefore, bits 0 through 7 must always return 0 when read.

### 7.6.3.6.8. HcPeriodCurrentED Register(Default Value: 0x0)

| Offset: 0x41c | | | | Register Name: HcPeriodCurrentED(PCED) |
|---|---|---|---|---|
| | Read/Write | | | |
| Bit | HCD | HC | Default/HEX | Description |
| 31:4 | R | R/W | 0x0 | PCED[31:4]<br>This is used by HC to point to the head of one of the Periodec list which will be processed in the current Frame. The content of this register is updated by HC after a periodic ED has been processed. HCD may read the content in determining which ED is currently being processed at the time of reading. |
| 3:0 | R | R | 0x0 | PCED[3:0]<br>Because the general TD length is 16 bytes, the memory structure for the TD must be aligned to a 16-byte boundary. So the lower bits in the PCED, through bit 0 to bit 3 must be zero in this field. |

### 7.6.3.6.9. HcControlHeadED Register(Default Value: 0x0)

| Offset: 0x420 | | | | Register Name: **HcControlHeadED[CHED]** |
|---|---|---|---|---|
| | Read/Write | | | |
| Bit | HCD | HC | Default/HEX | Description |
| 31:4 | R/W | R | 0x0 | EHCD[31:4]<br>The HcControlHeadED register contains the physical address of the first Endpoint Descriptor of the Control list. HC traverse the Control list starting with the HcControlHeadED pointer. The content is loaded from HCCA during the initialization of HC. |
| 3:0 | R | R | 0x0 | EHCD[3:0]<br>Because the general TD length is 16 bytes, the memory structure for the TD must be aligned to a 16-byte boundary. So the lower bits in the PCED, through bit 0 to bit 3 must be zero in this field. |

### 7.6.3.6.10.    HcControlCurrentED Register(Default Value: 0x0)

| Offset: 0x424 | | | | Register Name: HcControlCurrentED[CCED] |
|---|---|---|---|---|
| | Read/Write | | | |
| Bit | HCD | HC | Default/Hex | Description |
| 31:4 | R/W | R/W | 0x0 | CCED[31:4] |

| | | | | The pointer is advanced to the next ED after serving the present one. HC will continue processing the list from where it left off in the last Frame. When it reaches the end of the Control list, HC checks the ControlListFilled of in HcCommandStatus. If set, it copies the content of HcControlHeadED to HcControlCurrentED and clears the bit. If not set, it does nothing.<br>HCD is allowed to modify this register only when the ControlListEnable of HcControl is cleared. When set, HCD only reads the instantaneous value of this register. Initially, this is set to zero to indicate the end of the Control list. |
| 3:0 | R | R | 0x0 | CCED[3:0]<br>Because the general TD length is 16 bytes, the memory structure for the TD must be aligned to a 16-byte boundary. So the lower bits in the PCED, through bit 0 to bit 3 must be zero in this field. |

### 7.6.3.6.11.    HcBulkHeadED Register(Default Value: 0x0)

| Offset: 0x428 | | | | Register Name: HcBulkHeadED[BHED] |
|---|---|---|---|---|
| | Read/Write | | | |
| Bit | HCD | HC | Default/HEX | Description |
| 31:4 | R/W | R | 0x0 | BHED[31:4]<br>The HcBulkHeadED register contains the physical address of the first Endpoint Descriptor of the Bulk list. HC traverses the Bulk list starting with the HcBulkHeadED pointer. The content is loaded from HCCA during the initialization of HC. |
| 3:0 | R | R | 0x0 | BHED[3:0]<br>Because the general TD length is 16 bytes, the memory structure for the TD must be aligned to a 16-byte boundary. So the lower bits in the PCED, through bit 0 to bit 3 must be zero in this field. |

### 7.6.3.6.12.    HcBulkCurrentED Register(Default Value: 0x0)

| Offset: 0x42c | | | | Register Name: HcBulkCurrentED [BCED] |
|---|---|---|---|---|
| | Read/Write | | | |
| Bit | HCD | HC | Default/HEX | Description |
| 31:4 | R/W | R/W | 0x0 | BulkCurrentED[31:4]<br>This is advanced to the next ED after the HC has served the present one. HC continues processing the list from where it left off in the last Frame. When it reaches the end of the Bulk list, HC checks the ControlListFilled of HcControl. If set, it copies the content of HcBulkHeadED to HcBulkCurrentED and clears the bit. If it is not set, it does nothing. HCD is only allowed to modify this register when the BulkListEnable of HcControl |

| | | | | is cleared. When set, the HCD only reads the instantaneous value of this register. This is initially set to zero to indicate the end of the Bulk list. |
| 3:0 | R | R | 0x0 | BulkCurrentED [3:0] Because the general TD length is 16 bytes, the memory structure for the TD must be aligned to a 16-byte boundary. So the lower bits in the PCED, through bit 0 to bit 3 must be zero in this field. |

### 7.6.3.6.13.    HcDoneHead Register(Default Value: 0x0)

| Offset: 0x430 | | | | Register Name: HcDoneHead |
|---|---|---|---|---|
| | Read/Write | | | |
| Bit | HCD | HC | Default/Hex | Description |
| 31:4 | R | R/W | 0x0 | HcDoneHead[31:4] When a TD is completed, HC writes the content of HcDoneHead to the NextTD field of the TD. HC then overwrites the content of HcDoneHead with the address of this TD. This is set to zero whenever HC writes the content of this register to HCCA. It also sets the WritebackDoneHead of HcInterruptStatus. |
| 3:0 | R | R | 0x0 | HcDoneHead[3:0] Because the general TD length is 16 bytes, the memory structure for the TD must be aligned to a 16-byte boundary. So the lower bits in the PCED, through bit 0 to bit 3 must be zero in this field. |

### 7.6.3.6.14.    HcFmInterval    Register(Default Value: 0x2EDF)

| Offset: 0x434 | | | | Register Name: HcFmInterval Register |
|---|---|---|---|---|
| | Read/Write | | | |
| Bit | HCD | HC | Default/Hex | Description |
| 31 | R/W | R | 0x0 | FrameIntervalToggler HCD toggles this bit whenever it loads a new value to FrameInterval. |
| 30:16 | R/W | R | 0x0 | FSLargestDataPacket This field specifies a value which is loaded into the Largest Data Packet Counter at the beginning of each frame. The counter value represents the largest amount of data in bits which can be sent or received by the HC in a single transaction at any given time without causing scheduling overrun. The field value is calculated by the HCD. |
| 15:14 | / | / | 0x0 | Reserved |
| 13:0 | R/W | R | 0x2edf | FrameInterval This specifies the interval between two consecutive SOFs in bit times. The nominal value is set to be 11,999. HCD should store the current value of this field before resetting HC. By setting the HostControllerReset field of HcCommandStatus as this will cause the HC to reset this field to its nominal value. HCD may choose to restore the stored value upon the |

completion of the Reset sequence.

### 7.6.3.6.15. HcFmRemaining Register(Default Value: 0x0)

| Offset: 0x438 | | | | Register Name: HcFmRemaining |
|---|---|---|---|---|
| | Read/Write | | | |
| Bit | HCD | HC | Default/Hex | Description |
| 31 | R | R/W | 0x0 | FrameRemaining Toggle<br>This bit is loaded from the FrameIntervalToggle field of HcFmInterval whenever FrameRemaining reaches 0. This bit is used by HCD for the synchronization between FrameInterval and FrameRemaining. |
| 30:14 | / | / | 0x0 | Reserved |
| 13:0 | R | RW | 0x0 | FramRemaining<br>This counter is decremented at each bit time. When it reaches zero, it is reset by loading the FrameInterval value specified in HcFmInterval at the next bit time boundary. When entering the USBOPERATIONAL state, HC re-loads the content with the FrameInterval of HcFmInterval and uses the updated value from the next SOF. |

### 7.6.3.6.16. HcFmNumber Register(Default Value: 0x0)

| Offset: 0x43c | | | | Register Name: HcFmNumber |
|---|---|---|---|---|
| | Read/Write | | | |
| Bit | HCD | HC | Default/Hex | Description |
| 31:16 | | | | Reserved |
| 15:0 | R | R/W | 0x0 | FrameNumber<br>This is incremented when HcFmRemaining is re-loaded. It will be rolled over to 0x0 after 0x0ffff. When entering the USBOPERATIONAL state, this will be incremented automatically. The content will be written to HCCA after HC has incremented the FrameNumber at each frame boundary and sent a SOF but before HC reads the first ED in that Frame. After writing to HCCA, HC will set the StartofFrame in HcInterruptStatus. |

### 7.6.3.6.17. HcPeriodicStart Register(Default Value: 0x0)

| Offset: 0x440 | | | | Register Name: HcPeriodicStatus |
|---|---|---|---|---|
| | Read/Write | | | |
| Bit | HCD | HC | Default/Hex | Description |
| 31:14 | | | | Reserved |

| 13:0 | R/W | R | 0x0 | PeriodicStart<br>After a hardware reset, this field is cleared. This is then set by HCD during the HC initialization. The value is calculated roughly as 10% off from HcFmInterval. A typical value will be 0x2A3F (0x3e67??). When HcFmRemaining reaches the value specified, processing of the periodic lists will have priority over Control/Bulk processing. HC will therefore start processing the Interrupt list after completing the current Control or Bulk transaction that is in progress. |

### 7.6.3.6.18.    HcLSThreshold Register(Default Value: 0x0628)

| Offset: 0x444 | | | | Register Name: HcLSThreshold |
|---|---|---|---|---|
| | Read/Write | | | |
| Bit | HCD | HC | Default/Hex | Description |
| 31:12 | | | | Reserved |
| 11:0 | R/W | R | 0x0628 | LSThreshold<br>This field contains a value which is compared to the FrameRemaining field prior to initiating a Low Speed transaction. The transaction is started only if FrameRemaining ³ this field. The value is calculated by HCD with the consideration of transmission and setup overhead. |

### 7.6.3.6.19.    HcRhDescriptorA    Register(Default Value: 0x0200_1201)

| Offset: 0x448 | | | | Register Name: HcRhDescriptorA |
|---|---|---|---|---|
| | Read/Write | | | |
| Bit | HCD | HC | Default/Hex | Description |
| 31:24 | R/W | R | 0x2 | PowerOnToPowerGoodTime[POTPGT]<br>This byte specifies the duration HCD has to wait before accessing a powered-on port of the Root Hub. It is implementation-specific. The unit of time is 2 ms. The duration is calculated as POTPGT * 2ms. |
| 23:13 | | | | Reserved |
| 12 | R/W | R | 1 | NoOverCurrentProtection<br>This bit describes how the overcurrent status for the Root Hub ports are reported. When this bit is cleared, the OverCurrentProtectionMode field specifies global or per-port reporting.<br><table><tr><td>0</td><td>Over-current status is reported collectively for all downstream ports.</td></tr><tr><td>1</td><td>No overcurrent protection supported.</td></tr></table> |
| 11 | R/W | R | 0 | OverCurrentProtectionMode<br>This bit describes how the overcurrent status for the Root Hub ports are reported. At reset, these fields should reflect the same mode as PowerSwitchingMode. This field is valid only if the |

| | | | | NoOverCurrentProtection field is cleared. |
|---|---|---|---|---|
| | | | | <table><tr><td>0</td><td>Over-current status is reported collectively for all downstream ports.</td></tr><tr><td>1</td><td>Over-current status is reported on per-port basis.</td></tr></table> |
| 10 | R | R | 0x0 | Device Type<br>This bit specifies that the Root Hub is not a compound device. The Root Hub is not permitted to be a compound device. This field should always read/write 0. |
| 9 | R/W | R | 1 | PowerSwitchingMode<br>This bit is used to specify how the power switching of the Root Hub ports is controlled. It is implementation-specific. This field is only valid if the NoPowerSwitching field is cleared.<br><table><tr><td>0</td><td>All ports are powered at the same time.</td></tr><tr><td>1</td><td>Each port is powered individually. This mode allows port power to be controlled by either the global switch or per-port switching. If the PortPowerControlMask bit is set, the port responds only to port power commands (Set/ClearPortPower). If the port mask is cleared, then the port is controlled only by the global power switch (Set/ClearGlobalPower).</td></tr></table> |
| 8 | R/W | R | 0 | NoPowerSwithcing<br>These bits are used to specify whether power switching is supported or ports are always powered. It is implementation-specific. When this bit is cleared, the PowerSwitchingMode specifies global or per-port switching.<br><table><tr><td>0</td><td>Ports are power switched.</td></tr><tr><td>1</td><td>Ports are always powered on when the HC is powered on.</td></tr></table> |
| 7:0 | R | R | 0x01 | NumberDownstreamPorts<br>These bits specify the number of downstream ports supported by the Root Hub. It is implementation-specific. The minimum number of ports is 1. The maximum number of ports supported. |

### 7.6.3.6.20.　HcRhDescriptorB Register(Default Value: 0x0)

| Offset: 0x44c | | | | Register Name: HcRhDescriptorB Register |
|---|---|---|---|---|
| | Read/Write | | | |
| Bit | HCD | HC | Default/Hex | Description |
| 31:16 | R/W | R | 0x0 | PortPowerControlMask<br>Each bit indicates if a port is affected by a global power control command when PowerSwitchingMode is set. When set, the port's power state is only affected by per-port power control (Set/ClearPortPower). When cleared, the port is controlled by the global power switch (Set/ClearGlobalPower). If the device is configured to global switching mode (PowerSwitchingMode = 0 ), this field is not valid. |

| | | | | Bit0 | Reserved |
|---|---|---|---|---|---|
| | | | | Bit1 | Ganged-power mask on Port #1. |
| | | | | Bit2 | Ganged-power mask on Port #2. |
| | | | | … | |
| | | | | Bit15 | Ganged-power mask on Port #15. |
| | | | | DeviceRemovable | |
| | | | | Each bit is dedicated to a port of the Root Hub. When cleared, the attached device is removable. When set, the attached device is not removable. | |
| | | | | Bit0 | Reserved |
| | | | | Bit1 | Device attached to Port #1. |
| | | | | Bit2 | Device attached to Port #2. |
| | | | | … | |
| 15:0 | R/W | R | 0x0 | Bit15 | Device attached to Port #15. |

### 7.6.3.6.21. HcRhStatus Register(Default Value: 0x0)

| Offset: 0x450 | | | | Register Name: HcRhStatus Register |
|---|---|---|---|---|
| | Read/Write | | | |
| Bit | HCD | HC | Default/Hex | Description |
| 31 | W | R | 0 | (write)ClearRemoteWakeupEnable<br>Write a '1' clears DeviceRemoteWakeupEnable. Write a '0' has no effect. |
| 30:18 | / | / | 0x0 | Reserved |
| 17 | R/W | R | 0 | OverCurrentIndicatorChang<br>This bit is set by hardware when a change has occurred to the OverCurrentIndicator field of this register. The HCD clears this bit by writing a '1'.Writing a '0' has no effect. |
| 16 | R/W | R | 0x0 | (read)LocalPowerStartusChange<br>The Root Hub does not support the local power status features, thus, this bit is always read as '0'.<br>(write)SetGlobalPower<br>In global power mode (PowerSwitchingMode=0), This bit is written to '1' to turn on power to all ports (clear PortPowerStatus). In per-port power mode, it sets PortPowerStatus only on ports whose PortPowerControlMask bit is not set. Writing a '0' has no effect. |
| 15 | R/W | R | 0x0 | (read)DeviceRemoteWakeupEnable<br>This bit enables a ConnectStatusChange bit as a resume event, causing a USBSUSPEND to USBRESUME state transition and setting the ResumeDetected interrupt.<br><br>0 — ConnectStatusChange is not a remote wakeup event.<br>1 — ConnectStatusChange is a remote wakeup event. |

| | | | | (write)SetRemoteWakeupEnable |
|---|---|---|---|---|
| | | | | Writing a '1' sets DeviceRemoveWakeupEnable. Writing a '0' has no effect. |
| 14:2 | | | | Reserved |
| 1 | R | R/W | 0x0 | OverCurrentIndicator<br>This bit reports overcurrent conditions when the global reporting is implemented. When set, an overcurrent condition exists. When cleared, all power operations are normal.<br>If per-port overcurrent protection is implemented this bit is always '0' |
| 0 | R/W | R | 0x0 | (Read)LocalPowerStatus<br>When read, this bit returns the LocalPowerStatus of the Root Hub. The Root Hub does not support the local power status feature; thus, this bit is always read as '0'.<br>(Write)ClearGlobalPower<br>When write, this bit is operated as the ClearGlobalPower. In global power mode (PowerSwitchingMode=0), This bit is written to '1' to turn off power to all ports (clear PortPowerStatus). In per-port power mode, it clears PortPowerStatus only on ports whose PortPowerControlMask bit is not set. Writing a '0' has no effect. |

### 7.6.3.6.22.    HcRhPortStatus Register(Default Value: 0x100)

| Offset: 0x454 | | | | Register Name: HcRhPortStatus |
|---|---|---|---|---|
| | Read/Write | | | |
| Bit | HCD | HC | Default/Hex | Description |
| 31:21 | / | / | 0x0 | Reserved |
| 20 | R/W | R/W | 0x0 | PortResetStatusChange<br>This bit is set at the end of the 10-ms port reset signal. The HCD writes a '1' to clear this bit. Writing a '0' has no effect.<br><table><tr><td>0</td><td>port reset is not complete</td></tr><tr><td>1</td><td>port reset is complete</td></tr></table> |
| 19 | R/W | R/W | 0x0 | PortOverCurrentIndicatorChange<br>This bit is valid only if overcurrent conditions are reported on a per-port basis. This bit is set when Root Hub changes the PortOverCurrentIndicator bit. The HCD writes a '1' to clear this bit. Writing a '0' has no effect.<br><table><tr><td>0</td><td>no change in PortOverCurrentIndicator</td></tr><tr><td>1</td><td>PortOverCurrentIndicator has changed</td></tr></table> |
| 18 | R/W | R/W | 0x0 | PortSuspendStatusChange<br>This bit is set when the full resume sequence has been completed. This sequence includes the 20-s resume pulse, LS EOP, and 3-ms resynchronization delay. The HCD writes a '1' to clear this bit. Writing a '0' has no effect. This bit is also cleared when ResetStatusChange is set. |

| | | | | | |
|---|---|---|---|---|---|
| | | | | 0 | resume is not completed |
| | | | | 1 | resume completed |
| 17 | R/W | R/W | 0x0 | PortEnableStatusChange<br>This bit is set when hardware events cause the PortEnableStatus bit to be cleared. Changes from HCD writes do not set this bit. The HCD writes a '1' to clear this bit. Writing a '0' has no effect.<br><table><tr><td>0</td><td>no change in PortEnableStatus</td></tr><tr><td>1</td><td>change in PortEnableStatus</td></tr></table> | |
| 16 | R/W | R/W | 0x0 | ConnectStatusChange<br>This bit is set whenever a connect or disconnect event occurs. The HCD writes a '1' to clear this bit. Writing a '0' has no effect. If CurrentConnectStatus is cleared when a SetPortReset,SetPortEnable, or SetPortSuspend write occurs, this bit is set to force the driver to re-evaluate the connection status since these writes should not occur if the port is disconnected.<br><table><tr><td>0</td><td>no change in PortEnableStatus</td></tr><tr><td>1</td><td>change in PortEnableStatus</td></tr></table><br>Note: If the DeviceRemovable[NDP] bit is set, this bit is set only after a Root Hub reset to inform the system that the device is attached. | |
| 15:10 | / | / | 0x0 | Reserved | |
| 9 | R/W | R/W | - | (read)LowSpeedDeviceAttached<br>This bit indicates the speed of the device attached to this port. When set, a Low Speed device is attached to this port. When clear, a Full Speed device is attached to this port. This field is valid only when the CurrentConnectStatus is set.<br><table><tr><td>0</td><td>full speed device attached</td></tr><tr><td>1</td><td>low speed device attached</td></tr></table><br>(write)ClearPortPower<br>The HCD clears the PortPowerStatus bit by writing a '1' to this bit. Writing a '0' has no effect. | |
| 8 | R/W | R/W | 0x1 | (read)PortPowerStatus<br>This bit reflects the port's power status, regardless of the type of power switching implemented. This bit is cleared if an overcurrent condition is detected. HCD sets this bit by writing SetPortPower or SetGlobalPower. HCD clears this bit by writing ClearPortPower or ClearGlobalPower. Which power control switches are enabled is determined by PowerSwitchingMode and PortPortControlMask[NumberDownstreamPort]. In global switching mode(PowerSwitchingMode=0), only Set/ClearGlobalPower controls this bit. In per-port power switching (PowerSwitchingMode=1), if the PortPowerControlMask[NDP] bit for the port is set, only Set/ClearPortPower commands are enabled. If the mask is not set, only | |

| | | | | |
|---|---|---|---|---|
| | | | | Set/ClearGlobalPower commands are enabled. When port power is disabled, CurrentConnectStatus, PortEnableStatus, PortSuspendStatus, and PortResetStatus should be reset. |
| | | | | <table><tr><td>0</td><td>port power is off</td></tr><tr><td>1</td><td>port power is on</td></tr></table> (write)SetPortPower<br><br>The HCD writes a '1' to set the PortPowerStatus bit. Writing a '0' has no effect.<br><br>Note: This bit is always reads '1b' if power switching is not supported. |
| 7:5 | / | / | 0x0 | Reserved |
| 4 | R/W | R/W | 0x0 | (read)PortResetStatus<br>When this bit is set by a write to SetPortReset, port reset signaling is asserted. When reset is completed, this bit is cleared when PortResetStatusChange is set. This bit cannot be set if CurrentConnectStatus is cleared.<br><table><tr><td>0</td><td>port reset signal is not active</td></tr><tr><td>1</td><td>port reset signal is active</td></tr></table> (write)SetPortReset<br>The HCD sets the port reset signaling by writing a '1' to this bit. Writing a '0' has no effect. If CurrentConnectStatus is cleared, this write does not set PortResetStatus, but instead sets ConnectStatusChange. This informs the driver that it attempted to reset a disconnected port. |
| 3 | R/W | R/W | 0x0 | (read)PortOverCurrentIndicator<br>This bit is only valid when the Root Hub is configured in such a way that overcurrent conditions are reported on a per-port basis. If per-port overcurrent reporting is not supported, this bit is set to 0. If cleared, all power operations are normal for this port. If set, an overcurrent condition exists on this port. This bit always reflects the overcurrent input signal.<br><table><tr><td>0</td><td>no overcurrent condition.</td></tr><tr><td>1</td><td>overcurrent condition detected.</td></tr></table> (write)ClearSuspendStatus<br>The HCD writes a '1' to initiate a resume. Writing a '0' has no effect. A resume is initiated only if PortSuspendStatus is set. |
| 2 | R/W | R/W | 0x0 | (read)PortSuspendStatus<br>This bit indicates the port is suspended or in the resume sequence. It is set by a SetSuspendState write and cleared when PortSuspendStatusChange is set at the end of the resume interval. This bit cannot be set if CurrentConnectStatus is cleared. This bit is also cleared when PortResetStatusChange is set at the end of the port reset or when the HC is placed in the USBRESUME state. If an upstream resume is in progress, it should propagate to the HC. |

| | | | | 0 | port is not suspended |
|---|---|---|---|---|---|
| | | | | 1 | port is suspended |

(write)SetPortSuspend
The HCD sets the PortSuspendStatus bit by writing a '1' to this bit. Writing a '0' has no effect. If CurrentConnectStatus is cleared, this write does not set PortSuspendStatus; instead it sets ConnectStatusChange. This informs the driver that it attempted to suspend a disconnected port.

| 1 | R/W | R/W | 0x0 | (read)PortEnableStatus<br>This bit indicates whether the port is enabled or disabled. The Root Hub may clear this bit when an overcurrent condition, disconnect event, switched-off power, or operational bus error such as babble is detected. This change also causes PortEnabledStatusChange to be set. HCD sets this bit by writing SetPortEnable and clears it by writing ClearPortEnable. This bit cannot be set when CurrentConnectStatus is cleared. This bit is also set, if not already, at the completion of a port reset when ResetStatusChange is set or port suspend when SuspendStatusChange is set.<br><br>0 port is disabled<br>1 port is enabled<br><br>(write)SetPortEnable<br>The HCD sets PortEnableStatus by writing a '1'. Writing a '0' has no effect. If CurrentConnectStatus is cleared, this write does not set PortEnableStatus, but instead sets ConnectStatusChange. This informs the driver that it attempted to enable a disconnected Port. |
|---|---|---|---|---|
| 0 | R/W | R/W | 0x0 | (read)CurrentConnectStatus<br>This bit reflects the current state of the downstream port.<br><br>0 No device connected<br>1 Device connected<br><br>(write)ClearPortEnable<br>The HCD writes a '1' to clear the PortEnableStatus bit. Writing '0' has no effect. The CurrentConnectStatus is not affected by any write.<br>Note: This bit is always read '1' when the attached device is nonremovalble(DviceRemoveable[NumberDownstreamPort]). |

### 7.6.3.7. HCI Interface Control and Status Register Description

### 7.6.3.7.1. HCI Interface Control Register(Default Value: 0x0)

| Offset: 0x800 | Register Name: **HCI_ICR** |
|---|---|

| Bit | R/W | Default/Hex | Description |
|---|---|---|---|
| 31:21 | / | / | Reserved. |
| 20 | R/W | 0 | EHCI HS force<br>Set 1 to this field force the ehci enter the high speed mode during bus reset.<br>This field only valid when the bit 1 is set. |
| 19:18 | / | / | / |
| 17 | R/W | 0 | HSIC Connect detect<br>1 in this field enable the hsic phy to detect device connect pulse on the bus.<br>This field only valid when the bit 1 is set. |
| 16 | R/W | 0 | HSIC Connect Interrupt Enable<br>Enable the HSIC connect interrupt.<br>This field only valid when the bit 1 is set. |
| 15:13 | / | / | / |
| 12 | / | / | / |
| 11 | R/W | 0 | AHB Master interface INCR16 enable<br>1: Use INCR16 when appropriate<br>0: do not use INCR16,use other enabled INCRX or unspecified length burst INCR |
| 10 | R/W | 0 | AHB Master interface INCR8 enable<br>1: Use INCR8 when appropriate<br>0: do not use INCR8,use other enabled INCRX or unspecified length burst INCR |
| 9 | R/W | 0 | AHB Master interface burst type INCR4 enable<br>1: Use INCR4 when appropriate<br>0: do not use INCR4,use other enabled INCRX or unspecified length burst INCR |
| 8 | R/W | 0 | AHB Master interface INCRX align enable<br>1: start INCRx burst only on burst x-align address<br>0: Start burst on any double word boundary<br>Note: This bit must enable if any bit of 11:9 is enabled |
| 7:2 | / | / | Reserved |
| 1 | R/W | 0 | HSIC<br>0:/<br>1:HSIC<br>This meaning is only valid when the controller is HCI1. |
| 0 | R/W | 0 | ULPI bypass enable。<br>1: Enable UTMI interface, disable ULPI interface(SP used utmi interface)<br>0: Enable ULPI interface, disable UTMI interface |

### 7.6.3.7.2. HSIC status Register(Default Value: 0x0)

| Offset: 0x804 | Register Name: HSIC_STATUS |
|---|---|

| Bit | R/W | Default/Hex | Description |
|-----|-----|-------------|-------------|
| 31:17 | / | / | / |
| 16 | R/W | 0 | HSIC Connect Status<br>1 in this field indicates a device connect pulse being detected on the bus. This field only valid when the EHCI HS force bit and the HSIC Phy Select bit is set. When the HSIC Connect Interrupt Enable is set, 1 in this bit will generate an interrupt to the system.<br>This register is valid on HCI1. |
| 15:0 | / | / | / |

### 7.6.3.8. HCI Clock Control Register Description

#### 7.6.3.8.1. HCI SIE Control Register(Default Value: 0x0)

| Offset: 0x00A08000 | | | Register Name: HCI_SCR |
|-----|-----|-----|-----|
| Bit | R/W | Default/Hex | Description |
| 31:20 | / | / | / |
| 19 | R/W | 0 | USB HCI2 Module Reset<br>0: assert<br>1: de-assert |
| 18 | R/W | 0 | USB HCI1 Module Reset<br>0: assert<br>1: de-assert |
| 17 | R/W | 0 | USB HCI0 Module Reset<br>0: assert<br>1: de-assert |
| 16:7 | / | / | / |
| 6 | R/W | 0 | USB OHCI2 Special Clock(12M and 48M) Gating<br>0: mask<br>1: pass |
| 5 | R/W | 0 | USB HCI2 AHB Gating<br>0: mask<br>1: pass |
| 4 | / | / | / |
| 3 | R/W | 0 | USB HCI1 AHB Gating<br>0: mask<br>1: pass |
| 2 | R/W | 0 | USB OHCI0 Special Clock(12M and 48M) Gating<br>0: mask<br>1: pass |
| 1 | R/W | 0 | USB HCI0 AHB Gating |

| | | | 0: mask |
|---|---|---|---|
| | | | 1: pass |
| 0 | / | / | / |

### 7.6.3.8.2. HCI PHY Control Register(Default Value: 0x0)

| Offset: 0x00A08004 | | | Register Name: HCI_PCR |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:22 | / | / | / |
| 21 | R/W | 0 | HCI2_UTMIPHY_RST<br>0: Assert<br>1: De-assert |
| 20:19 | / | / | / |
| 18 | R/W | 0 | HCI1_HSIC_RST<br>0: Assert<br>1: De-assert |
| 17 | R/W | 0 | HCI0_PHY_RST<br>0: Assert<br>1: De-assert |
| 16:11 | / | / | / |
| 10 | R/W | 0 | 12M_GATING_HCI1_HSIC<br>0: Clock is OFF<br>1: Clock is ON |
| 9:6 | / | / | / |
| 5 | R/W | 0 | SCLK_GATING_HCI2_UTMIPHY<br>0: Clock is OFF<br>1: Clock is ON |
| 4 | R/W | 0 | 480M_GATING_HCI2_HSIC<br>0: Clock is OFF<br>1: Clock is ON |
| 3 | / | / | / |
| 2 | R/W | 0 | 480M_GATING_HCI1_HSIC<br>0: Clock is OFF<br>1: Clock is ON |
| 1 | R/W | 0 | SCLK_GATING_HCI0_PHY<br>0: Clock is OFF<br>1: Clock is ON |
| 0 | / | / | / |

### 7.6.3.9. USB Host Special Requirement

| Name | Description |
| --- | --- |
| HCLK | System clock (provided by AHB bus clock). This clock needs to be >30MHz. |
| CLK60M | Clock from PHY for HS SIE, is constant to be 60MHz. |
| CLK48M | Clock from PLL for FS/LS SIE, is constant to be 48MHz. |

# 7.7.  TDM

## 7.7.1.  Overview

The Digital Audio Interface Controller has been designed to transfer streaming audio-data between the system memory and the codec chip. The controller supports standard I2S format, Left-justified Mode format, Right-justified Mode format ,PCM Mode format and TDM Mode format.

The digital audio interface controller includes the following features:
- Supports industry-standard AMBA Peripheral Bus (APB) and it is fully compliant with the AMBA Specification, Revision 2.0
- Support different sample period width in each interface when using LRCK and LRCKR at the same time
- Support full-duplex synchronous work mode
- Support Master / Slave mode
- Support adjustable interface voltage
- Support clock up to 100MHz
- Support adjustable audio sample rate from 8-bit to 32-bit.
- Support up to 8 slots which has adjustable width from 8-bit to 32-bit.
- Support sample rate from 8KHz to 192KHz
- Support up to 4 data output pin
- Support 8-bits or 16-bits linear sample,8-bits u-law and 8-bits A-law companded sample
- One 128 x 24-bit width FIFO for data transmit, one 64 x 24-bit width FIFO for data receive
- Support programmable PCM frame width: 1 BCLK width (short frame) and 2 BCLKs width (long frame)
- Programmable FIFO thresholds
- Interrupt and DMA Support
- Support loopback mode for test

## 7.7.2.  Signal Description

### 7.7.2.1.  Digital Audio Interface Pin List

| Signal Name | Direction(M) | Description | Pin |
|---|---|---|---|
| MCLK | O | Digital Audio 0 MCLK Output | PB8 |
| BCLK | I/O | Digital Audio 0 Sample Rate Clock/Sync | PB5 |
| LRCK | I/O | Digital Audio 0 Serial Clock | PB4 |
| SDI | I | Digital Audio 0 Serial Data input | PB7 |
| SDO | O | Digital Audio 0 Serial Data Output | PB6 |

### 7.7.2.2.  Digital Audio Interface Clock Source and Frequency

| | Description |
|---|---|
| Audio_PLL | 24.576Mhz or 22.5792Mhz generated by AUDIO-PLL to produce 48KHz or 44.1KHz serial frequency |

## 7.7.3.  Functionalities Description

### 7.7.3.1.  Typical Applications

The I2S and PCM which provides a serial bus interface for stereo and multichannel audio data. This interface is most commonly used by consumer audio market, including compact disc, digital audio tape, digital sound processors, and digital TV-sound.

### 7.7.3.2.  Functional Block Diagram

The Digital Audio Interface block diagram is shown below:
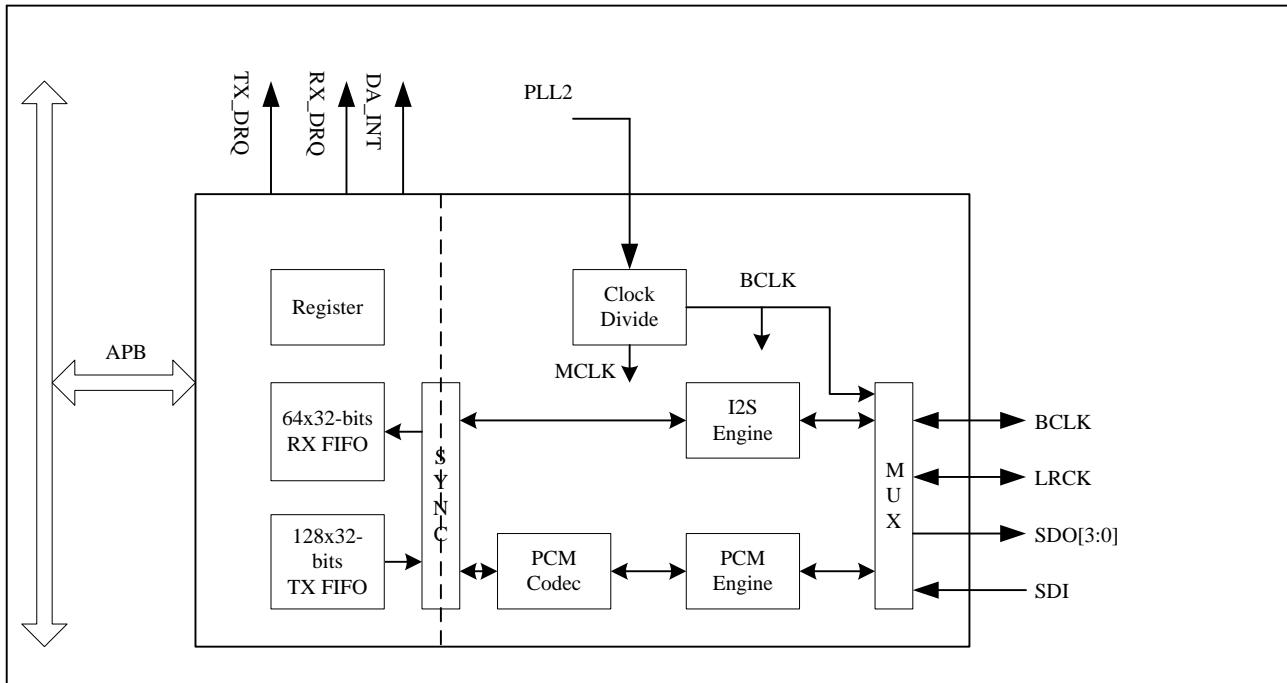
Figure 7-9. Digital Audio Interface System Block Diagram

## 7.7.4. Timing Diagram
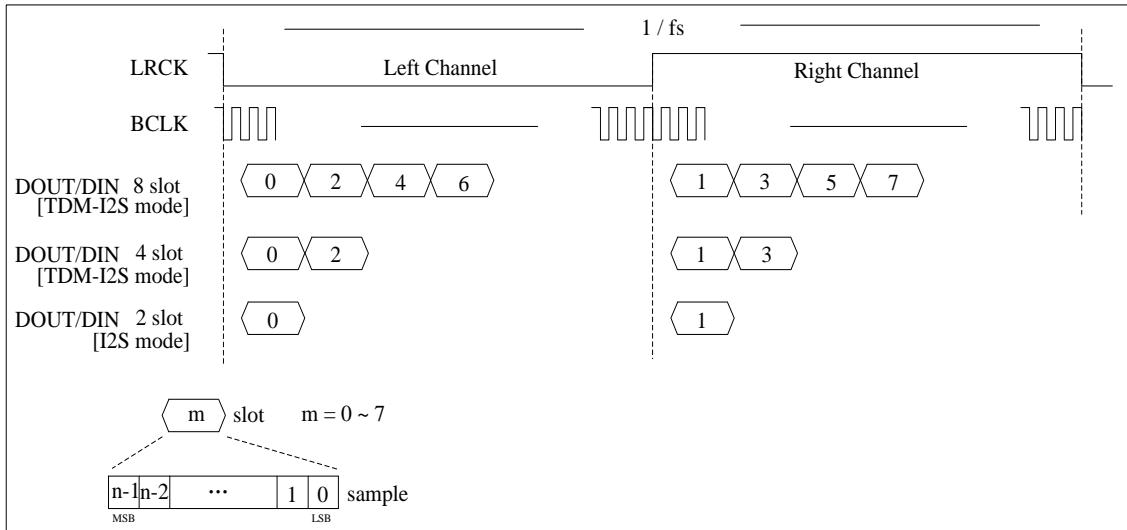


Figure 7-10. Timing diagram for I2S and Multi-Slot mode
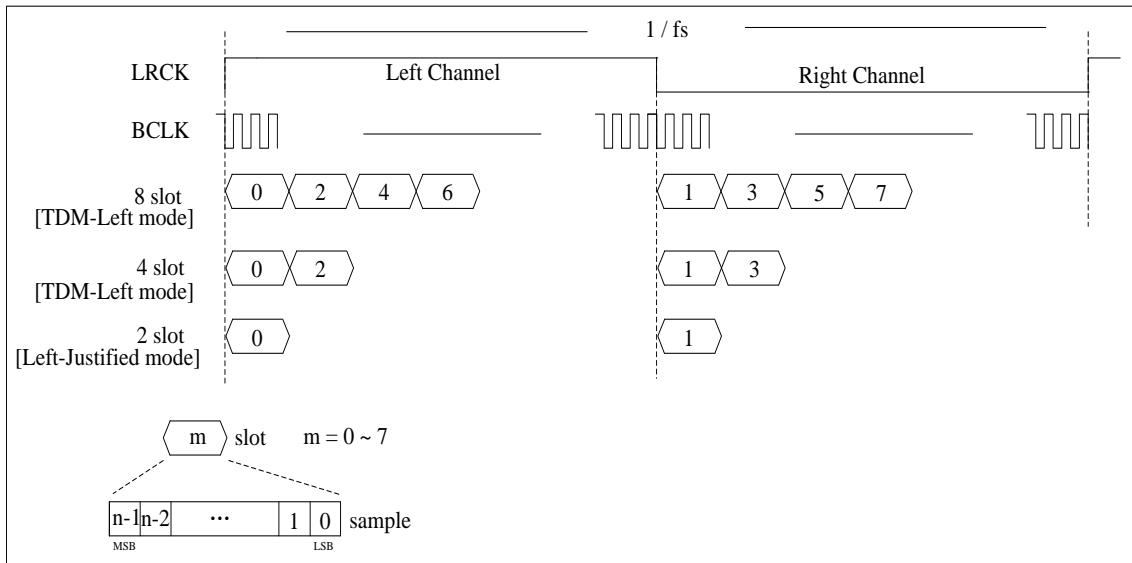
Figure 7-11. Timing diagram for Left-justified and Multi-Slot mode
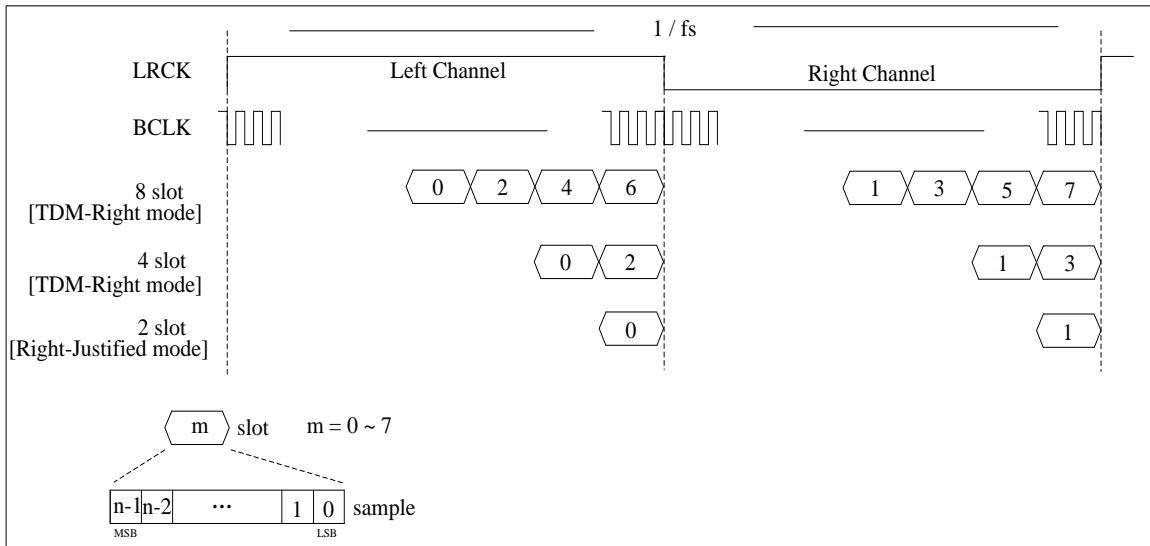


Figure 7-12. Timing diagram for Right-justified and Multi-Slot mode
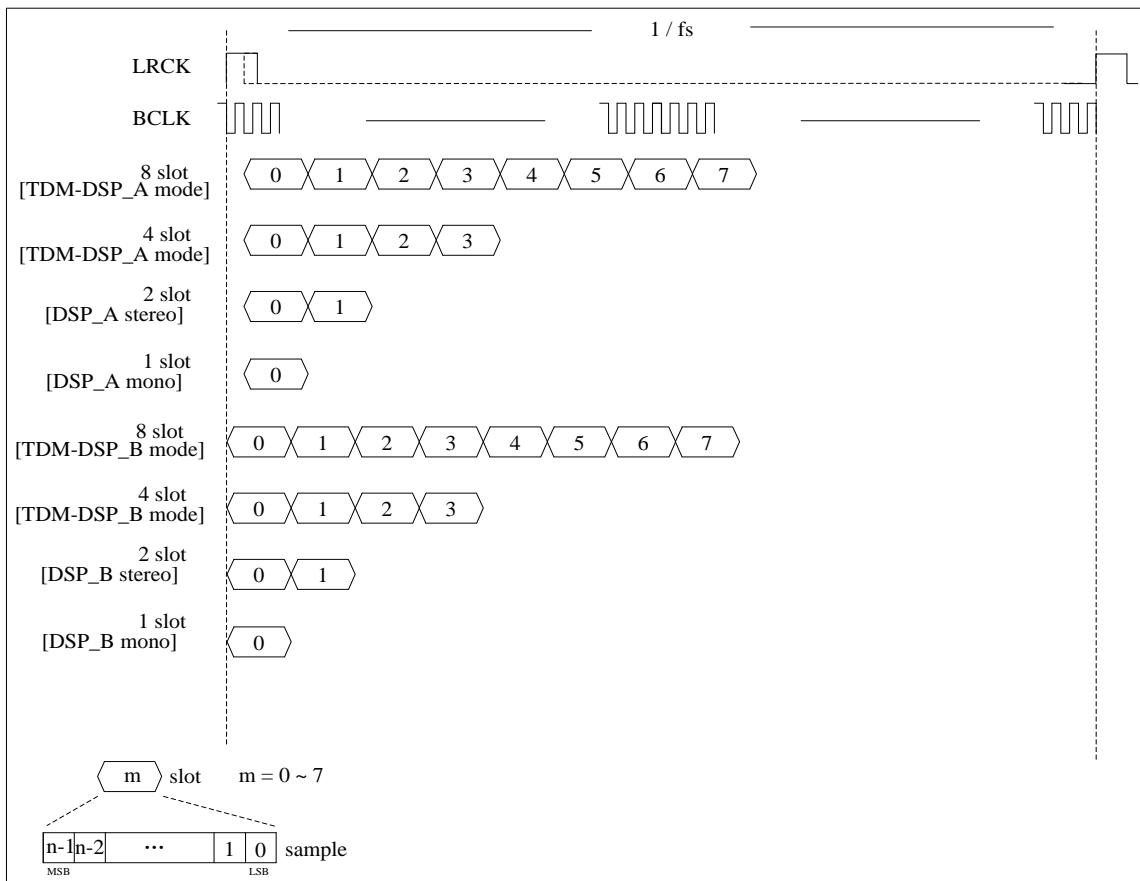
Figure 7-13. Timing diagram for PCM and Multi-Slot PCM mode

## 7.7.5.    Operation Modes

The software operation of the PCM/I2S is divided into five steps: system setup, PCM/I2S initialization, the channel setup, DMA setup and Enable/Disable module. These five steps are described in detail in the following sections.

### 7.7.5.1.    System setup and PCM/I2S initialization

The first step in the system setup is properly programming the GPIO. The PCM/I2S port is a multiplex pin and its function can be found in the **Port Controller**. The clock source for the PCM/I2S should be followed. At first you must reset the audio PLL through the **PLL_ENABLE** bit of **PLL_AUDIO_CTRL_REG** in the **CCU**. The second step, you must setup the frequency of the audio pll in the **PLL_AUDIO_CTRL_REG**. After that, you must open the PCM/I2S gating through the **DAUDIO0_CLK_REG**/**DAUDIO1_CLK_REG** when you checkout that the LOCK bit of **PLL_AUDIO_CTRL_REG** become 1. At last, you must reset the PCM/I2S the **BUS_SOFT_RST_REG3**'s bit[13:12] and open the PCM/I2S bus gating in the **BUS_CLK_GATING_REG2**'s bit[13:12].

After the system setup, the register of PCM/I2S can be setup. At first, you should initialization the PCM/I2S. You should closed the **Globe Enable** bit(**DA_CTL** [0]) , **Transmitter Block Enable** (**DA_CTL**[2]) and **Receiver Block Enable** (**DA_CTL**[1]) by writing 0 to it. After that, you must clear the TX/RX FIFO by writing 0 to register **DA_FCTL** [25:24]. At last, you can

clear the TX/RX FIFO counter by writing 0 to **DA_TXCNT**/**DA_RXCNT**.

### 7.7.5.2. The channel setup and DMA setup

At first, you can setup the PCM/I2S of mater and slave by set **DA_CTL**[18:16]. The configuration can be referred to the the protocol of PCM/I2S. Then, you can set the translation mode, the sample precision, the wide of slot, the frame mode and the trigger level.

The PCM/I2S support three methods to transfer the data. The most common way is DMA, the set of DMA can be found in the **DMA** part. In this module, you just to enable the DRQ. Because the PCM/I2S is in the CPUS domain, you can only use the rDMA which is also in the CPUS domain.

### 7.7.5.3. Enable and disable the PCM/I2S

To enable the function, you can enable TX/RX by writing the **DA_CTL**[2:1]. After that, you must enable PCM/I2S by writing the **Globe Enable** bit to 1 in the **DA_CTL**. Writting the **Globe Enable** to 0 can disable the PCM/I2S.

## 7.7.6. Digital Audio Interface Register List

| Module Name | Base Address |
|---|---|
| TDM | 0x01C2 3000 |

| Register Name | Offset | Description |
|---|---|---|
| DA_CTL | 0x00 | Digital Audio Control Register |
| DA_FMT0 | 0x04 | Digital Audio Format Register 0 |
| DA_FMT1 | 0x08 | Digital Audio Format Register 1 |
| DA_ISTA | 0x0C | Digital Audio Interrupt Status Register |
| DA_RXFIFO | 0x10 | Digital Audio RX FIFO Register |
| DA_FCTL | 0x14 | Digital Audio FIFO Control Register |
| DA_FSTA | 0x18 | Digital Audio FIFO Status Register |
| DA_INT | 0x1C | Digital Audio DMA & Interrupt Control Register |
| DA_TXFIFO | 0x20 | Digital Audio TX FIFO Register |
| DA_CLKD | 0x24 | Digital Audio Clock Divide Register |
| DA_TXCNT | 0x28 | Digital Audio TX Sample Counter Register |
| DA_RXCNT | 0x2C | Digital Audio RX Sample Counter Register |
| DA_CHCFG | 0x30 | Digital Audio Channel Configuration register |

| DA_CHCFG | 0x34 | Digital Audio Channel Configuration register |
|---|---|---|
| DA_TX1CHSEL | 0x38 | Digital Audio TX1 Channel Select Register |
| DA_TX2CHSEL | 0x3C | Digital Audio TX2 Channel Select Register |
| DA_TX3CHSEL | 0x40 | Digital Audio TX3 Channel Select Register |
| DA_TX0CHMAP | 0x44 | Digital Audio TX0 Channel Mapping Register |
| DA_TX0CHMAP | 0x48 | Digital Audio TX1 Channel Mapping Register |
| DA_TX0CHMAP | 0x4C | Digital Audio TX2 Channel Mapping Register |
| DA_TX0CHMAP | 0x50 | Digital Audio TX3 Channel Mapping Register |
| DA_RXCHSEL | 0x54 | Digital Audio RX Channel Select register |
| DA_RXCHMAP | 0x58 | Digital Audio RX Channel Mapping Register |

## 7.7.7. Digital Audio Interface Register Description

### 7.7.7.1. Digital Audio Control Register(Default Value: 0x0006_0000)

| Offset: 0x00 | | | Register Name: **DA_CTL** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:19 | / | / | / |
| 18 | R/W | 1 | BCLK_OUT<br>0: input<br>1: output |
| 17 | R/W | 1 | LRCK_OUT<br>0: input<br>1: output |
| 16 | R/W | 0 | LRCKR_OUT(No Pin)<br>0: input<br>1: output |
| 15:12 | / | / | / |
| 11 | R/W | 0 | SDO3_EN(No Pin)<br>0: Disable, Hi-Z state<br>1: Enable |
| 10 | R/W | 0 | SDO2_EN(No Pin)<br>0: Disable, Hi-Z state<br>1: Enable |
| 9 | R/W | 0 | SDO1_EN(No Pin)<br>0: Disable, Hi-Z state<br>1: Enable |
| 8 | R/W | 0 | SDO0_EN(No Pin)<br>0: Disable, Hi-Z state |

| | | | 1: Enable |
|---|---|---|---|
| 7 | / | / | / |
| 6 | R/W | 0 | OUT Mute<br>0: normal transfer<br>1: force DOUT to output 0 |
| 5:4 | R/W | 0 | MODE_SEL<br>Mode Selection<br>0: PCM mode (offset 0: DSP_B;    offset 1: DSP_A)<br>1: Left mode (offset 0: LJ mode;    offset 1: I2S mode)<br>2: Right-Justified mode<br>3: Reserved |
| 3 | R/W | 0 | LOOP<br>Loop back test<br>0: Normal mode<br>1: Loop back test<br>When set '1', connecting the SDO0 with the SDI |
| 2 | R/W | 0 | TXEN<br>Transmitter Block Enable<br>0: Disable<br>1: Enable |
| 1 | R/W | 0 | RXEN<br>Receiver Block Enable<br>0: Disable<br>1: Enable |
| 0 | R/W | 0 | GEN<br>Globe Enable<br>A disable on this bit overrides any other block or channel enables.<br>0: Disable<br>1: Enable |

### 7.7.7.2.  Digital Audio Format Register0 (Default Value: 0x0000_0033)

| Offset: 0x04 | | | Register Name: **DA_FMT0** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0 | SDI_SYNC_SEL<br>0: SDI use LRCK<br>1: SDI use LRCKR |
| 30 | R/W | 0 | LRCK_WIDTH<br>(only apply in PCM mode ) LRCK width<br>0: LRCK = 1 BCLK width (short frame)<br>1: LRCK = 2 BCLK width (long frame) |
| 29:20 | R/W | 0 | LRCKR_PERIOD<br>It is used to program the number of BCLKs per channel of sample frame.<br>This value is interpreted as follow: |

| | | | |
|---|---|---|---|
| | | | PCM mode: Number of BCLKs within (Left + Right) channel width |
| | | | I2S / Left-Justified / Right-Justified mode: Number of BCLKs within each individual channel width (Left or Right) |
| | | | N+1 |
| | | | For example: |
| | | | n = 7:    8 BCLK width |
| | | | … |
| | | | n = 1023:    1024 BCLKs width |
| 19 | R/W | 0 | LRCK_POLARITY/LRCKR_POLARITY<br>When apply in I2S / Left-Justified / Right-Justified mode:<br>0: Left channel when LRCK is low<br>1: Left channel when LRCK is high<br>When apply in PCM mode:<br>0: PCM LRCK/LRCKR asserted at the negative edge<br>1: PCM LRCK/LRCKR asserted at the positive edge |
| 18 | / | / | / |
| 17:8 | R/W | 0 | LRCK_PERIOD<br>It is used to program the number of BCLKs per channel of sample frame. This value is interpreted as follow:<br>PCM mode: Number of BCLKs within (Left + Right) channel width<br>I2S / Left-Justified / Right-Justified mode: Number of BCLKs within each individual channel width (Left or Right)<br>N+1<br>For example:<br>n = 7:    8 BCLK width<br>…<br>n = 1023:    1024 BCLKs width |
| 7 | R/W | 0 | BCLK_POLARITY<br>0: normal mode, negative edge drive and positive edge sample<br>1: invert mode, positive edge drive and negative edge sample |
| 6:4 | R/W | 3 | SR<br>Sample Resolution<br>0: Reserved<br>1: 8-bit<br>2: 12-bit<br>3: 16-bit<br>4: 20-bit<br>5: 24-bit<br>6: 28-bit<br>7: 32-bit |
| 3 | R/W | 0 | EDGE_TRANSFER<br>0: SDO drive data and SDI sample data at the different BCLK edge<br>1: SDO drive data and SDI sample data at the same BCLK edge<br>BCLK_POLARITY = 0, use negative edge<br>BCLK_POLARITY = 1, use positive edge |

| | | | SW |
| --- | --- | --- | --- |
| | | | Slot Width Select |
| | | | 0: Reserved |
| | | | 1: 8-bit |
| | | | 2: 12-bit |
| | | | 3: 16-bit |
| | | | 4: 20-bit |
| | | | 5: 24-bit |
| | | | 6: 28-bit |
| 2:0 | R/W | 0x3 | 7: 32-bit |

### 7.7.7.3. Digital Audio Format Register1 (Default Value: 0x0000_0030)

| Offset: 0x08 | | | Register Name: **DA_FMT1** |
| --- | --- | --- | --- |
| Bit | R/W | Default/Hex | Description |
| 31:8 | / | / | |
| 7 | R/W | 0 | RX MLS<br>MSB / LSB First Select<br>0: MSB First<br>1: LSB First |
| 6 | R/W | 0 | TX MLS<br>MSB / LSB First Select<br>0: MSB First<br>1: LSB First |
| 5:4 | R/W | 3 | SEXT<br>Sign Extend in slot [sample resolution < slot width]<br>0: Zeros or audio gain padding at LSB position<br>1: Sign extension at MSB position<br>2: Reserved<br>3: Transfer 0 after each sample in each slot |
| 3:2 | R/W | 0 | RX_PDM<br>PCM Data Mode<br>0: Linear PCM<br>1: reserved<br>2: 8-bits u-law<br>3: 8-bits A-law |
| 1:0 | R/W | 0 | TX_PDM<br>PCM Data Mode<br>0: Linear PCM<br>1: reserved<br>2: 8-bits u-law<br>3: 8-bits A-law |

### 7.7.7.4. Digital Audio Interrupt Status Register(Default Value: 0x0000_0010)

| Offset: 0x0C | | | Register Name: **DA_ISTA** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:7 | / | / | / |
| 6 | R/W | 0 | TXU_INT<br>TX FIFO Under run Pending Interrupt<br>0: No Pending Interrupt<br>1: FIFO Under run Pending Interrupt<br>Write 1 to clear this interrupt |
| 5 | R/W | 0 | TXO_INT<br>TX FIFO Overrun Pending Interrupt<br>0: No Pending Interrupt<br>1: FIFO Overrun Pending Interrupt<br>Write '1' to clear this interrupt |
| 4 | R/W | 1 | TXE_INT<br>TX FIFO Empty Pending Interrupt<br>0: No Pending IRQ<br>1: FIFO Empty Pending Interrupt when data in TX FIFO are less than TX trigger level<br>Write '1' to clear this interrupt or automatic clear if interrupt condition fails. |
| 3 | / | / | / |
| 2 | R/W | 0 | RXU_INT<br>RX FIFO Under run Pending Interrupt<br>0: No Pending Interrupt<br>1:FIFO Under run Pending Interrupt<br>Write 1 to clear this interrupt |
| 1 | R/W | 0 | RXO_INT<br>RX FIFO Overrun Pending Interrupt<br>0: No Pending IRQ<br>1: FIFO Overrun Pending IRQ<br>Write '1' to clear this interrupt |
| 0 | R/W | 0 | RXA_INT<br>RX FIFO Data Available Pending Interrupt<br>0: No Pending IRQ<br>1: Data Available Pending IRQ when data in RX FIFO are more than RX trigger level<br>Write '1' to clear this interrupt or automatic clear if interrupt condition fails. |

#### 7.7.7.5. Digital Audio RX FIFO register(Default Value: 0x0000_0000)

| Offset: 0x10 | | | Register Name: **DA_RXFIFO** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R | 0 | RX_DATA<br>RX Sample<br>Host can get one sample by reading this register. The left channel sample data is first and then the right channel sample. |

#### 7.7.7.6. Digital Audio FIFO Control Register (Default Value: 0x0004_00F0)

| Offset: 0x14 | | | Register Name: **DA_FCTL** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | R/W | 0 | HUB_EN<br>Audio hub enable<br>0: Disable<br>1: Enable |
| 30:26 | / | / | / |
| 25 | R/W | 0 | FTX<br>Write '1' to flush TX FIFO, self clear to '0'. |
| 24 | R/W | 0 | FRX<br>Write '1' to flush RX FIFO, self clear to '0'. |
| 23:19 | / | / | / |
| 18:12 | R/W | 0x40 | TXTL<br>TX FIFO Empty Trigger Level<br>Interrupt and DMA request trigger level for TXFIFO normal condition<br>Trigger Level = TXTL |
| 11:10 | / | / | / |
| 9:4 | R/W | 0xF | RXTL<br>RX FIFO Trigger Level<br>Interrupt and DMA request trigger level for RXFIFO normal condition<br>Trigger Level = RXTL + 1 |
| 3 | / | / | / |
| 2 | R/W | 0 | TXIM<br>TX FIFO Input Mode (Mode 0, 1)<br>0: Valid data at the MSB of TXFIFO register<br>1: Valid data at the LSB of TXFIFO register<br>Example for 20-bits transmitted audio sample:<br>Mode 0: FIFO_I[31:0] = {APB_WDATA[31:12], 12'h0}<br>Mode 1: FIFO_I[31:0] = {APB_WDATA[19:0], 12'h0} |
| 1:0 | R/W | 0 | RXOM<br>RX FIFO Output Mode (Mode 0, 1, 2, 3)<br>00: Expanding '0' at LSB of DA_RXFIFO register. |

| | | | 01: Expanding received sample sign bit at MSB of DA_RXFIFO register. |
| | | | 10: Truncating received samples at high half-word of DA_RXFIFO register and low half-word of DA_RXFIFO register is filled by '0'. |
| | | | 11: Truncating received samples at low half-word of DA_RXFIFO register and high half-word of DA_RXFIFO register is expanded by its sign bit. |
| | | | Example for 20-bits received audio sample: |
| | | | Mode 0: APB_RDATA[31:0] = {FIFO_O[31:12], 12'h0} |
| | | | Mode 1: APB_RDATA [31:0] = {12{FIFO_O[31]}, FIFO_O[31:12]} |
| | | | Mode 2: APB_RDATA [31:0] = {FIFO_O[31:16], 16'h0} |
| | | | Mode 3: APB_RDATA [31:0] = {16{FIFO_O[31], FIFO_O[31:16]]} |

### 7.7.7.7.   Digital Audio FIFO Status Register (Default Value: 0x1080_0000)

| Offset: 0x18 | | | Register Name: **DA_FSTA** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:29 | / | / | / |
| 28 | R | 1 | TXE<br>TX FIFO Empty<br>0: No room for new sample in TX FIFO<br>1: More than one room for new sample in TX FIFO (>= 1 word) |
| 27:24 | / | / | / |
| 23:16 | R | 0x80 | TXE_CNT<br>TX FIFO Empty Space Word Counter |
| 15:9 | / | / | / |
| 8 | R | 0 | RXA<br>RX FIFO Available<br>0: No available data in RX FIFO<br>1: More than one sample in RX FIFO (>= 1 word) |
| 7 | / | / | / |
| 6:0 | R | 0 | RXA_CNT<br>RX FIFO Available Sample Word Counter |

### 7.7.7.8.   Digital Audio DMA & Interrupt Control Register(Default Value: 0x0000_0000)

| Offset: 0x1C | | | Register Name: **DA_INT** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:8 | / | / | / |
| 7 | R/W | 0 | TX_DRQ<br>TX FIFO Empty DRQ Enable<br>0: Disable<br>1: Enable |
| 6 | R/W | 0 | TXUI_EN |

| | | | TX FIFO Under run Interrupt Enable<br>0: Disable<br>1: Enable |
|---|---|---|---|
| 5 | R/W | 0 | TXOI_EN<br>TX FIFO Overrun Interrupt Enable<br>0: Disable<br>1: Enable<br>When set to '1', an interrupt happens when writing new audio data if TX FIFO is full. |
| 4 | R/W | 0 | TXEI_EN<br>TX FIFO Empty Interrupt Enable<br>0: Disable<br>1: Enable |
| 3 | R/W | 0 | RX_DRQ<br>RX FIFO Data Available DRQ Enable<br>0: Disable<br>1: Enable<br>When set to '1', RXFIFO DMA Request line is asserted if Data is available in RX FIFO. |
| 2 | R/W | 0 | RXUI_EN<br>RX FIFO Under run Interrupt Enable<br>0: Disable<br>1: Enable |
| 1 | R/W | 0 | RXOI_EN<br>RX FIFO Overrun Interrupt Enable<br>0: Disable<br>1: Enable |
| 0 | R/W | 0 | RXAI_EN<br>RX FIFO Data Available Interrupt Enable<br>0: Disable<br>1: Enable |

### 7.7.7.9.   Digital Audio TX FIFO register(Default Value: 0x0000_0000)

| Offset: 0x20 | | | Register Name: **DA_TXFIFO** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | W | 0 | TX_DATA<br>TX Sample<br>Transmitting left, right channel sample data should be written this register one by one. The left channel sample data is first and then the right channel sample. |

### 7.7.7.10. Digital Audio Clock Divide Register(Default Value: 0x0000_0000)

| Offset: 0x24 | | | Register Name: **DA_CLKD** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:9 | / | / | / |
| 8 | R/W | 0 | MCLKO_EN<br>0: Disable MCLK Output<br>1: Enable MCLK Output<br>Notes: Whether in Slave or Master mode, when this bit is set to 1, MCLK should be output. |
| 7:4 | R/W | 0 | BCLKDIV<br>BCLK Divide Ratio from PLL2<br>0: reserved<br>1: Divide by 1<br>2: Divide by 2<br>3: Divide by 4<br>4: Divide by 6<br>5: Divide by 8<br>6: Divide by 12<br>7: Divide by 16<br>8: Divide by 24<br>9: Divide by 32<br>10: Divide by 48<br>11: Divide by 64<br>12: Divide by 96<br>13: Divide by 128<br>14: Divide by 176<br>15: Divide by 192 |
| 3:0 | R/W | 0 | MCLKDIV<br>MCLK Divide Ratio from PLL2 Output<br>0: reserved<br>1: Divide by 1<br>2: Divide by 2<br>3: Divide by 4<br>4: Divide by 6<br>5: Divide by 8<br>6: Divide by 12<br>7: Divide by 16<br>8: Divide by 24<br>9: Divide by 32<br>10: Divide by 48<br>11: Divide by 64<br>12: Divide by 96<br>13: Divide by 128<br>14: Divide by 176 |

| | | | 15: Divide by 192 |
|---|---|---|---|

#### 7.7.7.11. Digital Audio TX Counter register(Default Value: 0x0000_0000)

| Offset: 0x28 | | | Register Name: **DA_TXCNT** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R/W | 0 | TX_CNT<br>TX Sample Counter<br>The audio sample number of sending into TXFIFO. When one sample is put into TXFIFO by DMA or by host IO, the TX sample counter register increases by one. The TX sample counter register can be set to any initial valve at any time. After been updated by the initial value, the counter register should count on base of this initial value. |

#### 7.7.7.12. Digital Audio RX Counter register(Default Value: 0x0000_0000)

| Offset: 0x2C | | | Register Name: **DA_RXCNT** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:0 | R/W | 0 | RX_CNT<br>RX Sample Counter<br>The audio sample number of writing into RXFIFO. When one sample is written by Digital Audio Engine, the RX sample counter register increases by one. The RX sample counter register can be set to any initial valve at any time. After been updated by the initial value, the counter register should count on base of this initial value. |

#### 7.7.7.13. Digital Audio Channel Configuration register(Default Value: 0x0000_0000)

| Offset: 0x30 | | | Register Name: **DA_CHCFG** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:10 | / | / | / |
| 9 | R/W | 0 | TX_SLOT_HIZ<br>0: normal mode for the last half cycle of BCLK in the slot<br>1: turn to hi-z state for the last half cycle of BCLK in the slot |
| 8 | R/W | 0 | TXn_STATE<br>0: transfer level 0 when not transferring slot<br>1: turn to hi-z state when not transferring slot |
| 7 | / | / | / |
| 6:4 | R/W | 0 | RX_SLOT_NUM<br>RX Channel/Slot Number which between CPU/DMA and FIFO<br>0: 1 channel or slot |

| | | | ... |
| | | | 7: 8 channels or slots |
| 3 | / | / | / |
| | | | TX_SLOT_NUM |
| | | | TX Channel/Slot Number which between CPU/DMA and FIFO |
| | | | 0: 1 channel or slot |
| | | | ... |
| 2:0 | R/W | 0 | 7: 8 channels or slots |

### 7.7.7.14.  Digital Audio TXn Channel Select register(Default Value: 0x0000_0000)

| Offset: 0x34 + n*4 (n = 0, 1, 2, 3) | | | Register Name: **DA_TXnCHSEL** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31:14 | / | / | / |
| 13:12 | R/W | 0 | TXn_OFFSET<br>TXn offset tune, TXn data offset to LRCK<br>0: no offset<br>n: data is offset by n BCLKs to LRCK |
| 11:4 | R/W | 0 | TXn_CHEN<br>TXn Channel (slot) enable, bit[11:4] refer to slot [7:0]. When one or more slot(s) is(are) disabled, the affected slot(s) is(are) set to disable state<br>0: disable<br>1: enable |
| 3 | / | / | / |
| | | | TXn_CHSEL<br>TXn Channel (slot) number Select for each output<br>0: 1 channel / slot<br>... |
| 2:0 | R/W | 0 | 7: 8 channels / slots |

### 7.7.7.15.  Digital Audio TXn Channel Mapping Register(Default Value: 0x0000_0000)

| Offset: 0x44 + n*4 (n = 0, 1, 2, 3) | | | Register Name: **DA_TXnCHMAP** |
|---|---|---|---|
| Bit | R/W | Default/Hex | Description |
| 31 | / | / | / |
| 30:28 | R/W | 0 | TXn_CH7_MAP<br>TXn Channel7 Mapping<br>0: 1st sample<br>...<br>7: 8th sample |
| 27 | / | / | / |
| 26:24 | R/W | 0 | TXn_CH6_MAP |

| | | | TXn Channel6 Mapping<br>0: 1st sample<br>…<br>7: 8th sample |
|---|---|---|---|
| 23 | / | / | / |
| 22:20 | R/W | 0 | TXn_CH5_MAP<br>TXn Channel5 Mapping<br>0: 1st sample<br>…<br>7: 8th sample |
| 19 | / | / | / |
| 18:16 | R/W | 0 | TXn_CH4_MAP<br>TXn Channel4 Mapping<br>0: 1st sample<br>…<br>7: 8th sample |
| 15 | / | / | / |
| 14:12 | R/W | 0 | TXn_CH3_MAP<br>TXn Channel3 Mapping<br>0: 1st sample<br>…<br>7: 8th sample |
| 11 | / | / | / |
| 10:8 | R/W | 0 | TXn_CH2_MAP<br>TXn Channel2 Mapping<br>0: 1st sample<br>…<br>7: 8th sample |
| 7 | / | / | / |
| 6:4 | R/W | 0 | TXn_CH1_MAP<br>TXn Channel1 Mapping<br>0: 1st sample<br>…<br>7: 8th sample |
| 3 | / | / | / |
| 2:0 | R/W | 0 | TXn_CH0_MAP<br>TXn Channel0 Mapping<br>0: 1st sample<br>…<br>7: 8th sample |

### 7.7.7.16. Digital Audio RX Channel Select register(Default Value: 0x0000_0000)

| Offset: 0x54 | Register Name: **DA_RXCHSEL** |
|---|---|

| Bit | R/W | Default/Hex | Description |
|---|---|---|---|
| 31:14 | / | / | / |
| 13:12 | R/W | 0 | RX_OFFSET<br>RX offset tune, RX data offset to LRCK<br>0: no offset<br>n: data is offset by n BCLKs to LRCK |
| 11:3 | / | / | |
| 2:0 | R/W | 0 | RX_CHSEL<br>RX Channel (slot) number Select for input<br>0: 1 channel / slot<br>…<br>7: 8 channels / slots |

### 7.7.7.17. Digital Audio RX Channel Mapping Register (Default Value: 0x0000_0000)

| Offset: 0x58 | | | Register Name: DA_RXCHMAP |
|---|---|---|---|
| **Bit** | **R/W** | **Default/Hex** | **Description** |
| 31 | / | / | / |
| 30:28 | R/W | 0 | RX_CH7_MAP<br>RX Channel7 Mapping<br>0: 1st sample<br>…<br>7: 8th sample |
| 27 | / | / | / |
| 26:24 | R/W | 0 | RX_CH6_MAP<br>RX Channel6 Mapping<br>0: 1st sample<br>…<br>7: 8th sample |
| 23 | / | / | / |
| 22:20 | R/W | 0 | RX_CH5_MAP<br>RX Channel5 Mapping<br>0: 1st sample<br>…<br>7: 8th sample |
| 19 | / | / | / |
| 18:16 | R/W | 0 | RX_CH4_MAP<br>RX Channel4 Mapping<br>0: 1st sample<br>…<br>7: 8th sample |
| 15 | / | / | / |
| 14:12 | R/W | 0 | RX_CH3_MAP |

| | | | |
|---|---|---|---|
| | | | RX Channel3 Mapping<br>0: 1st sample<br>…<br>7: 8th sample |
| 11 | / | / | / |
| 10:8 | R/W | 0 | RX_CH2_MAP<br>RX Channel2 Mapping<br>0: 1st sample<br>…<br>7: 8th sample |
| 7 | / | / | / |
| 6:4 | R/W | 0 | RX_CH1_MAP<br>TX Channel1 Mapping<br>0: 1st sample<br>…<br>7: 8th sample |
| 3 | / | / | / |
| 2:0 | R/W | 0 | RX_CH0_MAP<br>RX Channel0 Mapping<br>0: 1st sample<br>…<br>7: 8th sample |

## 7.8.  EMAC

### 7.8.1.  Overview

The Ethernet MAC(EMAC) controller enables a host to transmit and receive data over Ethernet in compliance with the IEEE 802.3-2002 standard. It supports 10M/100M/1000M external PHY with MII/ RGMII interface in both full and half duplex mode. The Ethernet MAC-DMA is designed for packet-oriented data transfers based on a linked list of descriptors. 4K Byte TXFIFO and 16K Byte RXFIFO are provided to keep continuous transmission and reception. Flow Control, CRC Pad & Stripping, and address filtering are also supported in this module.

The Ethernet MAC Controller includes the following features:

- Supports 10/100/1000Mbps data transfer rates
- Supports MII/RGMII PHY interface
- Supports both full-duplex and half-duplex operation
- Automatic CRC and pad generation controllable on a per-frame basis
- Options for Automatic Pad/CRC Stripping on receive frames
- Programmable frame length to support Standard or Jumbo Ethernet frames with sizes up to 16 KB
- Programmable Inter Frame Gap (40-96 bit times in steps of 8)
- Supports a variety of flexible address filtering modes
- Separate 32-bit status returned for transmission and reception packets
- Optimization for packet-oriented DMA transfers with frame delimiters
- Dual-buffer (ring) or linked-list (chained) descriptor chaining
- Descriptor architecture, allowing large blocks of data transfer with minimum CPU intervention; each descriptor can transfer up to 4 KB of data
- Comprehensive status reporting for normal operation and transfers with errors
- 4KB TXFIFO for transmission packets and 16KB RXFIFO for reception packets
- Programmable interrupt options for different operational conditions

## 7.8.2.  Block Diagram

The EMAC Controller system block diagram is shown below:
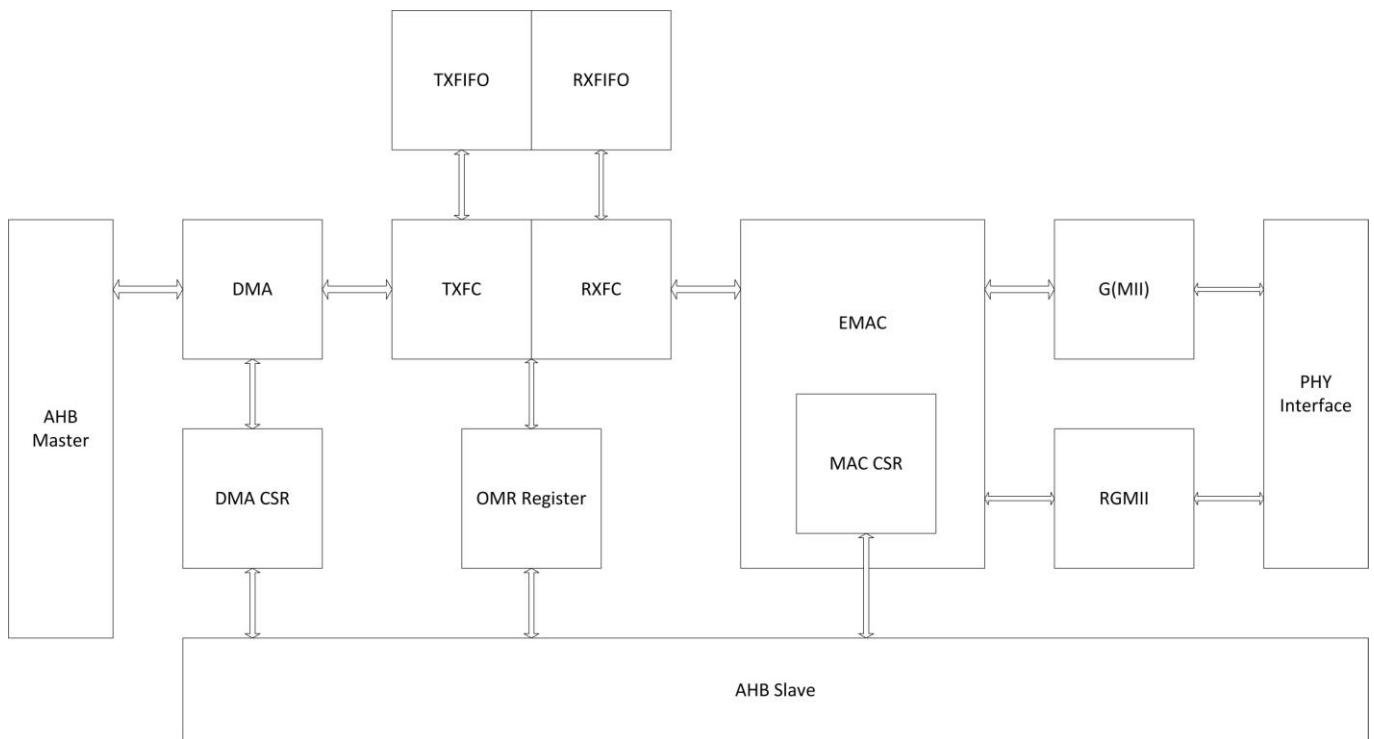


Figure 7-14. EMAC Block Diagram

## 7.8.3.  EMAC Register List

| Module Name | Base Address |
|---|---|
| EMAC | 0x01C30000 |

EMAC_CORE:

| Register Name | Offset | Description |
|---|---|---|
| EMAC_CONTROL | 0x00 | MAC Configuration Register |
| EMAC_FRAME_FILTER | 0x04 | MAC Frame Filter Register |
| EMAC_HASH_HIGH | 0x08 | Hash Table High Register |
| EMAC_HASH_LOW | 0x0C | Hash Table Low Register |
| EMAC_GMII_ADDR | 0x10 | GMII Address Register |
| EMAC_GMII_DATA | 0x14 | GMII Data Register |
| EMAC_FLOW_CTRL | 0x18 | Flow Control Register |
| EMAC_INT_STATUS | 0x38 | Interrupt Register |
| EMAC_INT_MASK | 0x3C | Interrupt Mask Register |
| EMAC_ADDR_HIGH_0 | 0x40 | MAC Address High Register 0 |
| EMAC_ADDR_LOW_0 | 0x44 | MAC Address High Register 0 |

| EMAC_ADDR_HIGH_x | 0x40+8*x | MAC Address High Register x |
|---|---|---|
| EMAC_ADDR_LOW_x | 0x44+8*x | MAC Address Low Register x |
| EMAC_RGMII_STATUS | 0xD8 | TBI Extended Status Register |

EMAC_DMA:

| Register Name | Offset | Description |
|---|---|---|
| EDMA_BUS_MODE | 0x1000 | Bus Mode Register |
| EDMA_XMT_POLL | 0x1004 | Transmit Poll Demand Register |
| EDMA_RCV_POLL | 0x1008 | Receive Poll Demand Register |
| EDMA_RCV_LIST | 0x100C | Receive Descriptor List Address Register |
| EDMA_XMT_LIST | 0x1010 | Transmit Descriptor List Address Register |
| EDMA_STATUS | 0x1014 | Status Register |
| EDMA_OPERATION | 0x1018 | Operation Mode Register |
| EDMA_INTR_ENA | 0x101C | Interrupt Enable Register |
| EDMA_MISSED_FRAME | 0x1020 | Missed Frame and Buffer Overflow Counter Register |
| EDMA_CUR_TX_DESC | 0x1048 | Current Host Transmit Descriptor Register |
| EDMA_CUR_RX_DESC | 0x104C | Current Host Receive Descriptor Register |
| EDMA_CUR_TX_BUF | 0x1050 | Current Host Transmit Buffer Address Register |
| EDMA_CUR_RX_BUF | 0x1054 | Current Host Receive Buffer Address Register |

## 7.8.4.   EMAC Core Register Description

### 7.8.4.1.   EMAC MAC Configuration Register(Default Value: 0x0000_0000)

| Offset: 0x00 | | | Register Name: EMAC_CONTROL |
|---|---|---|---|
| Bit | Read/Write | Default | Description |
| 31:25 | / | / | / |
| 24 | R/W | 0 | TC<br>Transmit Configuration in RGMII<br>0: No transmission information is driven to the PHY.<br>1: Enables the transmission of duplex mode, link speed, and link up/down information to the PHY in the RGMII ports. |
| 23 | R/W | 0 | WD<br>Watchdog Disable<br>0: Allows no more than 2,048 bytes (10,240 if JE is set high) of the frame being received and cuts off any bytes received after that<br>1: Disable disables the watchdog timer on the receiver, and can receive frames of up to 16,384 bytes |
| 22 | R/W | 0 | JD<br>Jabber Disable<br>0: Cuts off the transmitter if the application sends out more than 2,048 bytes of data (10,240 if JE is set high) during transmission. |

| | | | |
|---|---|---|---|
| | R/W | | 1: Disables the jabber timer on the transmitter, and can transfer frames of up to 16,384 bytes. |
| 21 | R/W | 0 | BE<br>Frame Burst Enable<br>When this bit is set, the EMAC allows frame bursting during transmission in GMII Half- Duplex mode. |
| 20 | R/W | 0 | JE<br>Jumbo Frame Enable<br>When this bit is set, EMAC allows Jumbo frames of 9,018 bytes (9,022 bytes for VLAN tagged frames) without reporting a giant frame error in the receive frame status. |
| 19:17 | R/W | 0 | IFG<br>Inter-Frame Gap<br>These bits control the minimum IFG between frames during transmission.<br>000: 96 bit times<br>001: 88 bit times<br>010: 80 bit times<br>….<br>111: 40 bit times<br>Note that in Half-Duplex mode, the minimum IFG can be configured for 64 bit times (IFG = 100) only. Lower values are not considered. In 1000-Mbps mode, the minimum IFG supported is 64 bit times (and above) in the EMAC-CORE configuration and 80 bit times (and above) in other system configurations. |
| 16 | R/W | 0 | DCRS<br>Disable Carrier Sense During Transmission<br>When set high, this bit makes the MAC transmitter ignore the (G)MII CRS signal during frame transmission in Half-Duplex mode. |
| 15 | R/W | 0 | PS<br>Port Select<br>0: GMII (1000 Mbps)<br>1: MII (10/100 Mbps) |
| 14 | R/W | 0 | FES<br>Indicates the speed in Fast Ethernet (MII) mode<br>0: 10 Mbps<br>1: 100 Mbps |
| 13 | R/W | 0 | DO<br>Disable Receive Own<br>When this bit is set, the EMAC disables the reception of frames when the gmii_txen_o is asserted in Half-Duplex mode. When this bit is reset, the EMAC receives all packets that are given by the PHY while transmitting. This bit is not applicable if the EMAC is operating in Full-Duplex mode. |
| 12 | R/W | 0 | LM<br>Loopback Mode<br>When this bit is set, the EMAC operates in loopback mode at GMII/MII. |
| 11 | R/W | 0 | DM |

| | | | Duplex Mode |
|---|---|---|---|
| | | | When this bit is set, the EMAC operates in a Full-Duplex mode where it can transmit and receive simultaneously. |
| 10 | R/W | 0 | IPC<br>Checksum Offload<br>When this bit set, the EMAC calculates the 16-bit one's complement of the one's complement sum of all received Ethernet frame payloads, and checks whether the IPv4 Header Checksum is correct for received frame and gives the status in the receive status word. |
| 9 | R/W | 0 | DR<br>Disable Retry<br>When this bit is set, the EMAC will attempt only 1 transmission. When a collision occurs on the GMII/MII, the EMAC will ignore the current frame transmission and report a Frame Abort with excessive collision error in the transmit frame status. When this bit is reset, the EMAC will attempt retries based on the settings of BL. This bit is applicable only to Half-Duplex mode and is reserved (RO with default value) in Full- Duplex-only configuration. |
| 8 | R/W | 0 | LUD<br>Link Up/Down<br>Indicates whether the link is up or down during the transmission of configuration in RGMII/SGMII interface<br>0: Link Down<br>1: Link Up |
| 7 | R/W | 0 | ACS<br>Automatic Pad/CRC Stripping<br>When this bit is set, the EMAC strips the Pad/FCS field on incoming frames only if the length's field value is less than or equal to 1,500 bytes. All received frames with length field greater than or equal to 1,501 bytes are passed to the application without stripping the Pad/FCS field. When this bit is reset, the EMAC will pass all incoming frames to the Host unmodified.<br>When this bit is reset, the EMAC will pass all incoming frames to the Host unmodified |
| 6:5 | R/W | 0 | BL<br>Back-Off Limit<br>The Back-Off limit determines the random integer number (r) of slot time delays (4,096 bit times for 1000 Mbps and 512 bit times for 10/100 Mbps) the EMAC waits before rescheduling a transmission attempt during retries after a collision. This bit is applicable only to Half-Duplex mode.<br>00: k = min (n, 10)<br>01: k = min (n, 8)<br>10: k = min (n, 4)<br>11: k = min (n, 1)<br>where n = retransmission attempt. The random integer r takes the value in the range $0 \le r < 2k$ |
| 4 | R/W | 0 | DC |

| | | | Deferral Check<br>0: Disable<br>1: Enable |
|---|---|---|---|
| 3 | R/W | 0 | TE<br>Transmitter Enable<br>When this bit is set, the transmitter state machine of the EMAC is enabled for transmission on the GMII/MII. When this bit is reset, the EMAC transmit state machine is disabled after the completion of the transmission of the current frame, and will not transmit any further frames. |
| 2 | R/W | 0 | RE<br>Receiver Enable<br>When this bit is set, the receiver state machine of the EMAC is enabled for receiving frames from the GMII/MII. When this bit is reset, the EMAC receive state machine is disabled after the completion of the reception of the current frame, and will not receive any further frames from the GMII/MII |
| 0:1 | / | / | / |

### 7.8.4.2. EMAC MAC Frame Filter Register(Default Value: 0x0000_0000)

| Offset: 0x04 | | | Register Name: EMAC_FRAME_FILTER |
|---|---|---|---|
| Bit | Read/Write | Default | Description |
| 31 | R/W | 0 | RA<br>Receive All<br>When this bit is set, the EMAC Receiver module passes to the Application all frames received irrespective of whether they pass the address filter. The result of the SA/DA filtering is updated (pass or fail) in the corresponding bits in the Receive Status Word. When this bit is reset, the Receiver module passes to the Application only those frames that pass the SA/DA address filter |
| 30:11 | / | / | / |
| 10 | R/W | 0 | HPF<br>Hash or Perfect Filter<br>When set, this bit configures the address filter to pass a frame if it matches either the perfect filtering or the hash filtering as set by HMC or HUC bits. When low and if the HUC/HMC bit is set, the frame is passed only if it matches the Hash filter. This bit is reserved (and RO) if the Hash filter is not selected during core configuration. |
| 9 | R/W | 0 | SAF<br>Source Address Filter Enable<br>The EMAC core compares the SA field of the received frames with the values programmed in the enabled SA registers. If the comparison matches, then the SAMatch bit of RxStatus Word is set high. When this bit is set high and the SA filter fails, the EMAC drops the frame. When this bit is reset, then the EMAC Core forwards the received frame to the application and with the updated SA Match bit of the RxStatus depending on the SA address |

| | | | |
|---|---|---|---|
| | | | comparison. |
| 8 | R/W | 0 | SAIF<br>SA Inverse Filtering<br>When this bit is set, the Address Check block operates in inverse filtering mode for the SA address comparison. The frames whose SA matches the SA registers will be marked as failing the SA Address filter. When this bit is reset, frames whose SA does not match the SA registers will be marked as failing the SA Address filter. |
| 7:6 | R/W | 0 | PCF<br>Pass Control Frames<br>These bits control the forwarding of all control frames (including unicast and multicast PAUSE frames). Note that the processing of PAUSE control frames depends only on RFE of EMAC Flow Control Register[2].<br>00: EMAC filters all control frames from reaching the application<br>01: EMAC filters all control frames from reaching the application<br>10: EMAC forwards all control frames to application even if they fail the Address Filter<br>11: EMAC forwards control frames that pass the Address Filter |
| 5 | R/W | 0 | DBF<br>Disable Broadcast Frames<br>When this bit is set, the AFM module filters all incoming broadcast frames. When this bit is reset, the AFM module passes all received broadcast frames. |
| 4 | R/W | 0 | PM<br>Pass All Multicast<br>When set, this bit indicates that all received frames with a multicast destination address (first bit in the destination address field is '1') are passed. When reset, filtering of multicast frame depends on HMC bit. |
| 3 | R/W | 0 | DAIF<br>DA Inverse Filtering<br>When this bit is set, the Address Check block operates in inverse filtering mode for the DA address comparison for both unicast and multicast frames. When reset, normal filtering of frames is performed. |
| 2 | R/W | 0 | HMC<br>Hash Multicast<br>When set, MAC performs destination address filtering of received multicast frames according to the hash table. When reset, the MAC performs a perfect destination address filtering for multicast frames, that is, it compares the DA field with the values programmed in DA registers. |
| 1 | R/W | 0 | HUC<br>Hash Unicast<br>When set, MAC performs destination address filtering of unicast frames according to the hash table. When reset, the MAC performs a perfect destination address filtering for unicast frames, that is, it compares the DA field with the values programmed in DA registers. |
| 0 | R/W | 0 | PR |

| | | | Promiscuous Mode |
| | | | When this bit is set, the Address Filter module passes all incoming frames regardless of its destination or source address. The SA/DA Filter Fails status bits of the Receive Status Word will always be cleared when PR is set. |

### 7.8.4.3. EMAC Hash Table High Register(Default Value: 0x0000_0000)

| Offset: 0x08 | | | Register Name: EMAC_HASH_HIGH |
| --- | --- | --- | --- |
| Bit | Read/Write | Default | Description |
| 31:0 | R/W | 0 | HTH<br>Hash Table High<br>This field contains the upper 32 bits of Hash table. |

### 7.8.4.4. EMAC Hash Table Low Register(Default Value: 0x0000_0000)

| Offset: 0x0C | | | Register Name: EMAC_HASH_LOW |
| --- | --- | --- | --- |
| Bit | Read/Write | Default | Description |
| 31:0 | R/W | 0 | HTL<br>Hash Table Low<br>This field contains the lower 32 bits of Hash table. |

### 7.8.4.5. EMAC GMII Address Register(Default Value: 0x0000_0000)

| Offset: 0x10 | | | Register Name: EMAC_GMII_ADDR |
| --- | --- | --- | --- |
| Bit | Read/Write | Default | Description |
| 31:16 | / | / | / |
| 15:11 | R/W | 0 | PA<br>PHY Address<br>This field tells which of the 32 possible PHY devices are being accessed |
| 10:6 | R/W | 0 | GR<br>GMII Register<br>These bits select the desired GMII register in the selected PHY device |
| 5 | / | / | / |
| 4:2 | R/W | 0 | CR<br>CSR Clock Range<br>The CSR Clock Range selection determines the AHB frequency and is used to decide the frequency of the MDC clock:<br>Selection    ahb_clk       MDC Clock<br>   000      60-100 MHz    ahb_clk /42<br>   001      100-150 MHz   ahb_clk /62<br>   010      20-35 MHz    ahb_clk /16 |

| | | | 011       35-60 MHz     ahb_clk /26 |
| | | | 100      150-250 MHz  ahb_clk /102 |
| | | | 101      250-300 MHz  ahb_clk /124 |
| | | | Others     Reserved |
| 1 | R/W | 0 | GW<br>GMII Write<br>When set, this bit tells the PHY that this will be a Write operation using the GMII Data register. If this bit is not set, this will be a Read operation, placing the data in the GMII Data register. |
| 0 | R/W | 0 | GB<br>GMII Busy<br>This bit should read a logic 0 before writing to EMAC_GMII_ADDR and EMAC_GMII_DATA. This bit must also be set to 0 during a Write to EMAC_GMII_ADDR. During a PHY register access, this bit will be set to 1'b1 by the Application to indicate that a Read or Write access is in progress. EMAC_GMII_DATA should be kept valid until this bit is cleared by the EMAC during a PHY Write operation. The EMAC_GMII_DATA is invalid until this bit is cleared by the EMAC during a PHY Read operation. The EMAC_GMII_ADDR should not be written to until this bit is cleared |

### 7.8.4.6.  EMAC GMII Data Register(Default Value: 0x0000_0000)

| Offset: 0x14 | | | Register Name: EMAC_GMII_DATA |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default** | **Description** |
| 31:16 | / | / | / |
| 15:0 | R/W | 0 | GD<br>GMII Data<br>This contains the 16-bit data value read from the PHY after a Management Read operation or the 16-bit data value to be written to the PHY before a Management Write operation. |

### 7.8.4.7.  EMAC Flow Control Register(Default Value: 0x0000_0000)

| Offset: 0x18 | | | Register Name: EMAC_FLOW_CTRL |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default** | **Description** |
| 31:16 | R/W | 0 | PT<br>Pause Time<br>This field holds the value to be used in the Pause Time field in the transmit control frame. |
| 15:8 | / | / | / |
| 7 | R/W | 0 | DZPQ<br>Disable Zero-Quanta Pause |

| | | | When set, this bit disables the automatic generation of Zero-Quanta Pause Control frames. When this bit is reset, normal operation with automatic Zero-Quanta Pause Control frame generation is enabled. |
|---|---|---|---|
| 6 | / | / | / |
| 5:4 | R/W | 0 | PLT<br>Pause Low Threshold<br>This field configures the threshold of the PAUSE timer at which the input flow control signal is checked for automatic retransmission of PAUSE Frame. The threshold values should be always less than the Pause Time configured in Bits[31:16]. |
| 3 | R/W | 0 | UP<br>Unicast Pause Frame Detect<br>When this bit is set, the EMAC will detect the Pause frames with the station's unicast address specified in MAC Address0 High Register and MAC Address0 Low Register, in addition to the detecting Pause frames with the unique multicast address. When this bit is reset, the EMAC will detect only a Pause frame with the unique multicast address specified in the 802.3x standard. |
| 2 | R/W | 0 | RFE<br>Receive Flow Control Enable<br>When this bit is set, the EMAC will decode the received Pause frame and disable its transmitter for a specified (Pause Time) time. When this bit is reset, the decode function of the Pause frame is disabled. |
| 1 | R/W | 0 | TFE<br>Transmit Flow Control Enable<br>In Full-Duplex mode, when this bit is set, the EMAC enables the flow control operation to transmit Pause frames. When this bit is reset, the flow control operation in the EMAC is disabled, and the EMAC will not transmit any Pause frames.<br>In Half-Duplex mode, when this bit is set, the EMAC enables the back-pressure operation. When this bit is reset, the backpressure feature is disabled. |
| 0 | R/W | 0 | FCB/BPA<br>Flow Control Busy/Backpressure Activate<br>This bit initiates a Pause Control frame in Full-Duplex mode and activates the backpressure function in Half-Duplex mode if TFE bit is set. In Full-Duplex mode, this bit should be read as 1'b0 before writing to the Flow Control register. To initiate a Pause control frame, the Application must set this bit to 1'b1. During a transfer of the Control Frame, this bit will continue to be set to signify that a frame transmission is in progress. After the completion of Pause control frame transmission, the EMAC will reset this bit to 1'b0. The Flow Control register should not be written to until this bit is cleared. In Half-Duplex mode, when this bit is set (and TFE is set), then backpressure is asserted by the EMAC Core. During backpressure, when the EMAC receives a new frame, the transmitter starts sending a JAM pattern resulting in a collision. |

### 7.8.4.8. EMAC Interrupt Status Register(Default Value: 0x0000_0000)

| Offset: 0x38 | | | Register Name: EMAC_INT_STATUS |
|---|---|---|---|
| Bit | Read/Write | Default | Description |
| 31:1 | / | / | / |
| 0 | R/W | 0 | RIS<br>RGMII Interrupt Status<br>This bit is set due to any change in value of the Link Status of RGMII interface. This bit is cleared when the user makes a read operation the RGMII Status register. This bit is valid only when the optional RGMII PHY interface is selected during configuration and operation. |

### 7.8.4.9. EMAC Interrupt Mask Register(Default Value: 0x0000_0000)

| Offset: 0x3C | | | Register Name: EMAC_INT_MASK |
|---|---|---|---|
| Bit | Read/Write | Default | Description |
| 31:1 | / | / | / |
| 0 | R/W | 0 | RIM<br>RGMII Interrupt Mask<br>This bit when set, will disable the assertion of the interrupt signal due to the setting of RGMII Interrupt Status bit. |

### 7.8.4.10. EMAC MAC Address 0 High Register(Default Value: 0x8000_FFFF)

| Offset: 0x40 | | | Register Name: EMAC_ADDR0_HIGH |
|---|---|---|---|
| Bit | Read/Write | Default | Description |
| 31 | R | 1 | MO<br>Always 1. |
| 30:16 | / | / | / |
| 15:0 | R/W | 0 | A[47:32]<br>MAC Address0 [47:32]<br>This field contains the upper 16 bits (47:32) of the 6-byte first MAC address. This is used by the MAC for filtering for received frames and for inserting the MAC address in the Transmit Flow Control (PAUSE) Frames |

### 7.8.4.11. EMAC MAC Address 0 Low Register(Default Value: 0xFFFF_FFFF)

| Offset: 0x44 | Register Name: EMAC_ADDR0_LOW |
|---|---|

| Bit | Read/Write | Default | Description |
|---|---|---|---|
| 31:0 | R/W | 0 | A[31:0]<br>MAC Address0 [31:0]<br>This field contains the lower 32 bits of the 6-byte first MAC address. This is used by the MAC for filtering for received frames and for inserting the MAC address in the Transmit Flow Control (PAUSE) Frames |

### 7.8.4.12. EMAC MAC Address x High Register(Default Value: 0x0000_FFFF)

| Offset: 0x40+8*x (x=1~7) | | | Register Name: EMAC_ADDRx_HIGH |
|---|---|---|---|
| Bit | Read/Write | Default | Description |
| 31 | R/W | 0 | AE<br>Address Enable<br>When this bit is set, this address will be used for address filter. When reset, the address filter will ignore this address. |
| 30 | R/W | 0. | SA<br>Source Address<br>When this bit is set, the address is used to compare with the source address field of the received frame.<br>When this bit is reset, this address is used to compare with the destination address field fo the received frame. |
| 29:24 | R/W | 0 | MBC<br>Mask Byte Control<br>When set high, the EMAC core does not compare the corresponding byte of received DA/SA with this address. Each bit controls the masking of the bytes as follows:<br>Bit 29: MAC Address High Reg x[15:8]<br>Bit 28: MAC Address High Reg x[7:0]<br>Bit 27: MAC Address Low Reg x[31:24]<br>…<br>Bit 24: MAC Address Low Reg x[7:0] |
| 23:16 | / | / | / |
| 15:0 | R/W | 0 | A[47:32]<br>MAC Address0 [47:32]<br>This field contains the upper 16 bits(47:32) of number x MAC address. |

### 7.8.4.13. EMAC MAC Address x Low Register(Default Value: 0xFFFF_FFFF)

| Offset: 0x44+8*x (x=1~7) | | | Register Name: EMAC_ADDRx_LOW |
|---|---|---|---|
| Bit | Read/Write | Default | Description |
| 31:0 | R/W | 0 | A[31:0]<br>MAC Address0 [31:0] |

| | | 0 | This field contains the lower 32 bits(31:0) of number x MAC address. |

*Note:*

1.MAC Address 0~7 can be used for unicast destination address filter;

2.MAC Address 0 can not be used for multicast destination address filter and unicast source address filter;

### 7.8.4.14. EMAC RGMII Status Register(Default Value: 0x0000_0000)

| Offset: 0xD8 | | | Register Name: EMAC_RGMII_STATUS |
|---|---|---|---|
| Bit | Read/Write | Default | Description |
| 31:4 | / | / | / |
| 3 | R | 0 | LST<br>Link Status.<br>Indicates whether the link is:<br>0: down<br>1: up |
| 2:1 | R | 0 | LSP<br>Link Speed<br>Indicates the current speed of the link:<br>00: 2.5 MHz<br>01: 25 MHz<br>10: 125 MHz. |
| 0 | R | 0 | LM<br>Link Mode<br>Indicates the current mode of operation of the link:<br>0: Half-Duplex mode<br>1: Full-Duplex mode |

## 7.8.5.  EMAC DMA Register Description

### 7.8.5.1.  EDMA Bus Mode Register(Default Value: 0x0000_0000)

| Offset: 0x1000 | | | Register Name: EDMA_BUS_MODE |
|---|---|---|---|
| Bit | Read/Write | Default | Description |
| 31:26 | / | / | / |
| 25 | R/W | 0 | AAL<br>Address-Aligned Beats<br>When this bit is set high and the FB bit equals 1, the AHB interface generates all bursts aligned to the start address LS bits. If the FB bit equals 0, the first burst (accessing the data buffer's start address) is not aligned, but subsequent bursts are aligned to the address. This bit is valid only in EMAC-AHB configuration, and reserved (RO with default value 0) in all other |

| | | | configurations. |
|---|---|---|---|
| 24 | R/W | 0 | 4xPBL Mode<br>When set high, this bit multiplies the PBL value programmed (bits [22:17] and bits [13:8]) four times. Thus the DMA will transfer data in to a maximum of 4, 8, 16, 32, 64 and 128 beats depending on the PBL value. |
| 23 | R/W | 0 | USP<br>Use Separate PBL<br>When set high, it configures the RxDMA to use the value configured in bits [22:17] as PBL while the PBL value in bits [13:8] is applicable to TxDMA operations only. When reset to low, the PBL value in bits [13:8] is applicable for both DMA engines. |
| 22:17 | R/W | 0 | RPBL<br>RxDMA PBL<br>These bits indicate the maximum number of beats to be transferred in one RxDMA transaction. This will be the maximum value that is used in a single block Read/Write. The RxDMA will always attempt to burst as specified in RPBL each time it starts a Burst transfer on the host bus. RPBL can be programmed with permissible values of 1, 2, 4, 8, 16, and 32. Any other value will result in undefined behavior. These bits are valid and applicable only when USP is set high. |
| 16 | R/W | 0 | FB<br>Fixed Burst<br>This bit controls whether the AHB Master interface performs fixed burst transfers or not. When set, the AHB will use only SINGLE, INCR4, INCR8 or INCR16 during start of normal burst transfers. When reset, the AHB will use SINGLE and INCR burst transfer operations. |
| 15:14 | R/W | 0 | PR<br>Rx:Tx priority ratio.<br>RxDMA requests given priority over TxDMA requests in the following ratio. This is valid only when the DA bit is reset.<br>00: 1:1<br>01: 2:1<br>10: 3:1<br>11: 4:1 |
| 13:8 | R/W | 0 | PBL<br>Programmable Burst Length<br>These bits indicate the maximum number of beats to be transferred in one DMA transaction. This will be the maximum value that is used in a single block Read/Write. The DMA will always attempt to burst as specified in PBL each time it starts a Burst transfer on the host bus. PBL can be programmed with permissible values of 1, 2, 4, 8, 16, and 32. Any other value will result in undefined behavior. |
| 7 | / | / | / |
| 6:2 | R/W | 0 | DSL<br>Descriptor Skip Length |

| | | | This bit specifies the number of Word/Dword/Lword (depending on 32/64/128-bit bus) to skip between two unchained descriptors. The address skipping starts from the end of current descriptor to the start of next descriptor. When DSL value equals zero, then the descriptor table is taken as contiguous by the DMA, in Ring mode |
|---|---|---|---|
| 1 | R/W | 0 | DA<br>DMA Arbitration scheme<br>0: Round-robin with Rx:Tx priority given in bits [15:14]<br>1: Rx has priority over Tx |
| 0 | R/W | 0 | SWR<br>Software Reset<br>When this bit is set, the MAC DMA Controller resets all EMAC Subsystem internal registers and logic. It is cleared automatically after the reset operation has completed in all of the core clock domains. Read a 0 value in this bit before re-programming any register of the core. |

### 7.8.5.2.  EDMA Transmit Receive Poll Demand Register(Default Value: 0x0000_0000)

| Offset: 0x1004 | | | Register Name: EDMA_XMT_POLL |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default** | **Description** |
| 31:0 | R/W | 0 | TPD<br>Transmit Poll Demand<br>When these bits are written with any value, the DMA reads the current descriptor. If that descriptor is not available (owned by Host), transmission returns to the Suspend state. If the descriptor is available, transmission resumes. |

### 7.8.5.3.  EDMA Receive Poll Demand Register(Default Value: 0x0000_0000)

| Offset: 0x1008 | | | Register Name: EDMA_RCV_POLL |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default** | **Description** |
| 31:0 | R/W | 0 | RPD<br>Receive Poll Demand<br>When these bits are written with any value, the DMA reads the current descriptor. If that descriptor is not available (owned by Host), reception returns to the Suspended state. If the descriptor is available, the Receive DMA returns to active state. |

### 7.8.5.4.  EDMA Receive Descriptor List Address Register(Default Value: 0x0000_0000)

| Offset: 0x100C | Register Name: EDMA_RCV_LIST |
|---|---|

| Bit | Read/Write | Default | Description |
|---|---|---|---|
| 31:0 | R/W | 0 | SRL<br>Start of Receive List<br>This field contains the base address of the First Descriptor in the Receive Descriptor list. The LSB bits [1:0] (for 32-bit bus width) will be ignored and taken as all-zero by the DMA internally. Hence these LSB bits are Read Only |

### 7.8.5.5. EDMA Transmit Receive Descriptor List Address Register(Default Value: 0x0000_0000)

| Offset: 0x1010 | | | Register Name: EDMA_XMT_LIST |
|---|---|---|---|
| Bit | Read/Write | Default | Description |
| 31:0 | R/W | 0 | STL<br>Start of Transmit List<br>This field contains the base address of the First Descriptor in the Transmit Descriptor list. The LSB bits [1:0] (for 32-bit bus width) will be ignored and taken as all-zero by the DMA internally. Hence these LSB bits are Read Only. |

### 7.8.5.6. EDMA Status Register(Default Value: 0x0000_0000)

| Offset: 0x1014 | | | Register Name: EDMA_STATUS |
|---|---|---|---|
| Bit | Read/Write | Default | Description |
| 31:27 | / | / | / |
| 26 | R | 0 | GLI<br>EMAC Line interface Interrupt<br>This bit reflects an interrupt event in the EMAC Core's PCS or RGMII interface block. |
| 25:23 | R | 0 | EB<br>Error Bits<br>These bits indicate the type of error that caused a Bus Error (error response on the AHB interface). |
| 22:20 | R | 0 | TS<br>Transmit Process State<br>These bits indicate the Transmit DMA FSM state. This field does not generate an interrupt.<br>000: Stopped; Reset or Stop Transmit Command issued.<br>001: Running; Fetching Transmit Transfer Descriptor.<br>010: Running; Waiting for status.<br>011: Running; Reading Data from host memory buffer and queuing it to transmit buffer (Tx FIFO).<br>100: Reserved<br>101: Reserved<br>110: Suspended; Transmit Descriptor Unavailable or Transmit Buffer Underflow. |

| | | | 111: Running; Closing Transmit Descriptor. |
|---|---|---|---|
| 19:17 | R | 0 | RS<br>Receive Process State<br>These bits indicate the Receive DMA FSM state. This field does not generate an interrupt.<br>000: Stopped: Reset or Stop Receive Command issued.<br>001: Running: Fetching Receive Transfer Descriptor.<br>010: Reserved for future use.<br>011: Running: Waiting for receive packet.<br>100: Suspended: Receive Descriptor Unavailable.<br>101: Running: Closing Receive Descriptor.<br>110: Reserved for future use.<br>111: Running: Transferring the receive packet data from receive buffer to host memory. |
| 16 | R | 0 | NIS<br>Normal Interrupt Summary<br>Normal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled:<br>• Transmit Interrupt<br>• Transmit Buffer Unavailable<br>• Receive Interrupt<br>• Early Receive Interrupt<br>Only unmasked bits affect the Normal Interrupt Summary bit. This is a sticky bit and must be cleared (by writing a 1 to this bit) each time a corresponding bit that causes NIS to be set is cleared. |
| 15 | R | 0 | AIS<br>Abnormal Interrupt Summary<br>Abnormal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled:<br>• Transmit Process Stopped<br>• Transmit Jabber Timeout<br>• Receive FIFO Overflow<br>• Transmit Underflow<br>• Receive Buffer Unavailable<br>• Receive Process Stopped<br>• Receive Watchdog Timeout<br>• Early Transmit Interrupt<br>• Fatal Bus Error<br>Only unmasked bits affect the Abnormal Interrupt Summary bit. This is a sticky bit and must be cleared each time a corresponding bit that causes AIS to be set is cleared. |
| 14 | R | 0 | ERI<br>Early Receive Interrupt<br>This bit indicates that the DMA had filled the first data buffer of the packet. |
| 13 | R | 0 | FBI |

| | | | Fatal Bus Error Interrupt |
|---|---|---|---|
| | | | This bit indicates that a bus error occurred. When this bit is set, the corresponding DMA engine disables all its bus accesses. |
| 12:11 | / | / | / |
| 10 | R | 0 | ETI<br>Early Transmit Interrupt<br>This bit indicates that the frame to be transmitted was fully transferred to the MTL Transmit FIFO. |
| 9 | R | 0 | RWT<br>Receive Watchdog Timeout<br>This bit is asserted when a frame with a length greater than 2,048 bytes is received (10,240 when Jumbo Frame mode is enabled). |
| 8 | R | 0 | RPS<br>Receive Process Stopped<br>This bit is asserted when the Receive Process enters the Stopped state. |
| 7 | R | 0 | RU<br>Receive Buffer Unavailable<br>This bit indicates that the Next Descriptor in the Receive List is owned by the host and cannot be acquired by the DMA. Receive Process is suspended. To resume processing Receive descriptors, the host should change the ownership of the descriptor and issue a Receive Poll Demand command. If no Receive Poll Demand is issued, Receive Process resumes when the next recognized incoming frame is received. Register 5[7] is set only when the previous Receive Descriptor was owned by the DMA. |
| 6 | R | 0 | RI<br>Receive Interrupt<br>This bit indicates the completion of frame reception. Specific frame status information has been posted in the descriptor. Reception remains in the Running state. |
| 5 | R | 0 | UNF<br>Transmit Underflow<br>This bit indicates that the Transmit Buffer had an Underflow during frame transmission. Transmission is suspended and an Underflow Error TDES0[1] is set. |
| 4 | R | 0 | OVF<br>Receive Overflow<br>This bit indicates that the Receive Buffer had an Overflow during frame reception. If the partial frame is transferred to application, the overflow status is set in RDES0[11]. |
| 3 | R | 0 | TJT<br>Transmit Jabber Timeout<br>This bit indicates that the Transmit Jabber Timer expired, meaning that the transmitter had been excessively active. The transmission process is aborted and placed in the Stopped state. This causes the Transmit Jabber Timeout TDES0[14] flag to assert. |
| 2 | R | 0 | TU |

| | | | |
|---|---|---|---|
| | | | Transmit Buffer Unavailable |
| | | | This bit indicates that the Next Descriptor in the Transmit List is owned by the host and cannot be acquired by the DMA. Transmission is suspended. Bits[22:20] explain the Transmit Process state transitions. To resume processing transmit descriptors, the host should change the ownership of the bit of the descriptor and then issue a Transmit Poll Demand command. |
| 1 | R | 0 | TPS<br>Transmit Process Stopped<br>This bit is set when the transmission is stopped |
| 0 | R | 0 | TI:<br>Transmit Interrupt<br>This bit indicates that frame transmission is finished and TDES1[31] is set in the First Descriptor. |

### 7.8.5.7. EDMA Operation Mode Register(Default Value: 0x0000_0000)

| Offset: 0x1018 | | | Register Name: EDMA_OPERATION |
|---|---|---|---|
| Bit | Read/Write | Default | Description |
| 31:26 | / | / | / |
| 25 | R/W | 0 | RSF<br>Receive Store and Forward<br>When this bit is set, the MTL only reads a frame from the Rx FIFO after the complete frame has been written to it, ignoring RTC bits. When this bit is reset, the Rx FIFO operates in Cut-Through mode, subject to the threshold specified by the RTC bits. |
| 24 | R/W | 0 | DFF<br>Disable Flushing of Received Frames<br>When this bit is set, the RxDMA does not flush any frames due to the unavailability of receive descriptors/buffers as it does normally when this bit is reset. |
| 23 | R/W | 0 | RFA[2]<br>MSB of Threshold for Activating Flow Control<br>If the EMAC-UNIV is configured for an Rx FIFO depth of 8 KB or more, this bit (when set) provides additional threshold levels for activating the Flow Control in both Half- Duplex and Full-Duplex modes. This bit (as Most Significant Bit) along with the RFA (bits [10:9]) give the following thresholds for activating flow control.<br>100: Full minus 5 KB<br>101: Full minus 6 KB<br>110: Full minus 7 KB<br>111: Reserved |
| 22 | R/W | 0 | RFD[2]<br>MSB of Threshold for Deactivating Flow Control<br>If the EMAC-UNIV is configured for an Rx FIFO depth of 8 KB or more, this bit (when set) provides additional threshold levels for deactivating the Flow Control |

| | | | in both Half- Duplex and Full-Duplex modes. This bit (as Most Significant Bit) along with the RFD (bits [12:11]) give the following thresholds for deactivating flow control.<br>100: Full minus 5 KB<br>101: Full minus 6 KB<br>110: Full minus 7 KB<br>111: Reserved |
|---|---|---|---|
| 21 | R/W | 0 | TSF<br>Transmit Store and Forward<br>When this bit is set, transmission starts when a full frame resides in the MTL Transmit FIFO. The TTC value is ignored. |
| 20 | R/W | 0 | FTF<br>Flush Transmit FIFO<br>When this bit is set, the transmit FIFO controller logic is reset to its default values and thus all data in the Tx FIFO is lost/flushed. This bit is cleared internally when the flushing operation is completed fully. The Operation Mode register should not be written to until this bit is cleared. |
| 19:17 | / | / | / |
| 16:14 | R/W | 0 | TTC<br>Transmit Threshold Control<br>These three bits control the threshold level of the MTL Transmit FIFO. Transmission starts when the frame size within the MTL Transmit FIFO is larger than the threshold. In addition, full frames with a length less than the threshold are also transmitted. These bits are used only when the TSF bit (Bit 21) is reset.<br>000: 64<br>001: 128<br>010: 192<br>011: 256<br>100: 40<br>101: 32<br>110: 24<br>111: 16 |
| 13 | R/W | 0 | ST<br>Start/Stop Transmission Command<br>When this bit is set, transmission is placed in the Running state.When this bit is reset, the transmission process is placed in the Stopped state after completing the transmission of the current frame. |
| 12:11 | R/W | 0 | RFD<br>Threshold for deactivating flow control (in both HD and FD)<br>These bits control the threshold (Fill-level of Rx FIFO) at which the flow-control is deasserted after activation.<br>00: Full minus 1 KB<br>01: Full minus 2 KB<br>10: Full minus 3 KB<br>11: Full minus 4 KB |

| | | | |
|---|---|---|---|
| 0 | / | / | Note that the deassertion is effective only after flow control is asserted. An additional bit (RFD[2]) is used for more threshold levels. |
| 10:9 | R/W | 0 | RFA<br>Threshold for activating flow control (in both HD and FD)<br>These bits control the threshold (Fill level of Rx FIFO) at which flow control is activated.<br>00: Full minus 1 KB<br>01: Full minus 2 KB<br>10: Full minus 3 KB<br>11: Full minus 4 KB<br>Note that an additional bit (RFA[2]) is used for more threshold levels. |
| 8 | R/W | 0 | EFC<br>Enable HW flow control<br>When this bit is set, the flow control signal operation based on fill-level of Rx FIFO is enabled. When reset, the flow control operation is disabled. |
| 7 | R/W | 0 | FEF<br>Forward Error Frames<br>When this bit is reset, the Rx FIFO drops frames with error status (CRC error, collision error, GMII_ER, giant frame, watchdog timeout, overflow). |
| 6 | R/W | 0 | FUF<br>Forward Undersized Good Frames<br>When set, the Rx FIFO will forward Undersized frames (frames with no Error and length less than 64 bytes) including pad-bytes and CRC). |
| 5 | / | / | / |
| 4:3 | R/W | 0 | RTC<br>Receive Threshold Control<br>These two bits control the threshold level of the MTL Receive FIFO. Transfer (request) to DMA starts when the frame size within the MTL Receive FIFO is larger than the threshold. In addition, full frames with a length less than the threshold are transferred automatically. Note that value of 11 is not applicable if the configured Receive FIFO size is 128 bytes. These bits are valid only when the RSF bit is zero, and are ignored when the RSF bit is set to 1.<br>00: 64<br>01: 32<br>10: 96<br>11: 128 |
| 2 | R/W | 0 | OSF<br>Operate on Second Frame<br>When this bit is set, this bit instructs the DMA to process a second frame of Transmit data even before status for first frame is obtained. |
| 1 | R/W | 0 | SR<br>Start/Stop Receive<br>When this bit is set, the Receive process is placed in the Running state. When this bit is cleared, RxDMA operation is stopped after the transfer of the current frame. |
| 0 | / | / | / |

### 7.8.5.8. EDMA Interrupt Enable Register(Default Value: 0x0000_0000)

| Offset: 0x101C | | | Register Name: EDMA_INTR_ENA |
|---|---|---|---|
| Bit | Read/Write | Default | Description |
| 31:17 | / | / | / |
| 16 | R/W | 0 | NIE<br>Normal Interrupt Summary Enable<br>When this bit is set, a normal interrupt is enabled. When this bit is reset, a normal interrupt is disabled. This bit enables the following bits:<br>• Transmit Interrupt<br>• Transmit Buffer Unavailable<br>• Receive Interrupt<br>• Early Receive Interrupt |
| 15 | R/W | 0 | AIE<br>Abnormal Interrupt Summary Enable<br>When this bit is set, an Abnormal Interrupt is enabled. When this bit is reset, an Abnormal Interrupt is disabled. This bit enables the following bits<br>• Transmit Process Stopped<br>•Transmit Jabber Timeout<br>• Receive Overflow<br>• Transmit Underflow<br>• Receive Buffer Unavailable<br>• Receive Process Stopped<br>• Receive Watchdog Timeout<br>• Early Transmit Interrupt<br>• Fatal Bus Error |
| 14 | R/W | 0 | ERE<br>Early Receive Interrupt Enable<br>When this bit is set with Normal Interrupt Summary Enable, Early Receive Interrupt is enabled. When this bit is reset, Early Receive Interrupt is disabled |
| 13 | R/W | 0 | FBE<br>Fatal Bus Error Enable<br>When this bit is set with Abnormal Interrupt Summary Enable, the Fatal Bus Error Interrupt is enabled. When this bit is reset, Fatal Bus Error Enable Interrupt is disabled. |
| 12:11 | / | / | / |
| 10 | R/W | 0 | ETE<br>Early Transmit Interrupt Enable<br>When this bit is set with an Abnormal Interrupt Summary Enable, Early Transmit Interrupt is enabled. When this bit is reset, Early Transmit Interrupt is disabled |
| 9 | R/W | 0 | RWE<br>Receive Watchdog Timeout Enable<br>When this bit is set with Abnormal Interrupt Summary Enable, the Receive |

| | | | Watchdog Timeout Interrupt is enabled. When this bit is reset, Receive Watchdog Timeout Interrupt is disabled |
|---|---|---|---|
| 8 | R/W | 0 | RSE<br>Receive Stopped Enable<br>When this bit is set with Abnormal Interrupt Summary Enable, Receive Stopped Interrupt is enabled. When this bit is reset, Receive Stopped Interrupt is disabled |
| 7 | R/W | 0 | RUE<br>Receive Buffer Unavailable Enable<br>When this bit is set with Abnormal Interrupt Summary Enable, Receive Buffer Unavailable Interrupt is enabled. When this bit is reset, the Receive Buffer Unavailable Interrupt is disabled |
| 6 | R/W | 0 | RIE<br>Receive Interrupt Enable<br>When this bit is set with Normal Interrupt Summary Enable, Receive Interrupt is enabled. When this bit is reset, Receive Interrupt is disabled |
| 5 | R/W | 0 | UNE<br>Underflow Interrupt Enable<br>When this bit is set with Abnormal Interrupt Summary Enable, Transmit Underflow Interrupt is enabled. When this bit is reset, Underflow Interrupt is disabled. |
| 4 | / | / | OVE<br>Overflow Interrupt Enable<br>When this bit is set with Abnormal Interrupt Summary Enable, Receive Overflow Interrupt is enabled. When this bit is reset, Overflow Interrupt is disabled |
| 3 | R/W | 0 | TJE<br>Transmit Jabber Timeout Enable<br>When this bit is set with Abnormal Interrupt Summary Enable, Transmit Jabber Timeout Interrupt is enabled. When this bit is reset, Transmit Jabber Timeout Interrupt is disabled. |
| 3 | R/W | 0 | THE<br>Transmit Jabber Timeout Enable<br>When this bit is set with Abnormal Interrupt Summary Enable, Transmit Jabber Timeout Interrupt is enabled. When this bit is reset, Transmit Jabber Timeout Interrupt is disabled. |
| 2 | R/W | 0 | TUE<br>Transmit Buffer Unavailable Enable<br>When this bit is set with Normal Interrupt Summary Enable, Transmit Buffer Unavailable Interrupt is enabled. When this bit is reset, Transmit Buffer Unavailable Interrupt is disabled. |
| 1 | R/W | 0 | TSE<br>Transmit Stopped Enable<br>When this bit is set with Abnormal Interrupt Summary Enable, Transmission Stopped Interrupt is enabled. When this bit is reset, Transmission Stopped Interrupt is disabled. |
| 0 | R/W | 0 | TIE |

| | | | Transmit Interrupt Enable |
|---|---|---|---|
| | | | When this bit is set with Normal Interrupt Summary Enable, Transmit Interrupt is enabled. When this bit is reset, Transmit Interrupt is disabled. |

### 7.8.5.9. EDMA Missed Frame and Buffer Overflow Counter Register(Default Value: 0x0000_0000)

| Offset: 0x1020 | | | Register Name: EDMA_MISSED_FRAME |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default** | **Description** |
| 31:29 | / | / | / |
| 28 | R/W | 0 | OFBO<br>Overflow bit for FIFO Overflow Counter |
| 27:17 | R/W | 0 | MFA<br>Missed Frame Counter by Application<br>Indicates the number of frames missed by the application. |
| 16 | R/W | 0 | OFBM<br>Overflow bit for Missed Frame Counter |
| 15:0 | R/W | 0 | MFC<br>Missed Frame Counter by controller<br>Indicates the number of frames missed by the controller due to the Host Receive Buffer being unavailable. |

### 7.8.5.10. EDMA Current Host Transmit Descriptor Register(Default Value: 0x0000_0000)

| Offset: 0x1048 | | | Register Name: EDMA_CUR_TX_DESC |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default** | **Description** |
| 31:0 | R | 0 | Host Transmit Descriptor Address Pointer Cleared on Reset. Pointer updated by DMA during operation |

### 7.8.5.11. EDMA Current Host Receive Descriptor Register(Default Value: 0x0000_0000)

| Offset: 0x104C | | | Register Name: EDMA_CUR_RX_DESC |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default** | **Description** |
| 31:0 | R | 0 | Host Receive Descriptor Address Pointer Cleared on Reset. Pointer updated by DMA during operation. |

### 7.8.5.12. EDMA Current Host Transmit Buffer Address Register(Default Value: 0x0000_0000)

| Offset: 0x1050 | | | Register Name: EDMA_CUR_TX_BUF |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default** | **Description** |

| 31:0 | R | 0 | Host Transmit Buffer Address Pointer Cleared on Reset. Pointer updated by DMA during operation. |

### 7.8.5.13. EDMA Current Host Receive Buffer Address Register(Default Value: 0x0000_0000)

| Offset: 0x1054 | | | Register Name: EDMA_CUR_RX_BUF |
|---|---|---|---|
| Bit | Read/Write | Default | Description |
| 31:0 | R | 0 | Host Receive Buffer Address Pointer Cleared on Reset. Pointer updated by DMA during operation. |

# Appendix

## Control signal and data port mapping

| I/O | Para RGB | Serial RGB 1st | Serial RGB 2nd | Serial RGB 3rd | CCIR 656 | CPU cmd | CPU 18bit 256K | CPU 16bit 256K 1st | CPU 16bit 256K 2nd | CPU 16bit 256K 3rd | CPU 16bit 256K 1st | CPU 16bit 256K 2nd | CPU 16bit 65K 1st | CPU 16bit 65K 2nd | CPU 8bit 256K 1st | CPU 8bit 256K 2nd | CPU 8bit 256K 3rd | CPU 8bit 65K 1st | CPU 8bit 65K 2nd | CPU 9bit 256K 1st | CPU 9bit 256K 2nd | Single LVDS | TTL |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| IO0 | VSYNC | | | | | CS | | | | | | | | | | | | | | | | D3N | STV |
| IO1 | HSYNC | | | | | RD | | | | | | | | | | | | | | | | D3P | SKV |
| IO2 | DCLK | | | | | WR | | | | | | | | | | | | | | | | CKP | STH |
| IO3 | DE | | | | | RS | | | | | | | | | | | | | | | | CKN | OEH |
| D23 | R7 | | | | | D23 | R5 | R5 | B5 | G5 | R5 | R5 | B5 | R4 | | | | | | | | D2N | R7 |
| D22 | R6 | | | | | D22 | R4 | R4 | B4 | G4 | R4 | R4 | B4 | R3 | | | | | | | | D2P | R6 |
| D21 | R5 | | | | | D21 | R3 | R3 | B3 | G3 | R3 | R3 | B3 | R2 | | | | | | | | D1N | R5 |
| D20 | R4 | | | | | D20 | R2 | R2 | B2 | G2 | R2 | R2 | B2 | R1 | | | | | | | | D1P | R4 |
| D19 | R3 | | | | | D19 | R1 | R1 | B1 | G1 | R1 | R1 | B1 | R0 | | | | | | | | D0N | R3 |
| D18 | R2 | | | | | D18 | R0 | R0 | B0 | G0 | R0 | R0 | B0 | G5 | | | | | | | | D0P | R2 |
| D17 | R1 | | | | | D17 | | | | | | | | | | | | | | | | | CKH |
| D16 | R0 | | | | | D16 | | | | | | | | | | | | | | | | | REV |
| D15 | G7 | | | | | D15 | G5 | | | | | | G4 | | | | | | | | | | G7 |
| D14 | G6 | | | | | D14 | G4 | | | | | | G3 | | | | | | | | | | G6 |
| D13 | G5 | | | | | D13 | G3 | | | | | | | | | | | | | | | | G5 |
| D12 | G4 | D17 | D27 | D37 | D7 | D12 | G2 | G5 | R5 | B5 | G5 | B5 | G5 | G2 | R5 | G5 | B5 | R4 | G2 | R5 | G2 | | G4 |
| D11 | G3 | D16 | D26 | D36 | D6 | D11 | G1 | G4 | R4 | B4 | G4 | B4 | G4 | G1 | R4 | G4 | B4 | R3 | G1 | R4 | G1 | | G3 |
| D10 | G2 | D15 | D25 | D35 | D5 | D10 | G0 | G3 | R3 | B3 | G3 | B3 | G3 | G0 | R3 | G3 | B3 | R2 | G0 | R3 | G0 | | G2 |
| D9 | G1 | | | | | D9 | | | | | | | | | | | | | | | | | REVB |
| D8 | G0 | | | | | D8 | | | | | | | | | | | | | | | | | CKH1 |
| D7 | B7 | D14 | D24 | D34 | D4 | D7 | B5 | G2 | R2 | B2 | G2 | B2 | G2 | B4 | R2 | G2 | B2 | R1 | B4 | R2 | B5 | | B7 |
| D6 | B6 | D13 | D23 | D33 | D3 | D6 | B4 | G1 | R1 | B1 | G1 | B1 | G1 | B3 | R1 | G1 | B1 | R0 | B3 | R1 | B4 | | B6 |
| D5 | B5 | D12 | D22 | D32 | D2 | D5 | B3 | G0 | R0 | B0 | G0 | B0 | G0 | B2 | R0 | G0 | B0 | G5 | B2 | R0 | B3 | | B5 |
| D4 | B4 | D11 | D21 | D31 | D1 | D4 | B2 | | | | | | | | | | | G4 | B1 | G5 | B2 | | B4 |
| D3 | B3 | D10 | D20 | D30 | D0 | D3 | B1 | | | | | | | | | | | G3 | B0 | G4 | B1 | | B3 |
| D2 | B2 | | | | | D2 | B0 | | | | | | | | | | | | | G3 | B0 | | B2 |
| D1 | B1 | | | | | D1 | | | | | | | | | | | | | | | | | OEV |
| D0 | B0 | | | | | D0 | | | | | | | | | | | | | | | | | CKH2 |